

SÉRGIO ALÍPIO DOMINGUES DEUSDADO

**INTEGRAÇÃO ADAPTATIVA DE
APLICAÇÕES *MULTICAST*
PARA CONFERÊNCIA MULTIMÉDIA**

Dissertação

*para satisfação parcial dos requisitos do
Curso de Mestrado em Informática
(Especialização em Sistemas Distribuídos,
Comunicações por Computador e
Arquitecturas de Computadores)*

**UNIVERSIDADE DO MINHO
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE INFORMÁTICA**

BRAGA, JULHO DE 2002

ABSTRACT

The Internet, mainly the new generation, has standing as panacea for the demands of digital communication in several domains.

Education, and distance learning particularly, did not escape the "webization" process. With WWW service evolution, new distributed multimedia conference applications were incorporated bringing improved interactivity and pro-human relations. Groupware applications are increasingly representative in the Internet home applications market, however, the Quality of Service (QoS) provided by the network is still a limitation, which impairs performance assurance. Such applications found in multicast technology an allied for their implementation and scalability.

Multicast support is mandatory in the design of new generation networks. Being an efficient method of group communication, multicast will boost future internetworking and applications development. Although reducing network traffic, group dimension and Internet heterogeneity may originate scalability problems. In environments where the available resources suffer considerable random fluctuations, applications are due to provide mechanisms to preserve quality of the group communication critical parameters.

The application ability to adapt itself dynamically depending on the resource availability can be considered a quality factor. Tolerant real-time applications, such as videoconferences, are candidates to benefit from this innovation. However, not all include adaptive technology able to provide both end-system and network quality-aware behaviour. Adaptation, in these cases, can be reached by introducing a multi-platform middleware layer responsible for tutoring the applications' resources (adjudication or limitation) based on the available processing and networking capabilities.

To meet all these technological contributions, an adaptive platform was developed integrating public domain multicast tools, and then applied to a web based distance learning system. The system is user-centered (e-student), aiming good pedagogical practices and proactive usability for multimedia and network resources. The services provided, including QoS adapted interactive multimedia multicast conferences (MMC), are fully integrated and transparent to the end-users.

RESUMO

A Internet, sobretudo a nova geração, perfila-se como panaceia para as exigências de comunicação digital nos mais vastos domínios.

Na educação, e particularmente no ensino à distância, a via da "webização" não ficou por explorar. À medida que o serviço WWW foi evoluindo, incorporaram-se ferramentas de conferência multimédia distribuída que fomentam a interactividade e as relações pró-humanas. As aplicações de comunicação em grupo são, cada vez mais, representativas no rol de aplicações que usam a Internet mas, insipientes em sustentar o seu desempenho devido a uma Qualidade de Serviço (QoS) limitada. Tais aplicações, encontraram na tecnologia *multicast* uma aliada para a sua implantação e escalabilidade.

O suporte à tecnologia de difusão selectiva ou *multicast* encontra-se entre as directivas para a concepção de redes informáticas da nova geração. Sendo uma metodologia eficiente para comunicar em grupo, o *multicast* exponenciará o desenvolvimento futuro das tecnologias de interligação e aplicacionais. Apesar do tráfego de rede decrescer, a dimensão dos grupos e a heterogeneidade da Internet podem colocar problemas de escalabilidade. Em ambientes com recursos variáveis, compete às aplicações providenciar mecanismos de preservação da qualidade dos parâmetros críticos da comunicação em grupo.

A capacidade de adaptação da QoS de uma aplicação em face dos recursos disponíveis é um factor de qualidade dessa mesma aplicação. As aplicações de tempo crítico tolerantes, como as videoconferências, são potenciais beneficiárias desta inovação, no entanto, nem todas incorporam tecnologia adaptativa que lhes permita um comportamento "consciente" perante a QoS prestada pela rede e pelo sistema final. A adaptação, nestes casos, pode ser lograda com recurso a uma camada de *middleware* multiplataforma, que tutelarà a adjudicação ou limitação dos recursos de processamento e interligação disponíveis para as aplicações.

No intuito de congregiar todas estas contribuições tecnológicas, desenvolveu-se uma plataforma adaptativa integrando ferramentas *multicast* de domínio público, sendo esta aplicada a um sistema de educação à distância. Este sistema, baseado no serviço WWW, está centrado no cliente (e-aluno), privilegiando as boas práticas pedagógicas e a usabilidade pró-activa de meios telemáticos e multimediáticos. Os serviços disponibilizados, incluindo as conferências multimédia *multicast* (CMM) interactivas com QoS adaptada, são usáveis de forma integrada e transparente para o utilizador.

Índice

CAPÍTULO 1

INTRODUÇÃO.....	1
1.1. Redes de computadores.....	5
1.1.1. A adequação da arquitectura da Internet ao transporte <i>multicast</i>	5
1.1.2. Problemas do teletráfego	7
1.1.3. Aplicações que dependem da QoS das redes	8
1.1.4. Nova concepção de redes	9
1.2. Enquadramento do trabalho	11
1.2.1. Objectivos	12
1.3. Organização da dissertação.....	13

CAPÍTULO 2

TRABALHO RELACIONADO.....	15
2.1. Algumas considerações sobre o ensino à distância.....	16
2.1.1. Contemporização do EAD.....	16
2.1.2. Caracterização dos modelos de ensino	18
2.1.3. Riscos de exploração do ensino à distância	19
2.2. Modelos de ambientes de educação à distância existentes	21
2.2.1. O ambiente <i>Centra One</i>	22
2.2.2. O ambiente WebCT	24
2.3. Adaptação de QoS das aplicações controlada por <i>middleware</i>	25
2.4. Experiências de adaptabilidade e QoS em <i>multicast</i>	27
2.5. Analogias e projecções	29

CAPÍTULO 3

MULTICAST IP	32
3.1. Estabelecimento do <i>multicast</i> IP	33
3.1.1. O modelo " <i>host group</i> "	34
3.1.2. Endereçamento <i>multicast</i> IP	35
3.1.3. Controlar o âmbito do encaminhamento <i>multicast</i>	37
3.2. Benefícios do <i>multicast</i>	38
3.3. Custos do <i>multicast</i>.....	41
3.4. Mbone - <i>Multicast Backbone</i>	41
3.4.1. História do Mbone.....	42
3.4.2. O Mbone na actualidade	43
3.5. <i>Multicast</i> em IPv6.....	44
3.6. Implantação da tecnologia <i>multicast</i> IP em Portugal	46
3.6.1. O Mbone visto da rede do IPB (Instituto Politécnico de Bragança).....	46
3.6.2. Os ISPs nacionais	47
3.7. Utilização da tecnologia <i>multicast</i> no trabalho desenvolvido	48

CAPÍTULO 4

APLICAÇÕES MULTICAST	49
4.1. Categorias de aplicações <i>multicast</i>	52
4.2. Videoconferência	53
4.2.1. Benefícios.....	56
4.2.2. Problemas.....	56
4.3. Aplicações <i>multicast</i> orientadas à videoconferência	57
4.3.1. <i>rat - robust audio tool</i>	59
4.3.2. <i>vic - videoconferencing tool</i>	62
4.3.3. <i>JMF - Java Media Framework</i>	64
4.4. Ferramentas para espaço de trabalho partilhado - <i>Whiteboard (wb)</i> e <i>network text editor (nte)</i>	67
4.5. Integrar as diferentes aplicações <i>multicast</i>	67
4.6. Avaliação das aplicações para CMM (conferência multimédia <i>multicast</i>)	68

CAPÍTULO 5

QUALIDADE DE SERVIÇO: A REDE E AS APLICAÇÕES MULTICAST	71
5.1. QoS "sobre" IP	73
5.2. Parâmetros de Qualidade de Serviço (QoS) da rede	74
5.2.1. Largura de Banda.....	74
5.2.2. Atraso (<i>Delay</i>).....	75
5.2.3. <i>Jitter</i>	76
5.2.4. Fiabilidade ou perda de pacotes.....	77
5.2.5. Disponibilidade.....	77
5.3. Parâmetros de QoS das aplicações	78
5.4. Necessidade de QoS das aplicações envolvidas	80
5.5. QoS - Alternativas técnicas	82
5.5.1. Integração de serviços em <i>multicast</i>	84
5.6. Relevância para o trabalho a desenvolver	88

CAPÍTULO 6

TRABALHO DESENVOLVIDO	90
6.1. Introdução ao sistema adaptativo aplicado ao Ensino à Distância	92
6.1.1. Funcionalidades.....	92
6.1.2. Descrição da plataforma experimental.....	94
6.2. Arquitectura do sistema adaptativo de comunicação <i>multicast</i>	96
6.3. Descrição da interface <i>Web</i> e dos serviços disponibilizados	99
6.3.1. Funcionalidades disponibilizadas.....	100
6.4. Organização das sessões <i>multicast</i> e dos utilizadores de serviços	110
6.5. Módulo provedor de adaptabilidade	112
6.5.1. <i>Applet</i> propector de condições.....	114

6.5.2.	Formulário HTML de visualização dos dados recolhidos	116
6.5.3.	<i>Applet leParametros.class</i>	116
6.5.4.	Formulário HTML de visualização/edição da parametrização adaptativa.....	117
6.5.5.	<i>Applet</i> de iniciação das ferramentas <i>multicast</i>	117
6.6.	Parametrização das aplicações <i>multicast</i> de áudio e vídeo	118
6.6.1.	Necessidades de segurança na execução de <i>applets</i> no sistema cliente	122
6.7.	Resultados experimentais	124
6.7.1.	Cenários experimentais e resultados	125
6.7.2.	Discussão dos resultados.....	131
 CAPÍTULO 7		
 CONCLUSÕES		
		132
7.1.	Avaliação crítica do trabalho.....	134
7.1.1.	Descrição genérica das características da plataforma	134
7.1.2.	Principais resultados.....	135
7.1.3.	Avaliação das condições de desenvolvimento.....	136
7.1.4.	Avaliação das condições de teste e utilização.....	137
7.2.	Desenvolvimentos futuros	138
7.3.	Inovar é preciso.....	140
 REFERÊNCIAS.....		141
 ANEXO I		
 PROTOCOLOS DE SUPORTE A APLICAÇÕES <i>MULTICAST</i>		154
Protocolos de controlo		155
IGMP - <i>Internet Group Management Protocol</i>		155
Protocolos de <i>routing</i>.....		158
DVMRP - <i>Distance-Vector Multicast Routing Protocol</i>		159
MOSPF - <i>Multicast Open Shortest Path First</i>		160
CBT - <i>Core Based Tree</i>		161
PIM - <i>Protocol Independent Multicast</i>		162
BGMP - <i>Border Gateway Multicast Protocol</i>		164
Protocolos de transporte		165
Transporte de tráfego multimédia		165
O protocolo RTP - <i>Real Time Protocol</i>		166
Multiplexagem de sessões RTP.....		171
RTCP - <i>Real Time Control Protocol</i>		172
 ANEXO II		
 CÓDIGO-FONTE DE ALGUNS MÓDULOS DO <i>MIDDLEWARE</i>.....		177
Código do <i>applet</i> SimplePlayerApplet.class		177
Código do <i>applet</i> Auscultador.class		181
Código do <i>applet</i> leParametros.class.....		194
Código do <i>applet</i> IniciaApAdaptada.class		195

Índice de Ilustrações

Figura 1 - Perfil de QoS para um utilizador de <i>video-on-demand</i> (VOD).....	4
Figura 2 - Implantação de computadores com acesso à Internet nas Escolas Norte-Americanas	21
Tabela 1 - Exemplos de ambientes educacionais baseados em serviço WWW.....	22
Figura 3 - Aspecto da interface do <i>Centra One</i>	23
Figura 4 - Arquitectura do sistema <i>Agilos</i> [Li00].....	25
Figura 5 - Enquadramento do sistema desenvolvido face aos trabalhos relacionados.....	29
Ilustração 1 - Padrões dos bits de ordem mais elevada no primeiro octeto [Maufer97].....	35
Quadro 1 - Alguns endereços <i>multicast</i> reservados permanentes [Maufer97].....	36
Quadro 2 - Alguns endereços <i>multicast</i> reservados permanentes para outras aplicações [Maufer97].	37
Figura 6 - Tipos de comunicação.....	39
Figura 7 - Transmissão ponto-multiponto com encaminhamento <i>unicast</i>	39
Figura 8 - Transmissão ponto-multiponto com encaminhamento <i>multicast</i>	40
Figura 9 - Aspecto global da rede Mbone.....	43
Figura 10 - Estrutura actual do Mbone [Maufer97].....	44
Figura 11 - Organização de um endereço <i>multicast</i> IPv6 [Miller98].....	45
Quadro 3 - Valores de controlo de âmbito em IPv6 [Miller98].....	45
Quadro 4 - Output do comando <i>mriinfo</i> no router do IPB.	47
Figura 12 - Integração das aplicações <i>multicast</i> no sistema EAD a desenvolver.....	51
Quadro 5 - Características principais dos sistemas de videoconferência [Helbig97].....	55
Figura 13 - Ajuste de parâmetros de transmissão na interface do <i>vic</i>	63
Figura 14 - Aspecto da configuração para emissão áudio no <i>JMF Studio</i>	65
Figura 15 - Arquitectura de alto nível do JMF.....	66
Figura 16 - Visão geral dos factores que influenciam a qualidade percebida numa CMM [Watson01]....	69
Quadro 6 - Taxas de transmissão típicas para aplicações de rede [Martins99].....	75
Quadro 7 - Valores de referência para o atraso em áudio-conferência [Miras02].....	75
Figura 17 - Efeito do <i>jitter</i> para as aplicações [Martins99].....	76
Figura 18 - Mapeamento dos requisitos de atraso e perda de pacotes para aplicações multimédia interactivas [Miras02].....	80
Figura 19 - Comparação entre sensibilidade ao atraso e criticidade do tráfego.....	81
Figura 20 - Cenário de QoS fim-a-fim [Aurrecoechea98].....	82
Figura 21 - Esquema para efectuar a reserva de recursos para o grupo <i>multicast</i> heterogéneo [Wang01].	84
Figura 22 - Relação entre os conceitos de sessão [Mendes98].....	87
Figura 23 - Modelo de QoS para aplicações adaptativas multimédia <i>time-sensitive</i>	89
Figura 24 - Controlo da QoS das aplicações <i>multicast</i>	93
Figura 25 - Gráfico mensal (julho/02) do tráfego entre o IPB e a Internet (<i>router</i> da RCTS).....	94
Figura 26 - Sistema desenvolvido para suporte de ensino à distância.....	95
Figura 27 - Arquitectura do sistema adaptativo de comunicação <i>multicast</i> desenvolvido.....	97
Figura 28 - Aspecto geral da interface <i>Web</i>	99
Figura 29 - Registo de novo membro da comunidade.....	101
Figura 30 - Serviços para alunos.....	102
Figura 31 - Formulário para criação de sessões <i>multicast</i> (da responsabilidade do tutor).....	103
Figura 32 - Exemplo de acesso a vídeo a pedido.....	105
Figura 33 - Recepção de RTP/vídeo e RTP/áudio directamente no <i>browser</i>	106
Figura 34 - Aceder a sessões <i>multicast</i> na perspectiva de participante pontualmente activo.....	106
Figura 35 - Sequência de passos para participar numa sessão <i>multicast</i> com interacção e adaptação.	107
Figura 36 - Vídeo-conferência <i>multicast</i> multiparticipada.....	108
Figura 37 - Aspecto do fórum de discussão.....	109
Figura 38 - Aspecto do <i>chat</i> multimédia <i>multicast</i>	110
Quadro 8 - Tabelas <i>mysql</i> para gestão de sessões e de utilizadores.....	111
Tabela 2 - Tabela de modos de pré-parametriação das aplicações <i>multicast rat</i> e <i>vic</i> para emissão/recepção de áudio e vídeo.....	121
Quadro 9 - Alguns métodos para incrementar os privilégios de <i>applets</i> no <i>browser</i> NN.....	123
Quadro 10 - Alguns métodos para incrementar os privilégios de <i>applets</i> no <i>browser</i> IE.....	123
Figura 39 - Topologia da rede de testes e largura de banda máxima disponível.....	124
Figura 40 - Carga do CPU, verificada em diferentes máquinas, na transmissão de vídeo pelo <i>vic</i>	125
Figura 41 - Necessidades de largura de banda para a transmissão de vídeo pelo <i>vic</i> nos diferentes modos de adaptação e em diferentes cenários.....	126

Figura 42 - Afecção dos recursos de largura de banda aos membros do grupo sem utilizar adaptação.	127
Figura 43 - Escalabilidade do grupo <i>multicast</i> usando adaptação na qualidade de transmissão do <i>vic</i> .	127
Figura 44 - Modos de adaptação para pré-parametrização das aplicações <i>multicast</i> dos novos membros.	128
Figura 45 - Sobrecarregar a rede para efectuar testes de transmissão sem abundância de recursos.	129
Figura 46 - Taxa de perda de pacotes de vídeo RTP no formato <i>motion jpeg</i> , medidos pelo <i>vic</i> .	130
Figura 47 - Taxa de perdas de pacotes em emissão RTP no formato H.261.	130
Figura 48 - <i>Internet Group Management Protocol</i> - Mensagem de interrogação [Maufer97].	156
Quadro 11 - Classificação de protocolos de encaminhamento <i>multicast</i> em função do tipo de associação aos grupos.	158
Figura 49 - Exemplo dos mecanismos de controlo de comunicações de grupo, no protocolo PIM-SM. O emissor 2 pretende transmitir para o grupo através da árvore partilhada.	164
Figura 50 - Comunicação entre duas entidades usando o protocolo RTP [Sousa97].	168
Figura 51 - Utilização do RTP em comunicações <i>multicast</i> [Sousa97].	169
Figura 52 - Formato de um pacote RTP [Sousa97].	169
Figura 53 - Formato de um pacote RTCP [Sousa97].	173

Agradecimentos

As primeiras palavras de agradecimento são dirigidas aos meus pais que sempre me motivaram para atingir os meus objectivos, não recompensando-me materialmente quando os atingia mas retribuindo-me em confiança e em outros sentimentos nobres que edificam a personalidade e sustentam um rumo de vida.

Aos meus irmãos que aguentam os meus devaneios nestas matérias, que me apoiam incondicionalmente sempre que são chamados a essa responsabilidade e que compõe, como pilares centrais, a estrutura familiar que me tem servido de base para alcançar todos os logros do passado, e certamente tornarão acessíveis as metas futuras.

À minha esposa Sandra, que sempre esteve ao meu lado nos melhores e piores momentos, contribuindo incomensuravelmente para que este trabalho vingasse.

Ao meu orientador, Doutor Paulo Carvalho, pela forma empenhada e equilibrada como soube orientar este trabalho, primando sempre pela qualidade, correcção e pelo bom ambiente de trabalho.

Aos meus amigos, que foram pacientes e colaboradores em tudo o que estava ao seu alcance, e que muitas vezes me ajudaram a abstrair deste trabalho, algo que se justifica também como forma de a ele voltar com ânimo redobrado.

Aos meus companheiros de trabalho que me facultaram todos os meios de que dispunham e se disponibilizaram e flexibilizaram no sentido de me libertar para poder dedicar-me à investigação e feitura desta dissertação.

Ao Dr. Nuno Rodrigues, meu colega de mestrado, a preparar tese, também na área das comunicações, que me ajudou na configuração do equipamento de *routing* do IPB para aceder à rede Mbone, que em Portugal está sob a jurisdição da FCCN, Fundação para a Computação Científica Nacional, e que sempre se demonstrou adjuvante, motivador e gerador de confiança.

À direcção da Escola Superior Agrária, pela facultação dos meios necessários.

Ao técnico Nuno Carvalho, funcionário do CIESA (Centro Informático da Escola Superior Agrária de Bragança), pela sua preciosa colaboração, sobretudo na fase de testes.

"the Internet is its own revolution"

Anthony-Michael Rutkowski, former Executive Director of the Internet Society

Capítulo 1

Introdução

A Internet está, cada vez mais, a tornar-se vítima do seu próprio sucesso. Sem precedentes no tocante ao crescimento, sem paralelo na sua heterogeneidade e imprevisível ou mesmo caótica no comportamento do seu tráfego.

À medida que a Internet cresce, novas soluções são necessárias para prover um elevado número de utilizadores de acesso à informação de forma eficiente, concorrente e *on-demand* [Callahan96]. No modelo cliente-servidor usado na WWW (*World Wide Web*), os serviços de informação estão centralizados e tornam-se "gargalos" sob forte procura, sobretudo quando a resposta passa por *unicast*¹.

Por muita evolução nas tecnologias de interligação que se verifique, o galopante número de utilizadores e a procura de aplicações cada vez mais hipermédia, tempo real,

¹ Comunicação ponto-a-ponto.

e com características de usabilidade e interacção cada vez mais pró-humanas, esgotam ou dizimam os recursos de largura de banda por abundantes que possam ser.

A solução poderá não passar por tentar satisfazer as necessidades previsivelmente intermináveis de largura de banda, que como sabemos, não crescerão com a mesma desmesura com que cresce a sua exigência, mas sim otimizar a sua utilização, evitando redundâncias de tráfego ou outras formas de desperdício que ocorram na rede e que são características de todos os sistemas que crescem vertiginosa e desgovernadamente.

Denomina-se *multicasting* ou difusão selectiva [Deering89, Deering91], a metodologia de distribuição eficiente de informação pela rede global, selectivamente aos múltiplos receptores que a solicitaram. A tecnologia *multicast* não é suportada nativamente em toda a Internet, pelo que uma rede virtual sobre a Internet, constituída por "ilhas *multicast*" e túneis que as interligam, tem servido de plataforma experimental, trata-se do Mbone (*Multicast Backbone*) [Eriksson94].

O uso da rede MBone proporciona economia na utilização da largura de banda e processamento das máquinas envolvidas, já que um único datagrama de difusão selectiva pode atingir diversas estações numa mesma sub-rede sem a necessidade de ser replicado. No Mbone actual usam-se maioritariamente, aplicações para comunicação em grupo pertencentes à categoria das aplicações tolerantes, pois podem tolerar falhas de QoS² dentro de certos limites.

A maioria das aplicações *real-time* tolerantes não operam em simbiose com os meios que as suportam (rede e processamento) [Gopalan99], de facto não existe coordenação entre as aplicações, sistema operativo e serviços de rede, para que as aplicações recebam os meios essenciais para o seu eficaz funcionamento. Por outro lado, a infra-estrutura de suporte não fornece garantias de sustentabilidade dos recursos disponíveis para as aplicações, sobretudo ao nível da rede, onde as variações de QoS são aleatórias [Paxson97], nem é adjuvada na sua função pela falta de comportamento "consciente" das aplicações. Contudo, é possível utilizar proactivamente estas aplicações implementando uma arquitectura de QoS integrada, que inclua mecanismos de adaptação para as aplicações [Bhatti99], sobretudo para áudio e vídeo interactivo.

Num sistema (leia-se Internet) onde impera a heterogeneidade, com variações de desempenho presentes, as múltiplas aplicações partilham em competição, um limitado conjunto de recursos desse sistema. Desejavelmente, as aplicações cada vez mais

² Qualidade de Serviço

complexas, deveriam adaptar-se de forma dinâmica às condições existentes [Li98, Casimiro01, Balan02], auto-regulando os seus débitos e créditos de comunicação e processamento. Por outro lado, os mecanismos de adaptação existentes nas aplicações, não colectam todos os dados do sistema para "perceberem" a justiça dos recursos atribuídos, nem os sistemas operativos dominam a semântica particular das aplicações para poderem ser "justos".

Diversas aplicações *multicast* [Quinn01] estão disponíveis e visam oferecer, de forma otimizada, comunicação em grupo [Ellis91], num meio hostil que é o da Internet, onde o tráfego é entregue em *best-effort*³ e sofre de várias patologias que lesam a QoS [Wang01]. Contudo, ao nível aplicacional, e do ponto de vista do utilizador comum, não é imediato nem trivial efectuar uma parametrização da aplicação de modo a conseguir um desempenho óptimo da mesma. Acresce ainda, o facto dessa (re)configuração só acontecer como remédio e não como profilaxia, normalmente após uma situação de crise, por exemplo, a rede congestionou e os media não chegam em condições satisfatórias.

Pretende-se com este trabalho contribuir na investigação e desenvolvimento da adaptação de QoS das aplicações no âmbito da rede Mbone, integrando aplicações de tempo real tolerantes para conferência multimédia interactiva, que sirvam de suporte a um sistema de ensino à distância baseado em serviço WWW.

Um sistema de conferência multimédia lida normalmente com várias fontes de media em paralelo, e a cada fonte estão associados parâmetros que descrevem a qualidade percebida pelos utilizadores. Por exemplo, no vídeo percebe-se a definição e a taxa de cadência das imagens, no áudio percebe-se a fidelidade e o atraso. Para negociar uma QoS aceitável (que não comprometa a eficácia da comunicação), será necessário compreender os parâmetros de qualidade percebidos pelo utilizador bem como os parâmetros de QoS que a rede e a unidade de processamento terão de fornecer para satisfazer o sucesso e sustentabilidade da sessão. Assim, torna-se necessário determinar um perfil de QoS [Chatterjee97, Kassler00] para cada utilizador, através de um conjunto de regras que determinará o comportamento do sistema adaptativo perante as necessidades do utilizador e os recursos disponíveis.

A forma de adaptação utilizada neste trabalho está baseada na determinação de um perfil de QoS para cada novo membro emissor do grupo *multicast*.

³ Sem garantias de qualidade de serviço, vulgarmente, pelo melhor esforço.

Cada utilizador, dando como exemplo a transmissão de *video-on-demand* (VOD), terá um perfil de QoS da aplicação específico (ver **Figura 1**), que resultará da solução de compromisso entre as suas necessidades reais e as capacidades efectivas dos sistemas que suportam a sua presença na Internet. Para um desempenho óptimo, esse perfil de QoS da aplicação terá ainda de ser complementado com um perfil de QoS da rede, algo que na Internet actual se torna um desafio.

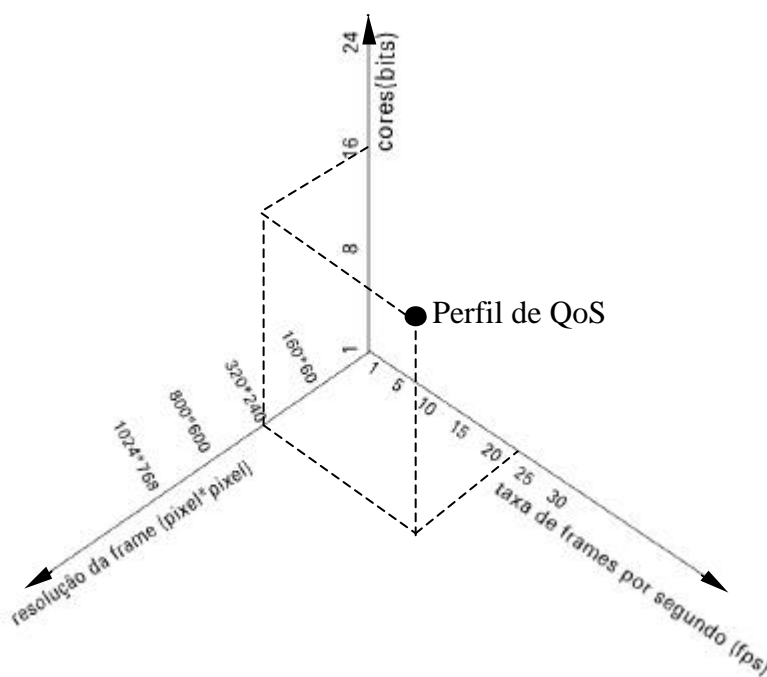


Figura 1 - Perfil de QoS para um utilizador de *video-on-demand* (VOD)

Diz-se por vezes, que transporte e comunicação estão em corrida, aquela que ganhar tornará a outra obsoleta. Encontros virtuais podem ser usados para ensino à distância, médicos especialistas diagnosticando/operando à distância, tele-trabalho, e um sem fim de aplicações.

O Mbone, tal como a própria Internet, cresceu sem a planificação de uma autoridade central, e como tal, partilha muitas das forças e fraquezas da grande Internet.

Se pensarmos que a Internet é um grande computador à escala mundial, então a evolução desse computador sofrerá um avanço extraordinário, não quando substituirmos dezenas, centenas ou mesmo milhares dos seus componentes de processamento, por sistemas com maior capacidade computacional, mas quando melhorarmos a forma como comunicam entre eles. Introduzir-se-ia uma melhoria universal, que afecta toda a rede e todas as comunicações. O preço da revolução é o empenho na I&D.

1.1. Redes de computadores

Durante as duas primeiras décadas de existência, os sistemas informáticos eram altamente centralizados [Tanenbaum96], usualmente restritos a uma sala dedicada a albergá-los. Em inúmeros casos esta sala era limitada por paredes de vidro, através das quais os visitantes podiam atravessar sensorialmente para o mundo das maravilhas tecnológicas presente no cubículo.

Desde meados dos anos 80 um trabalho considerável tem sido desenvolvido para alcançar um ambiente de interligação publicitado como *gigabit networking* [Partridge98], apurando a tecnologia de processamento e comunicações de modo a sustentar taxas de transmissão da ordem do *gigabit* por segundo. Este esforço produziu resultados tangíveis: redes protótipo, novas arquitecturas protocolares e novos caminhos de investigação, levando ao incremento do optimismo face ao futuro das tecnologias da informação bem como, fomentando o financiamento por parte da indústria e dos governos para este tipo de investigação e desenvolvimento.

O termo rede de computadores pretende significar, uma colecção de computadores autónomos interligados [Tanenbaum96]. O factor de interligação deverá necessariamente proporcionar intercâmbio de informação. O requisito de autonomia prende-se com o facto de excluir dessa definição sistemas claramente mestre/escravo, se um computador pode impor o início de interligação, o término, ou controlar outro, a autonomia não é total.

1.1.1. A adequação da arquitectura da Internet ao transporte *multicast*

A família de protocolos TCP/IP, desenvolvidos por uma equipa de investigadores nos anos 70, continua a suportar o *internetworking* na Internet, conectando milhões de computadores e utilizadores, e incorporando uma tremenda variedade de diferentes tecnologias de rede. O IP, descrito no RFC 791 pode ser considerado o coração dos protocolos da Internet, sendo responsável por duas funções principais: entregar pacotes em modalidade *best-effort* e providenciar fragmentação e reagrupamento de datagramas para interligar pontos com diferentes MTU (*Maximum-Transmission Unit*).

O IP também suporta endereços especiais, para *broadcasting* (enviar um datagrama para todas as estações na rede ou na sub-rede), publicado no RFC 919, e

multicasting, difusão selectiva (enviar uma cópia do datagrama para *hosts* seleccionados em qualquer ponto da rede), descrita no RFC 1112.

As limitações da família protocolar IP para sessões *multicast*, sobretudo quanto à escalabilidade dos grupos, já foram alvo de estudo apurado, pelo grupo de trabalho *Large-Scale Multicast Applications* (LSMA) e encontram-se publicadas no RFC 2502

Algumas aplicações abdicam da fiabilidade para elevar a capacidade. Utilizam serviços não orientados à conexão para alcançar transferências de informação a taxas inflacionadas, especialmente se podem tolerar perdas de informação. Estas aplicações sequenciam a informação e permitem sincronização entre o emissor e o receptor no nível de aplicação. Exemplificar estas aplicações passa certamente pela categoria das aplicações *real-time*, enviar/receber áudio ou vídeo, etc.

Soa algo contraditório, mas algumas aplicações não seriam confiáveis se utilizassem um serviço fiável. Por exemplo, uma aplicação de transmissão em tempo real de áudio e vídeo não pode ambicionar que informação transviada seja retransmitida, ou permitir que os mecanismos de controlo de fluxo, de algum modo descontinuem ou interrompam o fluxo natural da informação. Assim, os serviços de transporte que melhor respondem a este tipo de aplicações dos tempos modernos, estão a ser reaproveitados e reinventados.

Na pilha de protocolos TCP/IP, os serviços de transporte, são da responsabilidade de dois protocolos primários, o TCP⁴ (RFC 793) e o UDP⁵ (RFC 768), fornecendo respectivamente, serviços orientados à conexão e não orientados à conexão.

Em *multicast*, não se utiliza o TCP por ser um protocolo ponto-a-ponto, logo não orientado à comunicação em grupo. Algumas aplicações necessitam comunicar com várias estações simultaneamente usando um único endereço. Os protocolos orientados à conexão não conseguem fornecer este serviço [QuinnShute98]. Normalmente, utiliza-se um protocolo chamado RTP (*Real-Time Transport Protocol*), descrito no RFC 1889, (ver Anexo I), que, correndo sobre UDP, disponibiliza um mecanismo sequenciado de entrega de datagramas, sem fazer quaisquer verificações acerca do sucesso dessa entrega ou do controlo de fluxo. A ideia é que as aplicações gerem tantos dados quanto possível "*on-the-fly*" e que os seus clientes se consigam adaptar a vários níveis de atrasos na rede. Algumas aplicações anteriores ao aparecimento do RTP utilizam métodos de transporte baseados em pacotes do tipo *multicast* UDP.

⁴ *Transmission Control Protocol*

⁵ *User Datagram Protocol*

Uma vez que o serviço de datagramas é não orientado à conexão (i.e. não "amarrado" a uma única estação remota), é também capaz de emitir em *multicasting* (difusão selectiva) ou *broadcasting* (difusão).

1.1.2. Problemas do teletráfego

Cada pacote na rede compete com todos os outros - uma conexão que usa um caminho "desimpedido" pode obter a totalidade da largura de banda do caminho e transferir os seus dados com celeridade. Se muitas conexões competem ao longo de um caminho, então cada uma receberá uma porção (talvez injusta), da largura de banda disponível [Hein97].

A comutação de pacotes recolhe enorme robustez: possibilita que a rede encaminhe transparentemente no decurso de falhas no *router* ou na ligação, sem perturbar as conexões activas. Os *routers* não têm qualquer problema em aceitar o tráfego reencaminhado porque, tanto como lhes cabe "dizer", não é de forma alguma reposição - não possuem qualquer noção do tráfego "corrente". Contudo, as ligações podem tornar-se sobrecarregadas, porque a taxa de chegada dos pacotes para transmissão excede a capacidade da ligação. Tais pacotes irão ser armazenados em filas de espera (*buffers*) aguardando transmissão, mas se a taxa excessiva for sustentada - condição designada por congestão - então em situação limite os *buffers* nos *routers* irão encher-se e alguns pacotes terão de ser descartados.

Para assegurar que as fontes se comportam apropriadamente na presença de congestão na rede, os protocolos usados para transmitir informação na Internet, incluem mecanismos de controlo de congestão fim-a-fim, que decrementam automaticamente a taxa a que tal informação é transmitida quando a congestão é detectada. Uma consequência importante da utilização do controlo de congestão reside no facto de que, o tráfego na rede é formatado pelas condições que cada conexão encontrou no seu passado. Como tal, o tráfego da Internet inclui um mecanismo básico que introduz significantes e complicadas correlações ao longo do tempo, bem como complexas interacções abrangendo as conexões activas.

Estudos sobre o comportamento do tráfego nas redes demonstram que o tráfego é altamente variável ou muito *bursty* [Leland91, Paxson95, Carvalho96]. Isto significa

que não chega a uma taxa "bem comportada", mas por oposição em arranques e paragens aleatórias com picos pelo meio.

1.1.3. Aplicações que dependem da QoS das redes

A utilidade das aplicações de comunicação em grupo [Ellis91] depende decisivamente das condições de interligação. Existem, nessa óptica, duas categorias de aplicações, as não tolerantes e as tolerantes a falhas [Chawathe98]. Primeiro, se a aplicação apresentar criticidade de missão estritamente dependente de requisitos de comunicação em tempo real, então, uma vez violada a provisão de QoS, essa aplicação pode falhar. Para estas aplicações *real-time*, as garantias de qualidade de serviço são mais importantes que as adaptações. Na segunda categoria, consideram-se as aplicações flexíveis [Li98], como sendo aquelas que requerem um determinado nível de QoS, mas cada parâmetro de QoS pode conter uma certa amplitude admissível, sem por isso inutilizar a aplicação. Nestas aplicações, a adaptação pode desempenhar um papel importante quando as garantias de QoS estável e uniforme não são possíveis (Internet), existem limitações físicas, custos proibitivos ou ainda, quando seja impossível prever os recursos necessários, de tal forma que impossibilite a reserva adequada.

As aplicações flexíveis apresentam as seguintes propriedades:

- podem tolerar escassez de recursos até um certo limite inferior, mas funcionam melhor em condições de abundância;
- podem sacrificar alguns parâmetros de qualidade não críticos (*e.g.*, qualidade da imagem) em favor de preservar a qualidade para os parâmetros críticos (*e.g.*, inteligibilidade do discurso).

Em condições de variação da disponibilidade de recursos é preferível assegurar a qualidade para os parâmetros críticos.

No contexto da educação à distância, os pontos chave são o conhecimento e a comunicação, que ligam o instrutor e os estudantes [White99]. Este trabalho, tomando como exemplo aplicacional o ensino à distância, ocupa-se da parte da comunicação, fornecendo um conjunto flexível de canais de comunicação válidos e com disponibilidade. Entende-se que as aplicações fundamentais para o processo de ensino à distância, se inserem na categoria das aplicações tolerantes a falhas e como tal, podem

beneficiar com a adaptação, sobretudo as mais exigentes em termos de recursos, i.e., a transmissão interactiva de vídeo e áudio.

1.1.4. Nova concepção de redes

A investigação tradicional em arquitecturas de computadores focou-se primariamente na concepção e desempenho de máquinas individualmente consideradas. Com a crescente importância da tecnologia de rede e dos interfaces multimédia, as arquitecturas do futuro irão ultrapassar as fronteiras da máquina e será necessário fornecer suporte adequado a operações com tempo crítico.

O objectivo da concepção de redes, deverá ser, maximizar o desempenho das aplicações residentes [Shenker95]. Pois bem, as aplicações residentes estão a mudar, *Telnet* (terminal remoto) e *FTP* (transferência de ficheiros) estão a dar lugar às conferências multimédia.

A informação multimédia vê a sua presença incrementada no tráfego que actualmente circula nas redes, a tendência é no sentido de reforçar essa presença. Visando a adequação às necessidades e às funcionalidades emergentes, as redes deverão evoluir no sentido de responder física e logicamente aos seguintes requisitos [Schmidt99]:

- **Garantias de QoS**

A maioria das redes actuais fornece serviço do tipo melhor-esforço, mas a informação multimédia possui constrangimentos temporais que devem ser assegurados para garantir a apresentação apropriada da comunicação. Para satisfazer esta exigência, a rede deve possuir a capacidade de transmitir tráfego sensível ao tempo com uma probabilidade de sucesso razoável.

- **Elevada Largura de Banda**

Uma vez assumindo que diversas *streams* de media serão utilizadas pelas aplicações específicas, o impacto na largura de banda da rede será substancial. Adicionalmente, os requisitos de largura de banda para o servidor e seus clientes podem ser não uniformes, uma vez que cada cliente pode apenas receber um sub-conjunto de *streams* transmitidas pelo servidor.

- **Baixa Latência**

Atendendo a que a informação multimédia se caracteriza por ser temporalmente sensível, as modificações a aplicar interactivamente ao comportamento da rede, terão de ser lesadas pela latência, como tal, o requisito de latência reduzida é de suma importância sempre que intrinsecamente à rede são necessárias as ditas interacções.

- ***Multicast***

A difusão selectiva ou *multicast* impõe-se pela natureza da informação a transmitir, uma vez que um servidor transmitirá os conjuntos de *streams* de media para, potencialmente, um grande grupo de receptores (clientes), a utilização eficiente dos recursos da rede requer suporte integral a *multicast*. O *overhead* da transmissão reduz-se se aplicada a técnica de difusão selectiva.

- **Sinalização Rápida**

Sendo necessárias modificações dinâmicas aos conjuntos de *streams* de media que um cliente recebe, utilizando gestão de conexão de tipo *fine-grain*, e atendendo a que estas devem ser efectuadas de forma célere não conflituando com as necessidades ditadas pela informação sensível ao atraso, torna-se imprescindível um mecanismo de sinalização rápida.

- **Gestão Integrada de Tráfego**

Constatada a necessidade de transmitir diferentes tipos de media (áudio, vídeo) no tocante às aplicações multimédia, não se pode olvidar que outras aplicações com requisitos funcionais diferentes, também se servem da mesma infra-estrutura de rede. Assim, torna-se impreterível alocar e integrar diferentes géneros de tráfego, a gestão integrada de múltiplos tipos de tráfego parece ser a solução e deve ser a própria rede a fornecer esse serviço.

1.2. Enquadramento do trabalho

A Internet é recorrentemente apresentada como panaceia para a sociedade e em particular para a educação. Nunca antes, os agentes educativos estiveram numa posição tão favorável para capitalizar os avanços tecnológicos na nobre função de ensinar. Contudo, é preciso entender a melhor forma de explorar este potencial.

Uma forma pouco afortunada, de fazer a transição da aprendizagem tradicional para a versão digital, tem sido traduzir/converter o material existente para um meio alternativo, com ganhos muito limitados. Por exemplo, o facto do aluno possuir os apontamentos editados na *web*, representa um avanço tímido e pouco revolucionário, a tecnologia de suporte à educação remota pode ser encarada com certa vanidade se destituída de vocação humanizada. Em [Lawhead97] sumariaram-se os aspectos mais e menos apropriados da relação entre a educação à distância e a Internet.

As universidades, cada vez mais profissionalizadas e activas, procuram sempre acompanhar as últimas tecnologias, nomeadamente a Internet e a "webização". Cedo perceberam as vantagens da globalização da sua presença, da acessibilidade da sua informação [Steeple96], enquanto antecipavam (realisticamente ou não) soluções tecnológicas para incrementar os rácios estudantes/staff, diminuir despesas e melhorar as experiências curriculares dos alunos [Reinhardt95, Skillicorn96, ThomasCarswell96].

É preciso constituir uma comunidade de aprendizagem baseada em tecnologias de informação e interligação, urge repensar criticamente as nossas motivações por forma a não orientar essa edificação na usurpação de modelos tradicionais, mas sim, no alargamento em coexistência e na inovação.

É neste enquadramento que surge este trabalho, procurando amearhar um conjunto de directivas e *framework* que desemboquem num sistema integrado de ensino à distância baseado no melhor (conceptualmente) que a tecnologia pode oferecer.

1.2.1. Objectivos

Este trabalho insere-se no âmbito das aplicações orientadas à comunicação em grupo, tomando como instanciação o ensino à distância. Através da integração de aplicações *multicast* ou de difusão selectiva, de domínio público, para distribuição de áudio e vídeo, pretende-se melhorar a forma como uma aplicação integrada opera e utiliza as novas infra-estruturas de rede. Espera-se que tal aplicação, com recurso a uma camada de *middleware*, seja capaz de se adaptar em função das características dos sistemas finais e da plataforma de comunicação disponível. Assim, definem-se como objectivos:

- Definição e implementação de um *framework* capaz de integrar diferentes media com o objectivo de alterar "conscientemente" o comportamento das aplicações tolerantes, em função das características dos sistemas clientes e das condições de transmissão.
- Teste do *framework* desenvolvendo um sistema de educação à distância (EAD), com base em aplicações para conferência multimédia *multicast* (CMM), integradas numa interface *web* e recorrendo a uma camada de *middleware* para adaptação de QoS.
- Garantir um comportamento proactivo a favor da preservação de QoS da rede, sobretudo da largura de banda e, conseqüentemente beneficiar a sustentabilidade e escalabilidade das comunicações.
- Criação de uma interface *web*, algo equiparável a uma unidade de ensino virtual, onde se processa o acesso simplificado aos diferentes serviços *multicast* e calendarização de sessões, compostas pelas múltiplas aplicações *multicast* disponibilizadas.
- Teste de interoperabilidade entre participantes distribuídos por ambientes heterogéneos.

1.3. Organização da dissertação

A dissertação está estruturada em sete capítulos.

Neste capítulo, descreveu-se introdutoriamente a problemática e contextualizou-se o trabalho e a sua motivação. Completa-se o capítulo expondo os objectivos e as contribuições que se esperam alcançar.

No segundo capítulo procura-se estudar o estado da arte do EAD, baseado em serviço WWW, nos trabalhos relacionados efectuados no passado recente. Reunem-se também, alguns exemplos de estratégias para preservar a QoS em sistemas de conferência multimédia, quer por adaptabilidade ou conteúdos *web* dinâmicos. Visa-se assim integrar e justificar este trabalho com base na produção científica da área.

No capítulo três descreve-se a tecnologia *multicast* IP. Expõem-se os fundamentos técnicos do seu funcionamento e as contribuições para o aproveitamento da infraestrutura de rede, bem como, para o desenvolvimento das aplicações de CMM presentes e futuras. Este capítulo deverá fornecer a teoria acerca da tecnologia de comunicações que serve de base ao trabalho e deverá ser considerado conjuntamente com o Anexo I, que reúne os protocolos *multicast* mais relevantes no contexto desta dissertação.

Ainda no capítulo três, aborda-se a rede Mbone, a rede virtual sobre a Internet que serve de suporte às aplicações *multicast*, inclusivamente o seu desenvolvimento em Portugal. Avançam-se também as inovações introduzidas no IPv6 para *multicasting*.

No capítulo quatro, destinado às aplicações *multicast*, deverá ficar claro como se integraram no sistema de EAD, as aplicações *multicast* de domínio público orientadas à CMM. Descrevem-se essas aplicações, com ênfase colocada na forma de melhorar a sua qualidade, provendo a adaptabilidade e beneficiando a sua usabilidade.

Executar as aplicações de forma adaptada, preservando os recursos para os parâmetros críticos, depende da possibilidade de pré-parametrizar as aplicações por linha de comandos, numa sintaxe que inclua as opções. A escolha das aplicações foi, mormente, motivada por este factor. Apresentam-se também, para as aplicações de áudio e vídeo, aquelas que mereceram a adaptabilidade, a sinopse e as opções de pré-configuração que aceitam.

No quinto capítulo faz-se uma avaliação da influência da QoS no sistema de EAD baseado em tecnologia *multicast*. Explicam-se os parâmetros da rede e da aplicação que contribuem para a degradação ou obtenção da QoS. Define-se um modelo

de QoS para aplicar ao sistema de EAD, baseado nas propostas do IETF⁶, as arquiteturas *IntServ* e *DiffServ*.

No capítulo seis apresenta-se o trabalho desenvolvido.

Principia-se pela descrição da plataforma experimental utilizada. Segue-se-lhe a exposição da arquitectura do modelo implementado para garantir o funcionamento adaptativo do sistema de EAD.

Apresenta-se o *site* que funciona como universidade virtual e todos os serviços que disponibiliza. Cabe aqui também, descrever a base de dados associada, para guardar os dados dos e-alunos e as características e calendarização das sessões previstas.

Seguem-se os pormenores técnicos de implementação, sobretudo ao nível do *middleware* (*applets* e *javascript*) que assegura a adaptabilidade das aplicações às condições verificadas na rede e no sistema final. Descrevem-se também, os aspectos de segurança para os *applets* que têm de executar funções que requerem privilégios adicionais na máquina cliente (aluno).

Relativamente à adaptabilidade, apresenta-se a fórmula, com as variáveis e o algoritmo envolvente, que será computada para produzir o modo de adaptação das aplicações de áudio e vídeo. Cada modo de adaptação, num total de cinco, consiste num conjunto de pré-configurações a transmitir às aplicações de áudio e vídeo. Os resultados obtidos experimentalmente e a sua discussão encerrarão o sexto capítulo.

No capítulo sete deverão emergir os resultados substantivos bem como a sua discussão, e reformulação se for caso disso. Neste capítulo importa também, tirar conclusões acerca dos métodos e tecnologias utilizadas na consecução do trabalho prático, na óptica da comunicação em grupo.

Analisa-se as contribuições para o desenvolvimento do ensino em geral, e em particular para a modalidade de EAD.

Finalmente, serão aventados possíveis desenvolvimentos futuros.

⁶ Internet Engineering Task Force - <http://www.ietf.org>

Capítulo 2

Trabalho Relacionado

Importa, neste capítulo, apresentar em traços gerais o estado da arte do ensino à distância baseado em tecnologia da Internet, bem como as suas derivações, enquadrando o âmbito da aplicação que se pretende desenvolver, assente em ferramentas *multicast*. Além do EAD, carece relevar experiências em adaptabilidade por introdução de *middleware* e formas de garantir QoS para conferências multimédia *multicast* (CMM).

A natureza multidisciplinar do trabalho a desenvolver, abrange várias áreas científicas: engenharia informática e comunicações por computador, considerando a QoS em aplicações e redes informáticas na temática *multicast*; pedagogia, na vertente do ensino à distância; e multimédia, incidindo sobre a produção de conteúdos didáticos. O sistema de EAD que se pretende implementar, sem prejuízo para a inovação almejada, não ignorará certamente, os ensinamentos que as experiências nestas áreas nos legaram.

2.1. Algumas considerações sobre o ensino à distância

A atitude dos actores do processo formativo é condicionada pela aceleração tecnológica e comunicacional da aldeia tribal electrónica em que estão integrados. Se muitos ainda discutem benefícios e malefícios da televisão tradicional, só alguns se precatarem de que para lá dos mass media do presente, tanto a telemática como a hipermédia já estão implantadas no terreno educativo e questionam os próprios fundamentos do processo de ensino/aprendizagem, problematizando o modo como docentes, discentes e instituições desempenham os novos papéis que lhes estão consignados.

Educação à distância define o processo de prover uma ligação entre instruídos e a informação remota, fazendo uso, para isso, de alguma tecnologia [Yoakam98].

A história da educação à distância é uma sequência de novas ideias e tecnologias – sempre contrabalançadas por uma resistência a mudanças – e repeti-la aqui seria uma longa história, cujos primórdios remontam às cartas de Platão e às epístolas de São Paulo.

Figuras de proa do panorama educacional norte-americano reconheceram que computadores e telecomunicações podem ajudar a resolver alguns problemas na educação superior [Douglas93].

"A indústria educativa assistirá a uma revolução à medida que o crescimento de professores virtuais e meios de ensino se multiplicam pela super-autoestrada da informação mundial, com milhões de consumidores de conhecimento interligados, cada um habilitado a apreender...o curso específico ou as aulas desejadas, usando apenas uma fracção da energia e dispêndio daquilo que significa o ensino *face-to-face* tradicional. A completa noção de Universidade está em causa, as transformações ocorreram para acomodar alunos e conteúdos da nova geração [Abdel-Wahab93]."

2.1.1. Contemporização do EAD

Desde os anos 50 que os investigadores procuram compreender a forma como se desenrola o processo cognitivo, segundo [Cronbach57], factores como as emoções, empenho, esforço, sentimentos, desejos, etc., bem como outros de cariz psico-socio-fisiológico, fazem de cada indivíduo um consumidor de conhecimento adaptativo.

Estudos posteriores provaram que as aptidões no tratamento de apreensão cognitiva eram dificilmente replicadas, no entanto descobriu-se que um desenho instrucional levado à excelência podia reduzir as diferenças inatas e ascender o aproveitamento [Bunderson70].

A cultura de comunicação digital está a forçar uma mudança do estilo da educação tradicional que ocorre no confinamento entre quatro paredes de uma instituição - uma herança dos tempos da Revolução Industrial - para um sistema de educação no estilo aberto, onde escolha e qualidade são essenciais. Segundo Alvin Toffler, esta última é fundamental, pois "os analfabetos de futuro não serão apenas os que não sabem ler e escrever, mas aqueles que não sabem aprender, desaprender e reaprender".

O fenómeno CBT (*Computer Based Training*) invadiu literalmente as publicações comerciais e científicas. Por exemplo, existem mais de duzentos artigos com o assunto "ensino à distância", publicados na biblioteca IEEE/IEE de 1997 a 1999 [IEEE]. Projectos como o IRI (*Interactive Remote Instruction*), descrito em [Maly94] foram percursos do ensino colaborativo baseado em infocomunicação interactiva de conteúdos multimédia.

Actualmente, os investigadores procuram conciliar a profícua aplicação da tecnologia existente, com o estado evolutivo/social da humanidade. Os ambientes de aprendizagem baseados em meios telemáticos e conteúdos digitais devem adaptar-se e responder a esta diferença natural, a concepção de novas metodologias de ensino não se deve afastar da personalização apesar do processo virtual [Martinez00].

Podemos hoje reconhecer sete modelos emergentes de organizações universitárias [Hanna98], são eles:

- Extensão das Universidades tradicionais
- Centradas no atendimento de adultos
- EAD (Ensino à Distância) - Tecnologias
- Universidades corporativas
- Alianças Universidades - Empresas
- Certificação por competência
- Multinacionais globais

Diante deste novo universo onde a educação está intimamente ligada às transformações científicas e tecnológicas, é necessário evitar o equívoco comum de muitos educadores, de introduzir tecnologia na educação sem nenhuma tentativa de modificar as formas nas quais a educação é organizada [Moore96].

2.1.2. Caracterização dos modelos de ensino

Podem-se dividir as diversas abordagens de aplicação da informática na educação em duas grandes correntes: a primeira, denominada *algorítmica*, caracteriza-se pela existência de uma sequência pré-definida de actividades que guia o aluno na aprendizagem, rumo a um objectivo. A segunda, denominada *heurística*, onde temos uma exploração de actividades que propiciem o desenvolvimento de habilidades, por meio da solução de problemas, criatividade, manipulação de objectos, etc., sempre sob o controlo do próprio aluno. No primeiro enfoque, o computador é visto como uma "máquina de ensinar" e no segundo, é visto como uma "ferramenta para aprender".

A comunicação mediada por computadores é o mais importante instrumento pedagógico da sociedade contemporânea, sobretudo associada às especificidades da educação à distância. Observa-se nesta perspectiva, a instalação global de sistemas de comunicação *online* aplicados ao desenvolvimento de processos não exclusivamente presenciais de educação. Esse consórcio entre as áreas da comunicação e da educação não é simples mas, superado o mito de que a escola tem necessariamente que se localizar num espaço físico específico, expandem-se em todo o mundo as experiências que aliam acções de interacção presencial aos processos favorecidos pelas tecnologias da informação.

O professor é e continuará a ser um elemento fundamental; entretanto, o seu trabalho precisa de ser modificado e aprimorado, no sentido de ele se tornar o orientador da aprendizagem dos seus alunos, levando-os a aprender a aprender. O tutor assume, neste momento, um novo papel onde ele será sempre necessário, não apenas como orientador pedagógico, mas também, ampliará o seu mercado de trabalho, uma vez que será ele o editor técnico do conteúdo das matérias *on-line* e *off-line*.

A educação à distância exige uma rigorosa capacitação do aluno que irá desenvolvê-la, como também um rigoroso preparo do material didáctico-pedagógico, escrito ou de outra natureza.

Segundo [Lévy96], "uma comunidade virtual pode, por exemplo, organizar-se sobre uma base de afinidades por intermédio de sistemas de comunicação telemáticos. Os seus membros estão reunidos pelos mesmos núcleos de interesse, pelos mesmos problemas: a geografia e contingente, não representam, nem um ponto de partida, nem uma coerção. Apesar de "não-presente", essa comunidade está repleta de paixões e de projectos, de conflitos e de amizades. Ela vive, por oposição a referência estável, em toda parte onde se encontrem os seus membros móveis... ou em parte alguma. A virtualização reinventa uma cultura nómada, não por um regresso ao Paleolítico nem às antigas civilizações de pastores, mas fazendo surgir um meio de interacções sociais onde as relações se reconfiguram com um mínimo de inércia."

2.1.3. Riscos de exploração do ensino à distância

Os riscos de exploração do ensino *on-line* têm sido assumidos um pouco por todos os agentes escolares, encarando o fenómeno como uma tentativa de renovação e um catalisador de novas metodologias de ensino/aprendizagem.

A superioridade advogada por este método sobre o ensino tradicional prende-se com vários pontos [Neumann98, Update99], a saber:

- reutilização do material pedagógico;
- flexibilidade de horários para os estudantes, que se caracterizam pela sua heterogeneidade no tocante às idades, objectivos, profissões e localização física;
- auto-procura e selecção de tópicos na ordem pretendida, favorecendo a conduta autodidacta;
- a associação da pedagogia e tecnologia tem revelado bons indicadores de motivação e confiança por parte dos alunos;
- democratização do acesso a conteúdos pedagógicos de qualidade;
- aproximação do momento da formação com a necessidade efectiva, *just-in-time* formativo;
- redução de custos materiais e imateriais, associados com matrículas, deslocações, tempo, mobiliário e as infra-estruturas;
- alargamento da audiência, globalização do acesso a ensino/aprendizagem, com escalabilidade beneficiada pela difusão selectiva ou *multicast*.

No reverso da medalha encontram-se pelo menos outros tantos argumentos contra a fragilidade do método, em [Neumann98] enumeram-se alguns:

- diminuição ou mesmo inexistência de interactividade entre o professor e o aluno no processo lectivo/aprendizagem, o que reconhecidamente afecta a eficácia na apreensão dos ensinamentos;
- para suplantar a falta de interactividade podem interpor-se videoconferências ou *chats*, no entanto as limitações sensoriais persistem. A falta de cultura organizativa nessas práticas provoca frequentemente situações de descoordenação interventiva;
- a escassez de largura de banda disponível nas redes das universidades e instituições perfila-se como um impedimento à realização eficaz de transmissões dispendiosas em termos de carga de rede, como o são as videoconferências, mesmo com a utilização de aplicações *multicast*;
- sendo conhecida a necessidade de fomentar o espírito colaborativo entre os alunos, o facto dos grupos estarem dispersos geograficamente limita de sobremaneira o crescimento/fortalecimento das relações inter-discentes;
- os meios computacionais bem como as infra-estruturas de rede que interligam o grupo nem sempre são equitativos na qualidade de serviço que fornecem, impedindo que alunos menos favorecidos tecnologicamente obtenham aproveitamento satisfatório;
- nesta modalidade de ensino, as experiências laboratoriais não surtem todo o efeito pretendido dada a ausência física dos alunos;
- a certificação da qualidade do ensino ministrado ainda não está suficientemente desenvolvida.

Algo de irrefutável, porém, são os ritmos elevadíssimos com que a Internet chega às Instituições de ensino, incluindo o seu cerne, a sala de aula. Obviamente, esta

pré-condição assegurada em boa percentagem é um factor aliciante para a investigação e desenvolvimento destas inovações pedagógicas.

Na **Figura 2** afere-se a quase plena implantação dos sistemas informáticos com ligação à Internet nas escolas de todos os graus de ensino. Os dados referem-se aos EUA, mas os restantes países desenvolvidos seguem-lhe o exemplo.

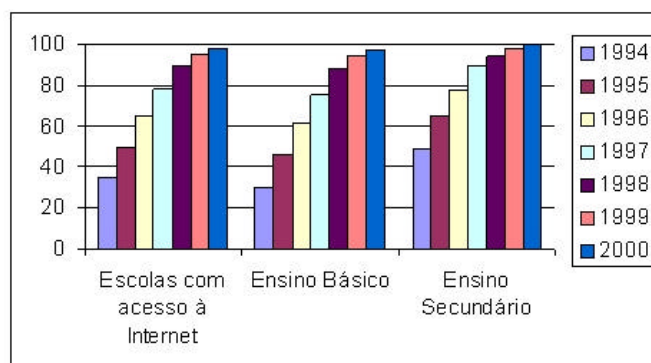


Figura 2 - Implantação de computadores com acesso à Internet nas Escolas Norte-Americanas

Fonte: US Department of Education Statistics 2000 - <http://nces.ed.gov>

Outro dado importante prende-se com o número de alunos por computador com acesso à Internet. Da mesma fonte, foi possível concluir que, gradualmente esse número se vai reduzindo desde as centenas, para as dezenas e recentemente para as unidades.

O *e-learning*, assim modernamente denominada esta tipologia/metodologia de ensino, tem reunidas um bom rol de condições adjuvantes. Caberá agora aos investigadores, fazer com que os sistemas a desenvolver não provoquem cisões no actual sistema de ensino mas, procurar colaborar e preencher as lacunas existentes de modo a poder avançar lado a lado com o sistema conservado numa atitude cooperante, sempre com o intuito de alcançar a excelência na pedagogia.

2.2. Modelos de ambientes de educação à distância existentes

Esta secção apresenta diversas experiências relacionadas com o desenvolvimento e implementação de ambientes educacionais, no contexto internacional, já que a nível nacional os projectos existentes ainda não atingiram maturidade.

Listam-se vários exemplos de soluções para fornecer serviços de educação à distância, mas apenas se descreverão pormenorizadamente os mais representativos. Os

critérios de selecção dos modelos educacionais foram simples; estudar aqueles que se posicionavam como líderes no seu sector de actuação (o comercial e o universitário), sendo condições necessárias estarem acessíveis via *web* e apostarem em ferramentas de conferência.

Em conformidade com os objectivos deste capítulo, interessa referenciar o desenvolvimento do sistema de EAD, em experiências já implantadas com sucesso.

Alguns trabalhos e ambientes educacionais existentes são apresentados na **Tabela 1**. Vários outros ambientes educacionais podem ser encontrados, sendo utilizados no processo de ensino, aprendizagem e treino das mais variadas áreas de conhecimento.

Ambiente	Organização	URL
Centra One	Centra Software, Inc.	http://www.centra.com
Virtual-U	Virtual Learning Environments Inc.	http://www.vlei.com/
Learning Space	Lotus Education of Lotus Institute	http://www.lotus.com/
Web Course in a Box	MadDuck Technologies	http://www.madduck.com/
TopClass	WBT Systems	http://www.wbt systems.com/
WebCT	University of British Columbia	http://www.webct.com/
Asymetrix	Librarian Asymetrix	http://www.asymetrix.com/
FirstClass	Classrooms SoftArc	http://www.softarc.com/
CourseInfo	Blackboard Inc.	http://www.blackboard.com/
ARIADNE	EPF Lausanne (EC DG XIII)	http://ariadne.unil.ch/tools/
CoMentor	Huddersfield University	http://comentor.hud.ac.uk
CoSE	Staffordshire University	http://www.staffs.ac.uk/cose
Learning Landscapes	Bangor University	http://toomol.bangor.ac.uk
HLM	George Mason University	http://cne.gmu.edu/hlmeter/

Tabela 1 - Exemplos de ambientes educacionais baseados em serviço WWW.

Seguidamente, são apresentados alguns ambientes educacionais mais detalhadamente. O *Centra One* é, provavelmente, o pacote de software comercial mais completo para ensino à distância e trabalho colaborativo, enquanto que o WebCT é orientado para o ensino superior, líder no seu sector, é uma plataforma que merece destaque pois tem sido utilizada com sucesso reconhecido na formação à distância.

2.2.1. O ambiente *Centra One*

Este software proporciona um ambiente completo, baseado em *web* e *open architecture* para distribuição de conhecimento e colaboração em tempo real. Num *framework* comum integram-se também ferramentas de autoria e gestão de conteúdos.

Este *framework*, virtualmente uma sala de aula, permite aos utilizadores, capturar, gerir e distribuir conhecimento a uma vasta audiência. Empresas como a BMW ou a McDonalds utilizam o *Centra One* para a formação contínua dos seus recursos humanos. O preço da aplicação completa, com 100 licenças é de 75000 euros.

As características seguintes ajudam a compreender as funcionalidades da plataforma aplicacional:

- interface integradora de todas as ferramentas disponíveis, facilitando a gestão dos conteúdos (ver **Figura 3**).
- disponibilização de conteúdos nas modalidades: ao vivo, interactiva e *self-service*. Possibilidade de gravação em ficheiros para consultas ulteriores;
- partilha de aplicações, slides e páginas *web*.
- videoconferência (*unicast*) e audioconferência, com VOIP (voice over IP).
- suporte para participação em 13 línguas diferentes;
- *Satellite Server*, um sistema que minimiza o impacto na largura de banda;
- encriptação de dados para comunicações seguras;
- *whiteboard* e *chat*;
- organização e calendarização de sessões, com possibilidade de integração com ferramentas de gestão de tarefas, e.g., *Microsoft Outlook*.

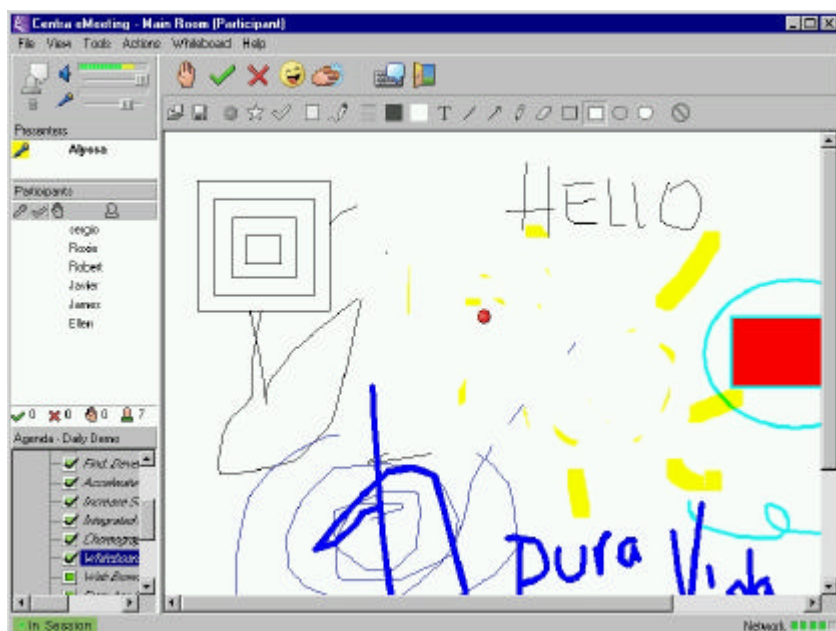


Figura 3 - Aspecto da interface do *Centra One*

Este *framework* será uma referência para o trabalho a desenvolver, apesar de, não incluir comunicações *multicast*, algo compreensível se atendermos à natureza

comercial da aplicação, o que limitaria a sua penetração no mercado, já que poucos ISPs encaminham tráfego *multicast*. Sem embargo, apresenta muitos pontos fortes que podem balizar este trabalho, comungando das directivas em mente para a implementação do sistema de EAD. Citam-se alguns pontos fortes: centrado em *web*, interface integradora, possibilidade de conferência multimédia e com soluções que garantam escalabilidade.

2.2.2. O ambiente WebCT

Desenvolvido pelo grupo de Murraw W. Goldberg, da *University of British Columbia*, o WebCT (*Web Course Tools*) [Goldberg96] fornece um conjunto de ferramentas que facilitam a criação de cursos educacionais baseados no ambiente WWW. Também pode ser utilizado como ferramenta complementar de um curso já existente, na disponibilização de material.

A principal vantagem associada ao WebCT está na possibilidade de se estabelecer um ambiente de ensino e aprendizagem integrado, contendo uma série de ferramentas educacionais tais como sistema de conferência, *chat*, correio electrónico, acompanhamento do aluno, suporte para projectos colaborativos, auto-avaliação, questionários, distribuição e controlo de notas, glossário, controlo de acesso, calendário do curso, geração automática de índices e pesquisa, entre outras.

Toda a interacção com o WebCT é baseada na WWW, usando portanto *software* multiplataforma. O acesso é igualmente aberto.

O WebCT é apresentado aos alunos, professores e outros utilizadores como um documento principal, a partir do qual se tem acesso aos tópicos dos cursos e demais ferramentas disponíveis. Nesse ambiente, o professor pode criar material didáctico e acompanhar o desempenho dos alunos.

A interface para autoria de cursos no WebCT contém opções para criar páginas (ou importar páginas de texto ou HTML existentes) e para incorporar ferramentas educacionais dentro das páginas.

Após a criação de uma página, o autor deve indicar a localização relativa dessa página no curso. A organização das páginas pode ser hierárquica, para acesso imediato a qualquer tópico, subtópico ou página individual; ou linear, para definir um caminho sequencial através do curso.

Além de ferramentas educacionais que auxiliam a aprendizagem, a comunicação e a colaboração, o WebCT também fornece um conjunto de ferramentas administrativas para auxiliar o autor no processo de gestão e melhoria contínua do curso.

Este ambiente tem os seus pontos fortes na forma, integrada e flexível, como se criam e adicionam os conteúdos que configuram um curso *on-line*, é muito completo ao nível da organização, fruto dos seus 20 anos de experiência. Como lacuna, poder-se-á apontar a falta de videoconferência, ficando a interactividade debilitada.

2.3. Adaptação de QoS das aplicações controlada por *middleware*

Em [Li98, Li00] apresentam-se soluções que apontam para a introdução de uma camada de *middleware*, que seria responsável por controlar as adaptações de QoS ao nível das aplicações. Pela assistência de uma arquitectura de *middleware*, as dinâmicas da aplicação e do sistema poderiam ser avaliadas para decidir quando, como e qual seria a extensão da adaptação a interpor.

Em [Li00] apresenta-se o sistema Agilos (*Agile QoS*), um *framework* baseado em *middleware*, que tomaria a responsabilidade de gerir a adjudicação dos recursos, de forma adaptada, às aplicações multimédia em tempo real distribuídas. Utilizando para tal, controlo dinâmico e a liberdade de configurar os parâmetros internos de cada aplicação tutelada.

A arquitectura básica do *Agilos* (ver **Figura 4**) compreende várias entidades em sequência lógica que consomem a adaptação de uma aplicação.

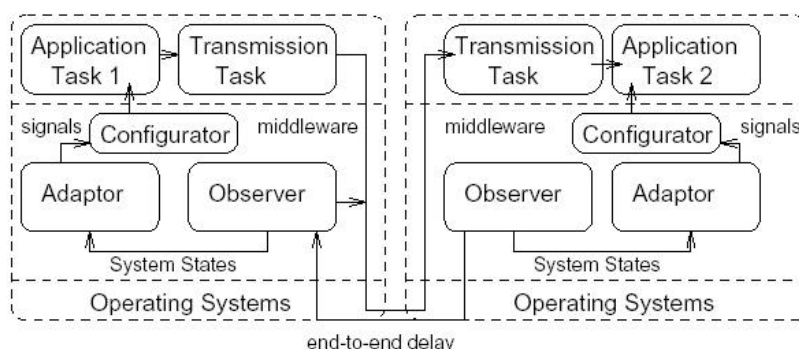


Figura 4 - Arquitectura do sistema *Agilos* [Li00].

Primeiro, o *Observer* recolhe os dados representativos do sistema para calcular a adaptação apropriada, depois o *Adaptator* computará esses dados (através de modelos

matemáticos), intentando a melhor e mais justa solução de adaptação, gerando uma solução de compromisso entre a QoS necessária e a QoS atingível. Por fim o *Configurator* transmite à aplicação o conjunto de configurações, que lhe providenciará o ajustamento que melhor servirá o sistema em geral.

Desenvolveu-se uma ferramenta desenhada e implementada com os propósitos de testar e validar o *Agilos*, o *Omnitrack*, uma aplicação de monitorização por vídeo, omnidireccional, que admite algumas configurações.

As adaptações que o *Omnitrack* sofrerá, devem priorizar os parâmetros críticos para a aplicação, neste caso importava garantir o êxito da monitorização pela precisão com que se acompanhava o objecto monitorizado. Os parâmetros que sofreram adaptação foram seis, que se enumeram a seguir:

- taxa de *frames* por segundo (fps);
- riqueza cromática das imagens (bits de cor);
- tamanho da imagem (pixel*pixel);
- compressão usada no *codec* (implica maior ou menor carga de CPU);
- número de janelas activas da aplicação *Omnitrack* (implica maior ou menor carga de CPU);
- tamanho da região transmitida (implica maior ou menor carga de CPU).

Os resultados foram animadores, demonstrando a eficácia da acção de controlo e adaptação, validando os pressupostos que motivaram o trabalho. O desempenho do *Agilos* revelou-se satisfatório para um número razoável de instanciações da aplicação *Omnitrack* activas, até 30. Quando esse número escala, multiplica-se o *overhead* introduzido, e os recursos de CPU praticamente esgotam-se. Neste cenário um sistema de gestão de recursos pouco pode fazer, pelo que a escalabilidade continua a ser um problema, mas agora mais afastado.

Como apreciação final, refira-se que o *Omnitrack* não é uma aplicação *multicast*, pelo que a gestão dos recursos, num sistema que se pretende funcional e não apenas para testes, como o sistema de EAD a desenvolver, deve principiar pela inclusão de aplicações que optimizam o consumo de recursos.

Este trabalho [Li00], é sem dúvida, aquele que mais se aproxima em termos técnicos, da solução que se pretende implementar para o sistema de EAD deste projecto de mestrado. Acrescente-se que à data da sua publicação, no decorrer do ano 2000, já muito trabalho deste projecto estava efectuado, no entanto foi com muita satisfação que

se constatou a similitude dos modelos e das arquiteturas idealizadas. Ainda assim, incorporaram-se alguns ensinamentos e ratificaram-se algumas ideias e conceitos.

2.4. Experiências de adaptabilidade e QoS em *multicast*

As aplicações *multicast* de domínio público são insensíveis a situações de congestão ou mesmo degradação das condições de emissão/recepção, tal comportamento é atentatório contra os recursos, sempre escassos em termos de serviço de transmissão, que partilham todos os utentes da Internet [Mahdavi97].

A aceitabilidade destas condições, está dependente, do grau de tolerância à falta de fiabilidade, que pode ocorrer nestas aplicações em situação de transmissão de media em tempo real. Vários protocolos que implementam soluções de fiabilidade em *multicast* estão desenvolvidos, a proeminência vai para o SRM (*Scalable Reliable Multicast*) descrito em [Floyd98].

O SRM baseia-se na retransmissão para suprir as perdas de pacotes sucedidas. Obviamente tal técnica, servindo muitas das aplicações *multicast*, não é adequada para o tipo de aplicações com que se lida neste trabalho, as de transmissão de media em tempo real. Pois, "um pacote atrasado equivale a um pacote perdido" e a retransmissão nada resolve, apenas complica. Por outro lado, outras equipas de investigação direccionaram o seu empenhamento para soluções que garantem fiabilidade pela introdução de elementos críticos redundantes, por exemplo múltiplos servidores em suplência. Esta técnica garante também benefícios ao nível da escalabilidade, já que no caso das solicitações aumentarem o trabalho de resposta pode ser partilhado.

Paradigmas como o SomeCast [Yoon00], talvez o estado da arte em matéria de transmissão *multicast* fiável e adaptativa com algumas garantias de QoS, apresentam soluções que generalizam soluções anteriores de optimização de transmissão *unicast* e *multicast*, fazendo emergir um novo modelo mais eficaz e versátil baseado no cliente, como é propósito deste trabalho também, como o provam os resultados experimentais divulgados no estudo conduzido por Yoon e Bestavros.

Outros enfoques sobre o mesmo problema proporcionaram novas soluções ao nível da aplicação, que também se relacionam com o âmbito deste trabalho, e vão no sentido de atender às definições/parametrizações das aplicações que lidam com tráfego em tempo real. A primeira a ser abordada, concentrou-se em estudar as implicações do

controlo da taxa de transmissão de imagens num sistema multimédia de ensino à distância, a segunda estuda como a monitorização das condições de transmissão, mormente largura de banda e *round-trip-time*, se pode aproveitar para adaptar e otimizar o funcionamento das aplicações envolvidas num sistema de EAD.

Provou-se em [Josifovski94], que controlando a taxa de imagens (*frame rate*) ajustando-a à carga da rede, uma forma de adaptabilidade, se obtêm resultados positivos ao nível da QoS. Neste estudo foram desenvolvidos mecanismos que regulam a taxa de imagens a transmitir, quer em sessões pedagógicas em directo ou pré-gravadas, e efectuaram-se verificações dos resultados experimentais donde se concluiu o seguinte:

- é preferível transmitir apenas a *frame-rate* garantida pela rede do que transmitir acima disso e as imagens só aparecerem parcialmente;
- os cálculos baseiam-se na teoria das probabilidades, foi desenvolvida uma fórmula para o efeito, que incorpora parâmetros como latência e previsões de tempo de entrega de pacotes, assim, se existir a probabilidade de uma *frame* chegar em mau estado, então essa *frame* é cancelada;
- a estabilidade da apresentação sai beneficiada pelo controlo da taxa de imagens transmitidas.

Em [Cheng99] o objectivo foi monitorar e prever a disponibilidade de largura de banda para os *sites* dos clientes, e com base nessa computação otimizar a distribuição de material pedagógico pela Internet. Desta forma procurava-se dar garantias de QoS, baseando as entregas em métodos de priorização, ordenando a informação com base nas medições efectuadas e comcomitantemente com as preferências manifestadas pelo utilizador. A implementação considera como parâmetros relevantes para a gerir a QoS, a qualidade da imagem JPEG, resolução e atraso.

Tal como acontece no trabalho aqui em desenvolvimento, o sistema que materializa o descrito no parágrafo anterior, também se baseia numa interface *web* que incorpora vários módulos em Java, CGI, Perl, etc., que resolvem problemas como os de monitorar e calcular a disponibilidade de largura de banda, gestão da QoS, conteúdos *web* dinâmicos, etc.

Por conteúdos *web* dinâmicos, deve entender-se a capacidade que o próprio serviço tem de se adaptar às condições efectivas, combinadas com as preferências do utilizador final (cliente), com o objectivo de lhe disponibilizar o material pedagógico, mormente material assíncrono neste estudo, de forma adequada às condições de largura

de banda verificadas. Por comparação com o trabalho desenvolvido e descrito nesta dissertação, a preocupação vai também para as condições do *hardware* do sistema informático do cliente (memória livre; ocupação do processador; por exemplo), que são obtidas transparentemente para o utilizador, mas sob a sua autorização.

Tal como acontecerá neste trabalho de mestrado, reporta-se que foi necessário implementar uma ferramenta para a monitorização da disponibilidade de largura de banda, que ausculta as condições cada 30 segundos em média, o que deixa antever que se o cliente fizer um *reload* da página poderá vê-la com melhores imagens, por exemplo, se entretanto as condições de transmissão melhorarem. No entanto o trabalho descrito no artigo [Cheng99] cingia-se a um ambiente *unicast*, com um único cliente, mas asseguram os seus autores, que pode ser generalizado para múltiplos clientes, enquanto que o nosso cenário é *multicast*.

2.5. Analogias e projecções

Depois de expostos alguns dos trabalhos relacionados, considerados de interesse para este projecto, pode-se considerar que o propósito deste trabalho comunga de todos eles, mas evidencia um "personalidade" própria que se passa a explicitar.

O objectivo magno será o de integrar num sistema, instanciado no EAD, várias componentes, que se provou experimentalmente pelos exemplos apresentados, serem funcionais e estarem na vanguarda da investigação: EAD via *web* (Mbone); utilização da tecnologia *multicast* nas comunicações e aplicações; e adaptação da QoS das aplicações por *middleware* multiplataforma (ver **Figura 5**).

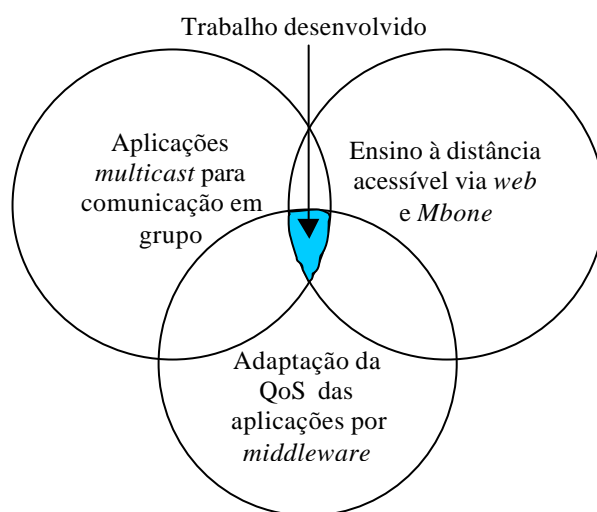


Figura 5 - Enquadramento do sistema desenvolvido face aos trabalhos relacionados.

Colmatar todas as exigências de QoS dos clientes é missão quase impossível, pela quase infinita gama de necessidades, muito específicas, se atendermos à aplicação de *groupware* que aborda este trabalho, o ensino à distância baseado em tecnologia *multicast*. Não nos trave esta ideia, de fazer algo inovador, aproveitando todos os ensinamentos recolhidos neste capítulo.

Um sistema de ensino à distância deverá ser por definição ubíquo, chegando ao lugar mais remoto, e sempre direccionado para o utilizador final. Como tal, não se enveredará por uma solução apenas praticável a nível laboratorial, onde conceptualmente estaríamos a falar de um sistema de tele-ensino, mas em que os e-alunos se tinham de deslocar a um laboratório com uma rede experimental, onde poderiam assistir às sessões com todos os requintes tecnológicos e abundância de recursos de interligação.

O sistema de ensino à distância deverá ser baseado em interface *web*, recorrendo a *middleware* (módulos em Java e Javascript) sempre que as funcionalidades requeridas o exijam. A Internet é uma das plataformas de acesso mais democrático, a comunidade de e-alunos já lá está em potência, e já experimentou eventualmente sistemas similares. A infoperícia não deverá ser factor de exclusão, pois os objectivos passam também por tornar trivial o acesso aos conteúdos, e realizar de forma transparente a adaptação e execução das aplicações que lidam com os conteúdos media, mesmo os de natureza síncrona, mormente as conferências multimédia.

Como trabalho académico da área da engenharia informática, a ênfase será colocada, não no conceito de "curso" e tudo o que isso envolve mas sim, no conceito de "serviço", particularmente serviços de media em tempo real para comunicação em grupo, com melhoramento de eficiência e alguma garantia de QoS.

Como requisitos incontornáveis, pontificarão a usabilidade e a acessibilidade dos conteúdos didáctico-pedagógicos, evitando criar um sistema que mimetize o sistema de ensino convencional, mas antes que o complemento.

A adaptabilidade deverá ser gerida pelo servidor mas, exercida no *site* do cliente, sendo que cada utilizador reúne um conjunto próprio de definições, será no seu sistema que a sua configuração optimizada será computada, baseando tal computação nas directivas do *middleware*, que importa do servidor como cliente *web*.

A solução no tocante às aplicações, deverá ser a da utilização de ferramentas *multicast* de domínio público, gratuitas e já com alguma difusão. Como se sabe, tais aplicações não se adaptam às variáveis condições de transmissão. Como dotá-las de

adaptabilidade? Uma possibilidade seria reescrever o seu código incorporando tais melhoramentos, mas essa solução não reúne grandes apoios, se pensarmos que a comunidade de e-alunos estará sempre dependente da aplicação proprietária e não poderá actualizar-se de forma independente com as ferramentas *multicast* que bem entender. Assim será preferível exercer o controlo de QoS independentemente da aplicação, incorporando algum *middleware* a montante.

Precedendo a sessão interactiva e por forma a adaptar a parametrização, prévia ao lançamento (automático e transparente) das aplicações *multicast* de media em tempo real, computar-se-ão as condições de transmissão e recursos de processamento para cada novo membro interventivo.

Acrescente-se que a comunidade de e-alunos, tão heterogénea como a própria Internet, está impossibilitada de se sincronizar em termos de tecnologia, infoperícia, tipos de conexões, etc., pelo que o sistema de ensino terá de adaptar-se a este cenário real e incontornável.

No próximo capítulo abordar-se-á a temática do *multicast* IP, fornecendo os aspectos teóricos fundamentais da metodologia comunicacional que o sistema de EAD implementará.

Capítulo 3

Multicast IP

Neste capítulo apresentam-se os fundamentos técnicos do *multicast IP*. A totalidade das aplicações usadas neste trabalho comunicam nesta modalidade, emitindo/recebendo os seus fluxos para/de entidades abstractas, os grupos *multicast*. Sendo este trabalho orientado também para a gestão de recursos associados ao processo de EAD, justifica-se a escolha da tecnologia *multicast IP* pela poupança de recursos associada.

Usando a tecnologia de comunicação de difusão selectiva, ou *multicast IP*, as aplicações podem esperar da rede, uma entrega eficiente da informação que transaccionam. A eficiência advém da utilização de um árvore de encaminhamento que evita replicações desnecessárias dos pacotes *multicast*, i.e., por cada ramo apenas passa uma cópia da informação a transmitir. As replicações acontecerão, eventualmente em última instância, no *router* que serve directamente vários membros de um grupo

multicast. Por este motivo, a comunicação *multicast* adequa-se totalmente às aplicações usadas neste trabalho, com relevância para a transmissão de áudio e vídeo em tempo real pois, diminui a carga da rede (usa menos largura de banda) e alivia o esforço de processamento dos *routers* intermédios.

Em [Almeroth99] pode encontrar-se uma resenha do desenvolvimento do *multicast* IP, inclusivamente as perspectivas de desenvolvimento para a Internet2.

Os protocolos *multicast*, relativos ao controlo, encaminhamento e transporte foram incluídos como anexo (ver Anexo I), fornecendo assim todo o suporte teórico para a completa compreensão dos mecanismos que regulam as comunicações *multicast*.

Como acontece em quase todos os quadrantes, existe a oposição ao *multicast* IP, alguns investigadores opinam que, a plataforma *multicast* IP apresenta problemas de escalabilidade e implantação e como tal propõem arquitecturas alternativas, um exemplo é a *End System Multicast* (ESM) descrita em [Chu01].

3.1. Estabelecimento do *multicast* IP

Comunicações e aplicações *multicast* em ambiente TCP/IP não são novidade. Em 1985, Steve Deering (o pai do *multicast*) e Dave Cheriton, escreveram o primeiro RFC (966) da IETF sobre *multicast* IP, eram então estudantes da Stanford University. Preconizavam que a classe D do protocolo IP deveria ser reservada para endereçamento *multicast*. Seguiu-se o RFC 988, que descrevia o mecanismo para determinar, por parte dos *routers*, os membros de um grupo. O desenvolvimento foi continuado pelo RFC 1054, que apresenta o funcionamento do protocolo IGMPv0 (*Internet Group Management Protocol*), o qual é usado pelas estações para notificar o *router* mais próximo, que se juntaram ou abandonaram um determinado grupo. O RFC 1112, de 1989, revê o anterior e anuncia a nova versão do IGMPv1, que se mantém como a mais comum especificação desse protocolo. Em 1988 fora apresentado o RFC 1075, que descrevia o primeiro protocolo de *routing multicast*, o DVMRP, *Distance Vector Multicast Routing Protocol*.

Sendo que a descrição e especificação do *multicast* IP surgiu nos anos 80, entenda-se para a comunidade científica da Internet, a expansão significativa do *multicast* deu-se após o aparecimento, em 1992, do Mbone (*Multicast Backbone*). O Mbone fornece a plataforma experimental, de rede global, para esses propósitos e especificações, bem

como o veículo para ensaiar novas aplicações *multicast* [Miller98]. Continua em franco crescimento, mormente utilizado como ferramenta de pesquisa e investigação por parte de académicos, ou por técnicos de organizações de índole comercial.

Multicast IP define-se como sendo a transmissão de um datagrama IP para um "host group" [Deering89], um conjunto de zero ou mais estações identificadas por um único endereço IP de classe D. Um datagrama *multicast* é entregue a todos os membros desse destino com a mesma fiabilidade do tipo "best-effort" como acontece com os datagramas *unicast* IP convencionais, i.e., não é garantido que o datagrama seja entregue a todos os membros pertencentes a esse grupo destino ou na mesma ordem relativamente aos outros datagramas.

O encaminhamento dos datagramas *multicast* IP é assegurado por "multicast routers" ou simplesmente, *m*routers. Uma estação transmite um datagrama *multicast* IP como *multicast* de rede local que atinge os membros vizinhos imediatos da sub-rede pertencentes ao grupo de destino. Se o datagrama possuir um TTL (*time-to-live*) maior que 1, os *routers multicast* acoplados à rede local tomam a responsabilidade de encaminhar esse tráfego a todas as outras redes que possuam membros do grupo destino. Nessas redes atingidas, portanto dentro do limite de TTL do IP, o *m*router local completa a entrega dos datagramas transmitindo-os aos membros como *multicast* local.

3.1.1. O modelo "host group"

O conceito fulcral em *multicast* IP tem a ver com o modelo "host group" [Maufer97]. Um "host group" é representado por um único endereço de grupo. Os pacotes destinados ao grupo são enviados para um endereço de grupo, então, caberá aos *routers* determinar como alcançar os membros do grupo.

Os "host groups" poderão ter qualquer número de membros dispostos de forma arbitrária ao longo da topologia da rede. Os grupos podem teoricamente apresentar qualquer forma, tamanho ou extensão geográfica, e outras estações poderão juntar-se ou abandonar o grupo sem restrição, sendo que ser membro de um grupo é uma decisão local. Trata-se como tal de grupos dinâmicos. Uma estação pode pertencer a mais de um grupo ao mesmo tempo.

Uma faceta importante dos "host groups" reside no facto de que os emissores não necessitam de pertencer, como membros, aos grupos aos quais enviam tráfego.

Provavelmente, o modelo "*host group*" caracteriza-se, com capital importância, pela inclusão num grupo estar dependente da iniciação como receptor. Quando o emissor envia tráfego, é da responsabilidade da rede entregá-lo a qualquer receptor conhecido. No caso de não existirem receptores, os *routers* afectos à rede do emissor não aceitam o tráfego para subsequente processamento e o tráfego não chega sequer a sair da sub-rede do emissor.

Os iniciados como receptores juntam-se não descuidando ou sacrificando a escalabilidade: à medida que o grupo cresce, crescem também os potenciais emissores que a qualquer momento podem participar activamente, fazendo com que a(s) árvore(s) de distribuição fique(m) mais participada(s).

Um "*host group*" pode ser permanente ou transiente [Deering89]. Um grupo permanente tem um endereço IP administrativamente atribuído, apenas o endereço é permanente, a quantidade de membros pode variar ou até ser nula. Os restantes endereços *multicast* IP que não estão reservados para grupos permanentes, estão disponíveis para atribuição dinâmica aos grupos transientes, cuja existência está dependente da existência de membros nesses grupos.

3.1.2. Endereçamento *multicast* IP

Cada endereço *multicast* representa um grupo de dimensão arbitrária, e os membros de grupos *multicast* podem encontrar-se algures no espaço de endereçamento *unicast*, compreendido na classe A, B ou C. Uma classe de endereçamento separada, foi destinada para identificar os grupos *multicast*, a saber, classe D.

A: 0aaa aaaa
B: 10bb bbbb
C: 110c cccc
D:1110 dddd

Ilustração 1 - Padrões dos bits de ordem mais elevada no primeiro octeto [Maufer97].

Os restantes dddd bits no byte mais significativo do endereço IP de classe D podem tomar qualquer valor, então a gama completa do espaço de endereçamento de classe D delimita-se entre 224.0.0.0 e 239.255.255.255 (em binário, de 1110 000 até 1110 1111).

Cada um destes endereços representa uma unidade lógica singular : um grupo de estações IP finais, também conhecido como "*host group*". Não existe qualquer estrutura dentro deste espaço de endereçamento similar à constituição de sub-redes no espaço de endereçamento *unicast*, embora o conceito de agregar conjuntos de endereços de grupos já tenha sido discutido. Como nenhuma estrutura foi imposta na classe D, ela é fundamentalmente plana. O "*host group*" denotado pelo endereço 239.4.17.32 não está necessariamente relacionado a qualquer grupo usando os endereços 239.4.17.33, ou 239.4.17.31, ou 225.1.42.8 por este motivo.

3.1.2.1 Endereços *multicast* reservados

O RFC-1700 estabelece: A gama de endereços entre 224.0.0.0 e 224.0.0.255, inclusive, é reservada para o uso de protocolos de *routing* e outra descoberta de topologia de baixo nível ou protocolos de manutenção, tais como descoberta de *gateway* e relato da constituição de grupos. Os *routers multicast* não devem propagar qualquer datagrama *multicast* com endereço de destino situado nesta gama, independentemente do seu TTL.

Nesta gama reservada, que pode também ser escrita como 224.0.0.0/24, várias atribuições já foram feitas, consulte-se o **Quadro 1**.

Endereço	Utilização
224.0.0.0	Endereço Base (Reservado)
224.0.0.1	Todos os sistemas numa sub-rede
224.0.0.2	Todos os <i>routers</i> numa sub-rede
224.0.0.3	Não atribuído
224.0.0.4	<i>Routers</i> DVMRP
224.0.0.5	Todos os <i>routers</i> OSPF
224.0.0.6	<i>Routers</i> OSPF designados
224.0.0.7	<i>Routers</i> ST
224.0.0.8	Estações ST
224.0.0.9	<i>Routers</i> RIP2
224.0.0.11	Agentes Móveis (<i>Mobile Agents</i>)
224.0.0.12	Agente DHCP <i>Server/Relay</i>
224.0.0.13	Todos os <i>routers</i> PIM
224.0.0.14	RSVP, ENCAPSULATION
224.0.0.15	Todos os <i>routers</i> CBT
224.0.0.16- 224.0.0.255	Não atribuídos

Quadro 1 - Alguns endereços *multicast* reservados permanentes [Maufer97].

Existem outros endereços *multicast* permanentemente atribuídos para aplicações não enquadradas na categoria das anteriormente focadas. É útil dispor de um endereço *multicast* IP bem determinado para certas aplicações. Segue-se uma pequena selecção desses endereços, presente no **Quadro 2**.

Endereço	Utilização
224.0.1.1	NTP (<i>Network Time Protocol</i>)
224.0.1.2	SIG - Dogfight
224.0.1.3	rwhod
224.0.1.5	Horizontes Artificiais - Aviator
224.0.1.20	Qualquer experimento privado
224.0.1.21	DVMRP e MOSPF
224.0.1.22	SVRLOC
224.0.1.23	XINGTV
224.0.1.32	mtrace

Quadro 2 - Alguns endereços *multicast* reservados permanentes para outras aplicações [Maufer97].

Para uma completa referência destes endereços veja-se o RFC-3171 ou sucessor.

A gestão deste espaço de endereçamento é mantida pela *Internet Assigned Numbers Authority (IANA)*, acessível pelo URL: <http://www.iana.org/>.

3.1.3. Controlar o âmbito do encaminhamento *multicast*

Existem vários mecanismos disponíveis para controlar o alcance, em termos de saltos de *routers*, da propagação de uma transmissão *multicast*. As duas principais formas de exercer esse controlo são os mecanismos de âmbito administrativamente controlado e a especificação de um valor de TTL (*time-to-live*) que diminuirá a cada salto de *router*. Acrescente-se que ambos métodos podem ser usados em conjunto, e que certos endereços de grupo incluem implicitamente uma abrangência limitada.

3.1.3.1 Abrangência controlada administrativamente

Os endereços *multicast* não-reservados são destinados para o uso interno de redes privadas, tais como "intranets" de empresas ou universidades. "Host groups" na Internet pública não devem usar estes endereços, e se redes privadas empregam tais endereços,

os pacotes destinados a esses grupos não deverão ser encaminhados para além das fronteiras da rede da organização.

Este tipo de abrangência providencia uma semântica mais clara e simplificada de controlar o âmbito das comunicações *multicast*. Basicamente os pacotes *multicast* controlados desta forma não atravessam as fronteiras da sub-rede, configuradas administrativamente. Adicionalmente, os endereços *multicast*, da gama dos não-reservados, são atribuídos localmente e não conflituam com outros grupos que podem usar o mesmo endereço noutra região *multicast*.

A gama de endereços *multicast* para utilizar em difusão selectiva privada vai de 239.0.0.0 até 239.255.255.255.

3.1.3.2 Abrangência baseada em TTL

O campo TTL no cabeçalho IP é um mecanismo de salvaguarda, inicialmente desenhado para descartar datagramas IP que se viam envolvidos em ciclos intermináveis de *routers*. Acresce-lhe ainda a funcionalidade de poder limitar a abrangência ou contingência dos datagramas, limitando o número de *routers* IP que pode atravessar. Cada *router* decreta de uma unidade o campo TTL, e não transmite o datagrama quando o TTL atinge zero.

Tipicamente, quando um datagrama expira (TTL=0), um *router* irá enviar uma mensagem de erro ao emissor do tipo *Internet Control Message Protocol* (ICMP) com o conteúdo "*TTL Exceed*". A aplicação que reporta o percurso de um datagrama *multicast-trace*-, ordinariamente utilizada em sistemas que utilizam TCP/IP, utiliza esta mensagem para localizar os *routers*. Contudo o RFC 1112 estipula que quando o TTL é excedido num datagrama enviado para uma endereço *multicast*, o *router* não deve enviar qualquer mensagem de erro ICMP. Os datagramas *multicast* expiram silenciosamente.

3.2. Benefícios do *multicast*

O IPv4 apresenta três tipos de endereços: *unicast*, *broadcast* e *multicast*. Um endereço *unicast* é usado para enviar um pacote para um único destino. Um endereço *broadcast* usa-se para enviar um datagrama para um sub-rede inteira. O endereço

multicast é utilizado para enviar um datagrama para várias estações que podem pertencer a diferentes sub-redes mas que estão filiadas num mesmo grupo *multicast*.

Os tipos de comunicação são igualmente três: ponto-a-ponto, ponto-multiponto e multiponto-multiponto (ver **Figura 6**). Por forma a otimizar as comunicações, cada aplicação, que requer um tipo diferente de comunicação, deveria usar um tipo de endereço adequado, mas tal nem sempre acontece. Nas distribuições de vídeo, por exemplo, é usada a comunicação ponto-a-multiponto. Nas vídeo-conferências, por exemplo, é usada a comunicação multiponto-a-multiponto. Para ambas seria adequado usar *multicast*.

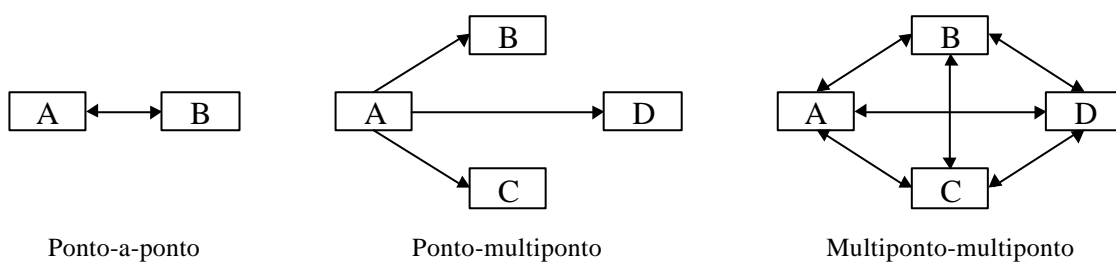


Figura 6 - Tipos de comunicação.

Tomamos como exemplo a **Figura 7**, para transmissão de vídeo do *host* H1 para os *hosts* H2, H3, H4 e H5, distribuídos em três sub-redes e interligados pelos *routers* R1, R2 e R3. Para efectuar esta transmissão, usando *routing unicast*, seria necessário abrir e manter pelo menos quatro sessões ponto-a-ponto (N-1). Se os demais *hosts* do grupo também desejarem transmitir dados, o número de sessões passa a ser $N*(N-1)$.

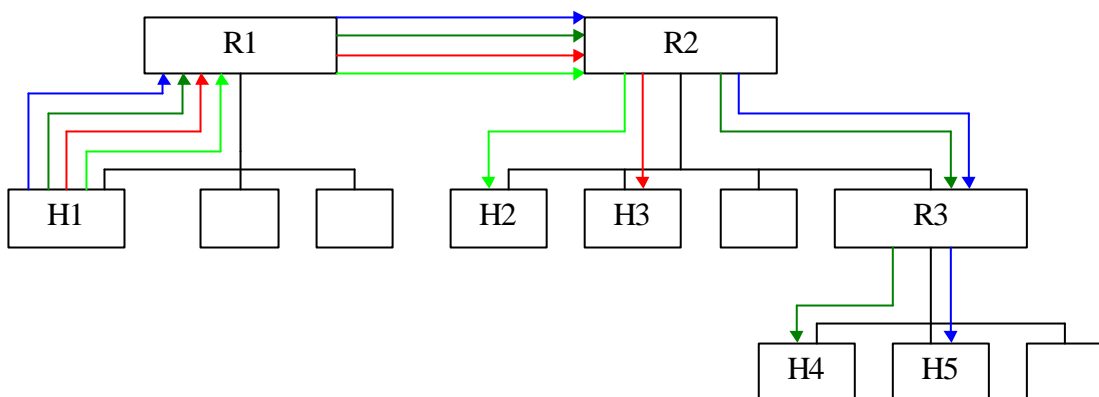


Figura 7 - Transmissão ponto-multiponto com encaminhamento *unicast*.

Para obviar o mesmo problema anteriormente exposto, desta feita usando a modalidade de comunicação *multicast*, os resultados serão bem mais interessantes, veja-se a **Figura 8**. Para esta transmissão seria necessário abrir e manter apenas uma sessão *multicast*. No cenário multiponto-multiponto, onde todos comunicam com todos, interactividade portanto, o encargo para a rede continuaria a ser mínimo.

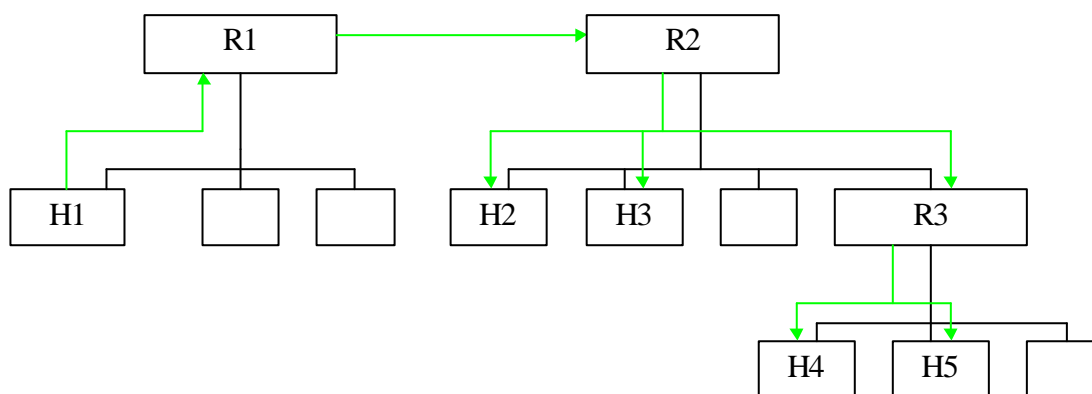


Figura 8 - Transmissão ponto-multiponto com encaminhamento *multicast*.

Pode ver-se o *multicasting* como uma útil abstracção de programação, mas é detentor de uma vantagem adicional, a de que pode ser suportado ao nível do hardware em sistemas físicos de comunicações baseado em barramento numa maneira inerentemente paralela [Greenhalgh96]. Numa rede onde os *routers* suportam *multicast*, o custo para o emissor de *multicasting* pode ser virtualmente o mesmo de um único envio *unicast* : a responsabilidade de copiar e multiplamente encaminhar a informação é assumida pela tecnologia de rede em uso.

Para implementar *multicast* utilizando apenas envios *unicast*, requer que a mesma informação seja transmitida na modalidade *unicast* para cada destino individualmente. Este processo sofre de duas desvantagens comparado com o genuíno *multicast* :

- o esforço do emissor tem uma componente $O(N_{\text{destinos}})$ relativamente aos envios repetidos; e
- a carga da rede (em segmentos partilhados e *routers*) é aumentada, especialmente perto do emissor, uma vez que a mesma informação deve ser manipulada tantas vezes quantos N_{destinos} no mesmo local. Com *multicast*, a informação deve apenas estar presente uma vez em cada segmento da rede, bem como chegar um único exemplar a cada *router*.

Apesar de não resolver todos os problemas de escalabilidade, o *multicasting* possibilita elevar o número de destinatários da informação sem alterar a tecnologia da infra-estrutura de rede.

3.3. Custos do *multicast*

Como usualmente acontece existe o reverso da medalha. Os custos do *multicasting* são mais subtis que os benefícios. Destacam-se as mensagens de controlo para assegurar o encaminhamento *multicast* e constituição de grupos, e o estado relativo dos grupos que deve ser mantido pelos nós de *routing multicast*.

Para *multicast* fiável também acresce alguma sobrecarga para assegurar fiabilidade na forma de confirmações (ACKs) e pedidos de reenvio. Para um grupo *multicast* esta sobrecarga de suporte à fiabilidade será pelo menos $O(N_{\text{destinos}})$.

A difusão selectiva é peculiar no tocante aos requisitos de segurança. Em geral, o desenho e concepção de um dado protocolo de segurança *multicast* é um empreendimento complexo que deve tomar em consideração um alargado número de factores. Todavia, a gestão de uma associação de segurança dinâmica e do seu material de codificação constitui a diferença básica entre protocolos de segurança *unicast* e *multicast*.

A segurança em *multicast* existe apenas em laboratórios de pesquisa e investigação desta área. Uma implantação mais alargada da tecnologia *multicast* depende, em grande medida, das garantias de segurança. Eis alguns dos trabalhos mais referenciados na área do estudo e implementação de *frameworks* para *multicast* seguro: [Mittra97, Canetti98, Kruus98, Finlayson99, Canetti99, Banerjee01].

3.4. Mbone - *Multicast Backbone*

As comunicações por difusão selectiva são executadas principalmente sobre a Internet, numa rede virtual denominada Mbone.

A razão pela qual se define como sendo uma rede virtual explica-se porque partilha a mesma infra-estrutura física da Internet, embora use um sistema de *routers* que podem suportar genuinamente tráfego *multicast*, os denominados *mrouters* que surgem nalguns pontos da topologia da Internet. Às regiões que estão plenamente habilitadas para

suportar tráfego *multicast*, dá-se vulgarmente o nome de ilhas *multicast*. Estas regiões estão interligadas por túneis de comunicação onde o tráfego é encapsulado em *unicast* IP e aquando da entrega é desencapsulado pelo *mrouter* sem perder as suas características originais.

Nestes sistemas baseados em *multicast* IP, uma estação transmite os dados e um grande grupo de pessoas em diferentes localizações pode receber a transmissão em "tempo real" simultaneamente. É uma tecnologia utilizada para transmissões de seminários, apresentações comerciais, aulas de ensino a distância, publicações e discussões de pesquisas e experimentos, etc.

3.4.1. História do Mbone

Como referido anteriormente, o conceito de endereçamento com difusão selectiva para comunicações de grupo foi introduzido por Steve Deering na sua tese de doutoramento [Deering91] na *Stanford University*, USA, e depois desenvolvido na Xerox PARC. O projecto e desenvolvimento do Mbone foi ajudado por Van Jacobson e o seu grupo no Lawrence Berkeley Labs (LBL) e Steve Casner no *Information Science Institute* (ISI), além de muitos outros engenheiros [Kumar95].

O Mbone foi criado como uma rede experimental de difusão selectiva para o protocolo IP, desenvolvido e testado na rede de pesquisa DARTnet. Na rede DARTnet eram transmitidas semanalmente conferências para os seus investigadores, porém, a primeira grande transmissão Mbone ocorreu somente em Março de 1992, durante o encontro do IETF em San Diego, USA, quando muitas sessões do encontro foram transmitidas em áudio utilizando transmissões *multicast* do *site* IETF para participantes de 20 *sites* em três continentes, abrangendo 16 zonas horárias [Casner92]. O evento foi uma demonstração da tecnologia desenvolvida na rede DARTnet, e teve grande significado pela dimensão da topologia da rede de difusão selectiva IP envolvida.

Foram transmitidas todas as sessões gerais do encontro e também algumas sessões de grupos de trabalho. Os participantes remotos tinham a possibilidade de falar, participando nas discussões e fazendo perguntas, e embora a transmissão não fosse perfeita, foi razoável em ambas as direcções. A transmissão do evento foi possível graças à utilização de difusão selectiva IP. Se fosse necessário enviar uma cópia de cada

pacote para cada destino, a largura de banda e processamento requerido seriam proibitivos.

No evento foram utilizados equipamentos e *software* testados na DARTnet. No *site* da IETF e na maioria dos *sites* remotos foi utilizada a ferramenta VAT, *Visual Audio Tool*, desenvolvida por Van Jacobson e Steve McCanne, do LBL.

3.4.2. O Mbone na actualidade

Na impossibilidade de esperar pela ubiquidade dos *routers multicast* na Internet para desenvolver e testar aplicações *multicast*, nasce a rede semi-permanente IP.

O Mbone cresceu de 40 sub-redes em 4 países em 1992, a várias centenas de sub-redes em 1999. Surgiram novas aplicações de difusão selectiva acompanhadas de novos serviços baseados em *multicast*, sendo previsível que a tecnologia *multicast* na Internet seguirá crescendo. A **Figura 9** mostra algumas ligações internacionais do Mbone.

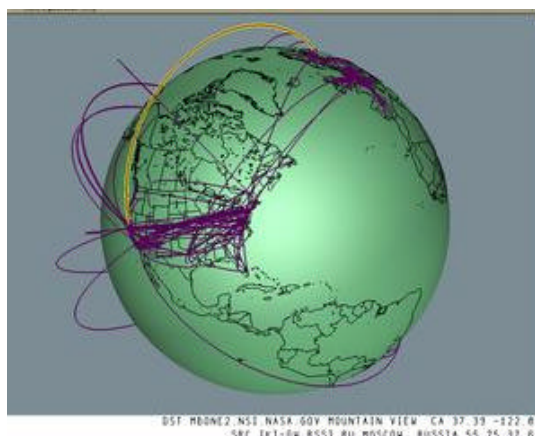


Figura 9 - Aspecto global da rede Mbone.

A maioria das regiões Mbone estão correntemente interligadas pelo DVMRP - *Distance Vector Multicast Routing Protocol*. Internamente, as regiões podem executar qualquer protocolo de *routing* que pretenderem: MOSPF - *Multicast extensions to Open Shortest Path First*, protocolos de *routing* PIM - *Protocol Independent Multicast*, ou o referido DVMRP.

À medida que as capacidades do software de *routing multicast* se tornam disponíveis de forma alargada nos *routers* da Internet, os fornecedores de serviços IP poderão gradualmente decidir usar *multicast* "nativo" como alternativa ao uso de vários

túneis (ver **Figura 10**). Diz-se que existe *multicast* "nativo" quando uma região de *routers* (ou um conjunto de regiões) operam sem túneis [Maufer97]. Todas as sub-redes estão conectadas, pelo menos, por um *router* que tem capacidade de encaminhar pacotes *multicast*. Nestas condições, os túneis são dispensáveis: os pacotes *multicast* apenas fluem para onde são necessários, baseado na localização relativa da(s) fonte(s) e receptor(es).

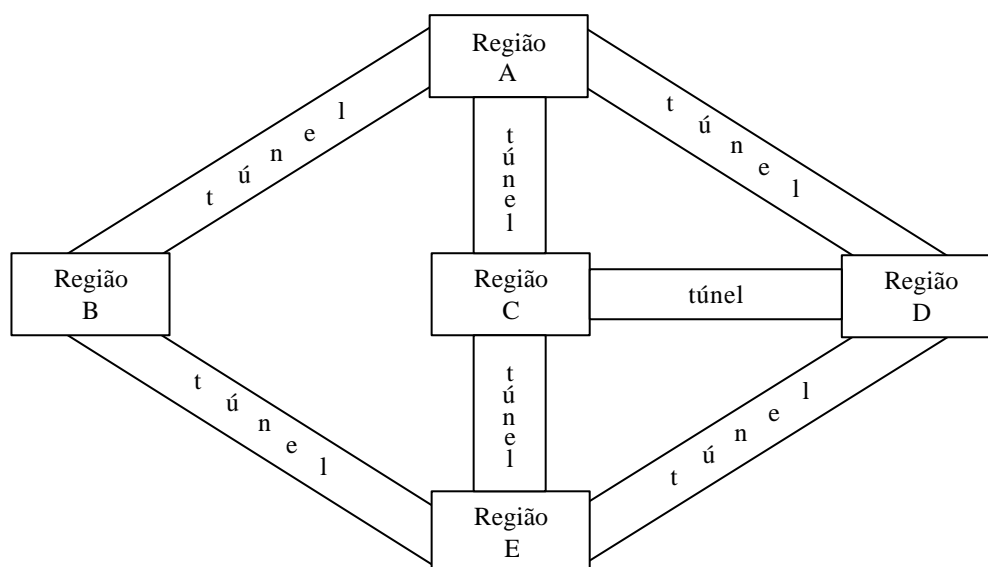


Figura 10 -Estrutura actual do Mbone [Maufer97].

O mercado de *routers multicast* é dominado pela *Cisco*, embora outros fabricantes como a *3Com*, *Bay Networks* e *Alantec/Fore* estejam a colocar *routers* no Mbone, agregando novos protocolos de *routing*. Com isto o Mbone não é de todo homogéneo, o protocolo PIM é provavelmente o protocolo *multicast* que ganha mais popularidade para interligação de domínios à medida o DVMRP perde terreno. Todos estes protocolos de encaminhamento *multicast* estão tratados com maior detalhe no Anexo I.

3.5. Multicast em IPv6

A motivação capital para o desenvolvimento do IPv6 [Thomas96] foi o alarme acerca do esgotamento do espaço de endereçamento do IPv4, devido ao crescimento exponencial da Internet. O IPv6 apresenta 128 bits de espaço de endereçamento, recorde-se que a versão actual, o IPv4 apresenta 32 bits.

Os endereços *multicast* em IPv6 são identificados pelos primeiros 8 bits do endereço serem todos iguais a 1, como se mostra na **Figura 11**.

8 bits	4 bits	4 bits	112 bits
11111111	flgs	scop	identificação do grupo <i>multicast</i>

Figura 11 - Organização de um endereço *multicast* IPv6 [Miller98].

Os dois campos seguintes, "flgs" e "scop", indicam o tipo de endereço *multicast* que está a ser descrito. Para utilizações *multicast*, os três bits de ordem mais elevada de "flgs" são colocados a zero. O bit de ordem mais baixa, se colocado a um, indica que o endereço *multicast* é transiente.

O campo "scop" composto por 4 bits é usado para limitar o âmbito do grupo *multicast* - que se pode entender como o controlo de âmbito efectuado pelo IPv4 para limitar o número de *routers* atravessados por um datagrama. Os valores admissíveis para "scop" são os seguintes:

0	Reservado
1	Âmbito de nodo local
2	Âmbito de ligação local
3	(Não atribuído)
4	(Não atribuído)
5	Âmbito de <i>site</i> local
6	(Não atribuído)
7	(Não atribuído)
8	Âmbito de organização local
9	(Não atribuído)
A	(Não atribuído)
B	(Não atribuído)
C	(Não atribuído)
D	(Não atribuído)
E	Âmbito Global
F	Reservado

Quadro 3 - Valores de controlo de âmbito em IPv6 [Miller98].

Esta configuração deixa livres 112 bits para identificar o endereço de grupo, que constituem o campo de endereçamento *multicast* do IPv6.

3.6. Implantação da tecnologia *multicast* IP em Portugal

Em Portugal a FCCN⁷ é a entidade responsável pela organização e gestão das iniciativas e comunicações sobre o Mbone. A infra-estrutura de rede é composta por quinze POPs (Pontos de Presença) geograficamente distribuídos pelo país. Estes POPs comunicam entre si, com a comunidade académica e de investigação nacional e a Internet em geral, utilizando a RCTS - Rede Ciência, Tecnologia e Sociedade.

As instituições portuguesas que aderiram à rede Mbone são académicas, Universidades e Politécnicos, onde se podem contar a Universidade do Minho e Instituto Politécnico de Bragança, que através deste trabalho encontraram uma forma de cooperação.

As instituições aderentes devem sujeitar-se às regras impostas pela FCCN e registar-se como participantes. Para enviar/receber tráfego *multicast*, integrando o Mbone, devem configurar o seu equipamento interno de comutação. Mais informação relativa ao processo encontra-se em: <http://www.fccn.pt/projectos/multicast/index.html>.

3.6.1. O Mbone visto da rede do IPB (Instituto Politécnico de Bragança)

No **Quadro 4** mostra-se o *output* do comando *mrinfo* (informação da configuração do *mrouter*) gerado pelo *mrouter* que interliga o IPB ao Mbone.

Pode apreciar-se que o protocolo de encaminhamento *multicast* usado é o PIM (*Protocol Independent Multicast*) e as respostas dos *routers multicast* vizinhos. Neste caso são os *routers* dispersos pelas três escolas do IPB.

O *mrinfo* demonstra que o *mrouter* do IPB se encontra ligado ao *mrouter* da da RCCN (Rede para a Computação Científica Nacional) localizado em Bragança, que é um dos POPs da RCTS e através dessa WAN ao Mbone.

Espera-se que os operadores privados permitam aos seus subscritores, a breve termo, o acesso à rede Mbone e conseqüentemente ao tráfego *multicast*. Tal feito alargaria sobremaneira o espectro de população com capacidade para aceder aos serviços que este trabalho se propõe desenvolver, sempre e quando, obviamente a largura de banda o permita, já que no mínimo seria exigível uma ligação RDIS, ou idealmente superior.

⁷ Fundação para a Computação Científica Nacional

```

193.136.195.254 (router3.eth0.ccom.ipb.pt) [version 12.0] [flags: PMSA]:
193.136.195.254 -> 193.136.195.158 (router3.eth5.ccom.ipb.pt) [1/0/pim]
193.136.195.254 -> 193.136.231.126 (router3.eth2.ccom.ipb.pt) [1/0/pim]
193.136.195.254 -> 193.136.195.126 (router3.eth1.ccom.ipb.pt) [1/0/pim]
193.136.195.254 -> 193.136.231.158 (router3.eth4.ccom.ipb.pt) [1/0/pim]
193.136.195.254 -> 193.136.194.254 (router3.eth3.ccom.ipb.pt) [1/0/pim]
193.136.195.126 -> 193.136.195.158 (router3.eth5.ccom.ipb.pt) [1/0/pim]
193.136.195.126 -> 193.136.231.126 (router3.eth2.ccom.ipb.pt) [1/0/pim]
193.136.195.126 -> 193.136.231.158 (router3.eth4.ccom.ipb.pt) [1/0/pim]
193.136.195.126 -> 193.136.195.254 (router3.eth0.ccom.ipb.pt) [1/0/pim]
193.136.195.126 -> 193.136.194.254 (router3.eth3.ccom.ipb.pt) [1/0/pim]
193.136.231.126 -> 193.136.231.158 (router3.eth4.ccom.ipb.pt) [1/0/pim]
193.136.231.126 -> 193.136.194.254 (router3.eth3.ccom.ipb.pt) [1/0/pim]
193.136.231.126 -> 193.136.195.126 (router3.eth1.ccom.ipb.pt) [1/0/pim]
193.136.231.126 -> 193.136.195.254 (router3.eth0.ccom.ipb.pt) [1/0/pim]
193.136.231.126 -> 193.136.195.158 (router3.eth5.ccom.ipb.pt) [1/0/pim]
193.136.194.254 -> 193.136.195.158 (router3.eth5.ccom.ipb.pt) [1/0/pim]
193.136.194.254 -> 193.136.231.126 (router3.eth2.ccom.ipb.pt) [1/0/pim]
193.136.194.254 -> 193.136.195.126 (router3.eth1.ccom.ipb.pt) [1/0/pim]
193.136.194.254 -> 193.136.231.158 (router3.eth4.ccom.ipb.pt) [1/0/pim]
193.136.194.254 -> 193.136.195.254 (router3.eth0.ccom.ipb.pt) [1/0/pim]
193.136.231.158 -> 193.136.195.158 (router3.eth5.ccom.ipb.pt) [1/0/pim/querier]
193.136.231.126 -> 193.136.195.158 (router3.eth5.ccom.ipb.pt) [1/0/pim]
193.136.194.254 -> 193.136.195.158 (router3.eth5.ccom.ipb.pt) [1/0/pim]
193.136.194.254 -> 193.136.231.126 (router3.eth2.ccom.ipb.pt) [1/0/pim]
193.136.194.254 -> 193.136.195.126 (router3.eth1.ccom.ipb.pt) [1/0/pim]
193.136.194.254 -> 193.136.231.158 (router3.eth4.ccom.ipb.pt) [1/0/pim]
193.136.194.254 -> 193.136.195.254 (router3.eth0.ccom.ipb.pt) [1/0/pim]
193.136.231.158 -> 193.136.195.158 (router3.eth5.ccom.ipb.pt) [1/0/pim/querier]
193.136.231.158 -> 193.136.231.126 (router3.eth2.ccom.ipb.pt) [1/0/pim/querier]
193.136.231.158 -> 193.136.195.126 (router3.eth1.ccom.ipb.pt) [1/0/pim/querier]
193.136.231.158 -> 193.136.194.254 (router3.eth3.ccom.ipb.pt) [1/0/pim/querier]
193.136.231.158 -> 193.136.195.254 (router3.eth0.ccom.ipb.pt) [1/0/pim/querier]
193.136.195.158 -> 193.136.231.158 (router3.eth4.ccom.ipb.pt) [1/0/pim]
193.136.195.158 -> 193.136.194.254 (router3.eth3.ccom.ipb.pt) [1/0/pim]
193.136.195.158 -> 193.136.195.126 (router3.eth1.ccom.ipb.pt) [1/0/pim]
193.136.195.158 -> 193.136.195.254 (router3.eth0.ccom.ipb.pt) [1/0/pim]
193.136.195.158 -> 193.136.231.126 (router3.eth2.ccom.ipb.pt) [1/0/pim]
193.137.58.2 -> 193.137.58.1 (ROUTER21.Serial1.Braganca.rccn.net) [1/0/pim]

```

Quadro 4 - Output do comando *mrfinfo* no *router* do IPB.

3.6.2. Os ISPs nacionais

Nenhum dos ISPs nacionais encaminha, actualmente, tráfego *multicast*, no entanto esta situação pode alterar-se num futuro próximo à medida que os clientes, sobretudo as empresas de alguma dimensão comecem a utilizar ferramentas *multicast*.

As empresas aderirão seguramente a esta tecnologia pois, permite-lhes reduzir os custos associados às comunicações, quer seja para aplicações de tele-formação, voz sobre IP, distribuição de software, etc.

3.7. Utilização da tecnologia *multicast* no trabalho desenvolvido

A necessidade de largura de banda e a capacidade de processamento afectam-se mutuamente. Num ambiente de escassez de largura de banda reduz-se ao máximo a informação a transmitir, sendo o processamento muito elevado para computar os algoritmos de compressão e descompressão da informação veiculada, por oposição, na abundância de largura de banda, em caso extremo a informação pode ser transmitida sem qualquer compressão e como tal o processamento é drasticamente reduzido.

Sabemos que fisicamente, ambas as tecnologias, comunicação e processamento têm um limite, mais tarde ou mais cedo será atingido, primeiro num campo depois no outro, e só a optimização do aproveitamento de ambos, um equilíbrio perfeito de ambas funcionalidades permitirá continuar na senda do desenvolvimento e da superação.

A gestão dos recursos pode ser a ideia chave para compreender o Mbone, a razão pela qual a entrega de um pacote *multicast* se pode considerar eficiente, atendendo ao dispêndio de recursos envolvidos, reside no facto de ser transmitido com um encargo mínimo para a rede (em termos de largura de banda e processamento do equipamento de *routing*).

Num sistema de EAD, incluindo aplicações de conferência multimédia interactiva, que se pretende tenha a possibilidade de escalar minimamente (algumas dezenas de e-alunos seria razoável), a gestão dos recursos é fundamental. O *multicast* IP, no tocante ao fornecimento de comunicações parece ser a resposta ideal.

No capítulo seguinte apresentar-se-ão as aplicações *multicast* que se integrarão no sistema de EAD a desenvolver, referindo para as aplicações de áudio e videoconferência a forma como aceitam pré-parametrização da sua dinâmica, possibilitando a adaptação da QoS sem anular a sua utilidade.

Capítulo 4

Aplicações *Multicast*

O *multicast* IP irá desempenhar um papel proeminente na Internet nos anos vindouros [Quinn01]. Trata-se de uma necessidade não de uma opção, se pensarmos em escalar a Internet. O *multicast* permitirá aos que desenvolvem aplicações vocacionadas para a Internet, "adicionar funcionalidade sem um impacto significativo nos recursos da rede" [Bradner97].

As aplicações *multicast* genuínas são aquelas que requerem a presença de uma camada de rede *multicast* possibilitando o seu funcionamento [Maufer97]. Torna-se necessário entregar a mesma informação de uma fonte para vários destinatários, ou de múltiplas origens para múltiplos destinos. Como explicitado no capítulo anterior, o *modus operandi multicast* evita replicações desnecessárias, sendo que cada filiado num grupo *multicast* recebe a informação com o mínimo encargo possível para a rede.

O factor de maior criticidade aquando da determinação da natureza *multicast* de uma aplicação, prende-se com o facto de averiguar se a aplicação desempenha a replicação de pacotes por sua conta própria, ou bem pela máquina emissora, ou ainda pelo despacho da camada de aplicação, ou se, por outro lado, relega para a rede a execução desta tarefa. No caso do IPv4, pode observar-se o endereço de destino dos pacotes da aplicação e verificar se se encontram no intervalo da classe D.

O RFC 3170 descreve claramente qual o estado da arte das aplicações *multicast*, fornecendo soluções e desafios para as ferramentas *multicast* da nova geração.

Neste capítulo, o propósito principal será apresentar e descrever as aplicações *multicast* de domínio público que serão utilizadas no desenvolvimento do sistema de EAD, justificando a sua escolha. Importa igualmente, destacar nessas aplicações, as formas de pré-parametrização admitidas para transmissão dos media e concomitantemente, perceber as diferenças, entre os formatos de codificação de áudio e vídeo, que possam fundamentar as escolhas aquando do exercício da adaptabilidade.

Efectivamente, serão usadas aplicações *multicast*, que preencham cabalmente os requisitos de uma conferência multimédia *multicast* (CMM) com interactividade, desde áudio, vídeo e espaço de trabalho digital (texto+gráficos) partilhado.

Pode dispor-se actualmente de uma pletera de software de criação de conteúdos multimédia, sistemas de suporte a teleconferência e trabalho cooperativo. Contudo, poucos sistemas, vocacionados para a instrução remota ou educação à distância são efectivamente, de reconhecida usabilidade. Outro considerando importante prende-se com o facto de poder garantir escalabilidade, acolhendo com qualidade um número dinâmico, eventualmente crescente, de membros (alunos) interactivos [Maly94].

O advento da tecnologia digital suportando locais virtuais de ensino multimédia interactivo, em *campus*, ou paragens distantes é uma passo importante na solução de problemas de facultação de ensino de elevada qualidade, de forma eficaz e eficiente a estudantes que por razões financeiras, de deficiência física, familiares, vida profissional, etc., não podem assistir ao vivo às sessões pedagógicas.

Não sendo um dos objectivos deste trabalho desenvolver aplicações *multicast*, o sistema de EAD a desenvolver utilizará aplicações *multicast* académicas, de domínio público. Nestas aplicações, mesmo sendo do tipo *open source*, não se enveredará por reescrever parcialmente o seu código, mas antes integrá-las num sistema que lhe maximizará o aproveitamento para a finalidade da educação à distância. O trabalho a desenvolver incidirá também, sobre algum *middleware* necessário, que fará a interface

entre as aplicações *multicast* e o utilizador, no sentido de simplificar a sua utilização e ao mesmo tempo dotá-las de adaptabilidade aos recursos disponíveis, monitorizados em tempo de execução, na rede e no sistema final, veja-se a **Figura 12**. A adaptabilidade só foi considerada para as aplicações *multicast* de áudio e vídeo, as restantes (espaço de trabalho digital partilhado), pelo escasso tráfego que produzem não requerem cuidados especiais de garantia de QoS.

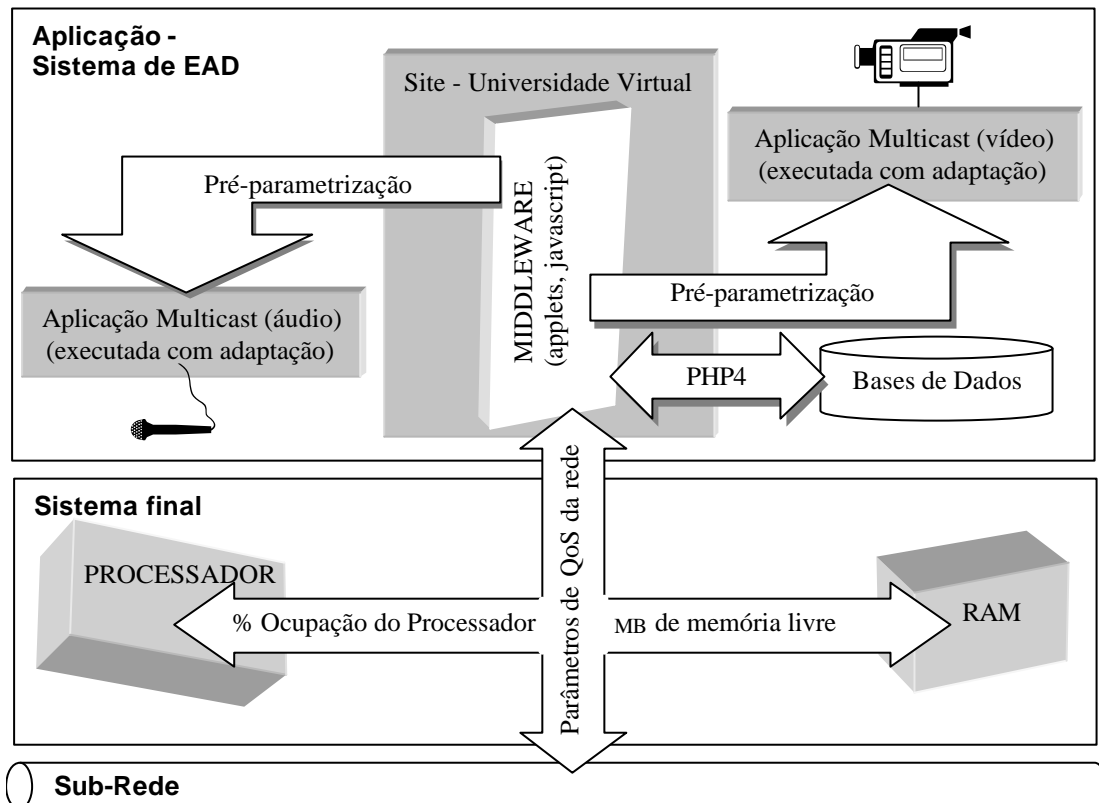


Figura 12 - Integração das aplicações *multicast* no sistema EAD a desenvolver.

Permitir suporte *multicast* em aplicações e protocolos, com a possibilidade de escalar sem problemas num ambiente de rede heterogéneo torna-se num desafio significativo. Especificamente, enviar pacotes de dados a taxa constante, confiabilidade na entrega dos dados, segurança, e gestão das comunicações muitos-para-muitos requer considerações especiais. Algumas soluções encontram-se disponibilizadas, mas muitos desses serviços são ainda áreas activas de pesquisa. Em [Roca00] pode obter-se uma visão geral sobre as diferentes tecnologias *multicast* que a investigação nesta área produziu.

4.1. Categorias de aplicações *multicast*

Dispondo de uma rede IP capacitada para comunicação *multicast*, são possíveis serviços únicos e aplicações poderosas. O *multicast* possibilita a coordenação - está predisposto para sistemas distribuídos fracamente acoplados (de pessoas, servidores, bases de dados, processos, dispositivos...) [Estrin98].

Uma aplicação *multicast* é simplesmente definida como sendo qualquer aplicação que envia a e/ou recebe de um endereço *multicast* IP.

O que diferencia aplicações *multicast* IP de aplicações *unicast* IP, um-para-um, são as outras relações de emissor e receptor que o *multicast* possibilita. Existem três categorias principais de aplicações *multicast* [Quinn01]:

- **Um-para-muitos (1toM)**: Uma simples estação a transmitir para dois ou mais receptores.
- **Muitos-para-muitos (MtoM)**: Qualquer número de estações enviando para o mesmo endereço de grupo *multicast*, bem como recebendo comunicação dele.
- **Muitos-para-um (Mto1)**: Qualquer número de receptores respondendo à fonte (emissor) via *multicast* ou *unicast*.

O ensino à distância poderá ser enquadrado em duas categorias, consoante a modalidade da comunicação a levar a cabo, por exemplo, se for uma palestra pré-gravada e emitida a pedido, sem possibilidade de participar activamente, então será **1toM**. Se a durante uma transmissão, quisermos dar aos instruendos a capacidade de interpelar o tutor, ou ainda se a palestra envolver vários oradores remotos, então a categoria será **MtoM**.

A categoria de aplicações que mais nos interessa neste trabalho é sem dúvida a **MtoM**, pois permite-nos CMMs interactivas.

Os aspectos de coordenação são também importantes. Na moderação da conferência é fundamental determinar claramente quem tem a palavra, ou quem deve ser focado preferencialmente pela transmissão vídeo. As aplicações *multicast* usadas

neste trabalho, funcionam usualmente por auto-gestão do utilizador, não sendo exercida necessariamente moderação rígida.

Os campos de aplicação que poderão beneficiar da tecnologia *multicast* estão, em bom número, ainda por explorar, mas já se destacam alguns:

- distribuição em grande escala de dados, bases de dados, páginas *web*, etc.;
- conferência multimédia e trabalho colaborativo;
- ensino à distância (EAD);
- distribuição de informação financeira;
- monitorização, telemetria;
- registos distribuídos;
- acompanhamento de processos migratórios;
- jogos multi-jogador;
- aplicações com sensores;
- difusão selectiva de software;
- difusão selectiva de televisão.

4.2. Videoconferência

Antes de expor as diferentes ferramentas usadas neste trabalho, importa clarificar o conceito de videoconferência pois ocupa lugar central no trabalho desenvolvido.

Esta modalidade de comunicação responde categoricamente às necessidades do ensino à distância. Como tal recorrer-se-á a aplicações que possibilitem a transmissão RTP, ao vivo ou em diferido, de objectos áudio e vídeo, baseadas em tecnologia *multicast*.

O enorme potencial da videoconferência justifica-se pelo nível de interacção proporcionado pela adição do áudio e do vídeo à comunicação entre instrutores e alunos. O vídeo possibilita ou aperfeiçoa tarefas como demonstrações, práticas que envolvem habilidades, modelação de comportamento e soluções de problemas colaborativos [Yoakam98]. O áudio, por sua vez, está entre os componentes mais críticos de um sistema que se proponha garantir uma comunicação interactiva eficiente. E tratando-se de interactividade, o ditado que reza que uma imagem vale mais que mil

palavras nem sempre é válido, embora ele continue verdadeiro para outras actividades em que, diferentemente da educação, a interactividade não seja um elemento crucial. Por exemplo, entre assistir a uma partida de futebol utilizando uma televisão sem som e um rádio, muitas pessoas certamente fariam a primeira opção. No entanto, numa aula, é muito difícil que um aluno permaneça interessado sem o áudio.

A videoconferência enquadra-se na categoria das tele-apresentações, assim entende-se por tele-apresentação, uma apresentação na qual o apresentador e/ou a audiência, não estão física e temporalmente co-localizados mas estão tele-presentes, distribuídos em diferentes locais e/ou participando repartidas vezes [GemmeII97]. Acredita-se que as tele-apresentações já começaram a revolucionar a educação, conferências, formação profissional, etc., pela redução de custos associados e disponibilizando o material pedagógico, ou outro, a um público imensamente mais vasto [GemmeIII97]. As tele-apresentações que nos interessam neste trabalho são as de tipo *multicast*, consistindo em áudio, vídeo e apresentações gráficas. Estas últimas podem incluir texto, gráficos, imagens, animações e efeitos especiais.

As aplicações de conferência assumem hoje um evidente destaque dentro das ferramentas de trabalho em grupo. Um serviço de conferência eficiente é fundamental para facilitar o desenvolvimento e controlo deste tipo de aplicações. Uma particularidade nuclear das aplicações de conferência é a sua capacidade de comunicação multiponto entre vários utilizadores em ambientes heterogéneos.

Os avanços técnicos nos sistemas de comunicação e a organização das instituições e empresas modernas (formação de grupos, distribuição de recursos e a permanente necessidade de troca de informação), incrementaram a importância e a procura deste tipos de aplicações, tornando-as estratégicas para as organizações.

Existem muitas arquiteturas proprietárias para este tipo de aplicações colaborativas, no entanto, entendeu-se que se devia optar por soluções recomendadas por instituições de normalização internacional: ITU⁸ e IETF⁹.

Há dois tipos principais de controlo da conferência: fortemente acoplada/formal e fracamente acoplada/informal, por vezes com alguma mistura associada. A abordagem fortemente acoplada tem um mecanismo de controlo centralizado e com autoridade, administrando os membros e o acesso à conferência. Contrariamente, a abordagem fracamente acoplada tem um controlo distribuído, não necessitando de um mecanismo

⁸ International Telecommunication Union

⁹ Internet Engineering Task Force

explícito de controlo dos membros e das aplicações. A parte da moderação das conferências é uma área muito interessante, mas foge ao âmbito deste trabalho.

Tipicamente, uma conferência fracamente acoplada consiste em sessões multiponto com *streams* de áudio e vídeo, suportadas pelos protocolos RTP e RTCP (ver anexo I) usando *multicast* IP, como é o caso das comunidades Internet com a rede Mbone [Handley96].

Um grande número de características das aplicações de conferência, é comum a vários cenários. O uso de funcionalidades genéricas simplifica e incrementa o desenvolvimento deste tipo de aplicações, criando condições para o recurso a modelos normalizados, possibilitando interligação de sistemas diferentes.

De uma forma genérica, os sistemas de conferência normalizados (*whiteboard*, áudio e vídeo) caracterizam-se por [Helbig97]:

Transmissão de dados	<ul style="list-style-type: none"> • <i>unicast/multicast</i>, • serialização-<i>marshalling</i> de dados, • prioridades, • endereçamento e localização transparentes, • <i>internetworking</i>.
Gestão de conferências	<ul style="list-style-type: none"> • <i>set-up</i> e término de uma conferência, • administração de bases de dados, • junção e separação de conferências, • gestão de recursos, etc.
Suporte de aplicações distribuídas	<ul style="list-style-type: none"> • sincronização, • <i>conductorship</i>.

Quadro 5 - Características principais dos sistemas de videoconferência [Helbig97].

No ponto "gestão de conferências" encontra-se uma característica relevante para o trabalho, trata-se da gestão de recursos. A maior parte das aplicações *multicast* delega no utilizador essa tarefa não trivial. Obviamente nem todos os utilizadores possuem os conhecimentos técnicos suficientes para determinar os recursos de rede e processamento disponíveis, logo dotar a aplicação da capacidade de se auto-adaptar às condições presentes no sistema final de forma transparente para o utilizador, seria um avanço importante.

No processo de ensino à distância, as videoconferências apresentam vantagens e desvantagens.

4.2.1. Benefícios

A videoconferência influencia a aprendizagem dos alunos das seguintes maneiras [Reed98]:

- Eleva a motivação: os alunos ficam entusiasmados por utilizarem uma nova tecnologia para interagir com professores e outros alunos remotos.
- Aumenta a capacidade de comunicação e de apresentação: os estudantes consideram os "visitantes" importantes e ficam mais conscientes da importância de aparecer e falar bem. Além disso, ao planear e preparar uma videoconferência, os estudantes desenvolvem a capacidade de comunicação e de gestão do processo: "Os estudantes vêem-se no écran e percebem que estão a ser vistos da mesma maneira do outro lado. Ao longo do curso, notei mudanças de postura, de atitude e até na maneira de se vestirem. E tudo para melhor." (Paul Massmann, Concordia University Irvin)
- Aumenta o contacto com o mundo externo: muitas vezes uma visita ao vivo não é possível e, assim, o aluno tem a possibilidade de manter contacto com pessoas distantes e, às vezes, bem diferentes dele. No entanto, recomenda-se que, se a relação se prevê contínua entre os participantes de uma videoconferência, seja realizado pelo menos um contacto presencial.
- Aumenta a profundidade da aprendizagem: os estudantes aprendem a fazer melhores perguntas e a aprendizagem dá-se a partir de uma fonte primária, em vez de livros.

4.2.2. Problemas

Pela sua natureza, a videoconferência encerra diversos problemas [ChenDing95]:

- O ideal de videoconferência: é necessário permitir a visão do orador (emissor) a todos os participantes (receptores). É necessário conseguir uma forma de enviar vídeo a múltiplos receptores eficientemente.

- **Organização:** estabelece como organizar a videoconferência. O que fazer quando novos participantes decidem agregar-se, ou outro grupo inicia uma conferência? Haverá conflitos entre grupos?
- **Realtimeness:** cada participante deve ter, em tempo real, a percepção dos acontecimentos. É impraticável que exista um lapso de tempo significativo antes do vídeo ser recebido.
- **Qualidade de Serviço (QoS):** possibilitar uma imagem razoavelmente nítida; Garantir uma animação decente (uma taxa de imagens por segundo aceitável); etc.;
- **Lidar com a heterogeneidade:** como satisfazer necessidades diversas dos vários participantes que podem operar com equipamentos diferentes. *Sites* rápidos (com ligações rápidas) devem receber imagens com maior definição enquanto que aos *sites* mais lentos deve garantir-se a recepção das imagens essenciais.

4.3. Aplicações *multicast* orientadas à videoconferência

No passado recente, um esforço colaborativo da comunidade de investigação em comunicações por computador produziu uma gama de ferramentas para conferência multimédia sobre a Internet, sobretudo em dois projectos inovadores, resultantes da colaboração entre duas Universidades, Londres e Califórnia.

Em Dezembro de 1992 principia na UCL¹⁰, o projecto MICE (*Multimedia Integrated Conferencing for Europe*), sucedido em 1995 por um projecto mais envolvente, o MERCI (*Multimedia European Research Conferencing Integration*) e em 1997 pelo MECCANO (*Multimedia Education and Conferencing Collaboration over ATM Networks and Others*), sendo responsáveis por desenvolvimentos significativos ao nível da rede Mbone Europeia e das ferramentas *multicast* que usam essa rede virtual.

Nos laboratórios Berkeley da Universidade da Califórnia, o projecto MASH (<http://www.openmash.org/mash/>), deu também um grande contributo no desenvolvimento das conferências multimédia na Internet, nomeadamente com a

¹⁰ University College of London

implementação de ferramentas *multicast* em parceria com a UCL, algumas delas usadas neste trabalho.

As primeiras aplicações a fornecer vídeo sobre a Internet foram, em primeiro lugar, uma ferramenta produzida pela Xerox PARC, denominada por *nv* - *Network Video* [Frederik94I], e o INRIA *Videoconferencing System*, *ivs* [Bolot94]. O *ivs*, que combina áudio e vídeo na mesma aplicação, foi uma boa contribuição nesta arena, mas não teve a mesma popularidade do *nv*.

Enquanto estes dois sistemas partilhavam o objectivo de suportar vídeo sobre a Internet, distribuído em *multicast* a baixas taxas de transferência, as formas como resolviam o problema eram marcadamente diferenciadas. O *ivs* é um sistema integrado de áudio/videoconferência que assenta exclusivamente em H.261 [ITUT93] para compressão de vídeo. Em contraste, o *nv* é uma aplicação unicamente para transmissão de vídeo, que utiliza um esquema de codificação próprio, arquitectado especificamente para a Internet e direccionado para implementação de software eficiente [Frederick94II]. Resultante da sua baixa complexidade computacional, o *codec nv* pode correr muito mais rápido que o *codec* H.261 [Handley93], mas em contrapartida exige mais da rede.

Inevitavelmente, em trabalho pioneiro como o *nv* e o *ivs*, devem ser impostas restrições no processo de desenho para facilitar a experiência. Sendo que, ambos sistemas são baseados em software de compressão, os modelos de captura de vídeo são desenhados à volta de modelos que não utilizam compressão.

O mais popular arquivo de aplicações para CMM está acessível em <http://mice.ed.ac.uk/mice/>. Aqui é possível obter aplicações *multicast* académicas de domínio público, que vão desde ferramentas de difusão selectiva dos mais variados media (áudio, video, imagens, texto e gráficos) até ferramentas de monitorização de tráfego *multicast*, passando pelas aplicações de controlo de conferências que permitem organizar as sessões *multicast*. Realce-se que estão disponíveis versões para as plataformas mais comuns.

4.3.1. *rat - robust audio tool*

O *rat* foi desenvolvido na UCL [Hardman95], está concebido para utilização em ambientes de áudio-conferência *unicast* e *multicast*. Foi pensado para interagir com outras ferramentas para transmissão de áudio e apresentado para múltiplas plataformas. Requer apenas uma placa de som, colunas de som e microfone. Mais detalhes e o próprio software, de domínio público e *open-source*, estão disponíveis para as plataformas: FreeBSD, HP-UX, IRIX, Linux, NetBSD, Solaris, SunOS e Windows 9X/NT, em <http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/>.

Neste trabalho o *rat* será a ferramenta utilizada pelos clientes para a sua participação via áudio. As razões decisivas foram: a possibilidade de pré-parametrização, ser uma ferramenta de domínio público que está acessível a todos gratuitamente, possuir um desempenho melhorado no tocante à perda de pacotes [Hardman95], e finalmente, tem um cariz académico que nos interessa, nomeadamente pelo fornecimento de estatísticas de funcionamento que podem ser úteis para avaliar o desempenho final do sistema.

Cumulativamente, o *rat* não é uma ferramenta acabada, a cada nova versão são introduzidas melhorias e novos formatos de codificação mais avançados.

4.3.1.1 Codificação de áudio

No âmbito deste trabalho, interessa descrever sumariamente os tipos de codificação que o *rat* assume, pois será eventualmente necessário, no exercício da adaptação, pré-parametrizar a aplicação para utilizar um sistema de codificação e transmissão mais económico em termos de carga da rede, encontrando uma solução de compromisso entre a qualidade e o "preço a pagar".

No *rat* (versão 3) existem disponíveis cinco tipos de codificação áudio: 16-bit linear, PCM, DVI, GSM e LPC.

A técnica PCM, *Pulse Code Modulation*, é a mais simples, definida na norma ITU G.711. O sinal é amostrado 8000 vezes por segundo, permitindo que sejam guardadas digitalmente, frequências até 4kHz, algo restritivo, já que o discurso humano pode chegar aos 10kHz. Ainda assim, a maioria dos sons registados nas conversas humanas pertencem às frequências mais baixas e os 4kHz são razoáveis. O PCM requer

uma taxa de transmissão de 64 kbit/s. O PCM básico usa 8 bits, se forem usados 16 bits temos o modo "16-bit linear", que necessita de 128 kbit/s.

A simplicidade do PCM não significa eficiência, este método não aproveita as muitas redundâncias naturais que existem no discurso humano, nomeadamente as pausas. O ADPCM (*Adaptive Differential Pulse Code Mode*) (ITU G.723) tira partido das características peculiares do discurso humano, efectuando uma compressão do sinal sem apreciável perda de qualidade no registo digital, necessitando de apenas 32 kbit/s. O formato DVI, aquele que é usado por defeito no *rat*, é muito similar ao ADPCM, menos oneroso em termos de processamento e necessita também de 32 kbit/s.

As técnicas mais recentes tiram maior partido do conhecimento, ficando do lado do *software* de codificação/descodificação (*codec*), a tarefa mais "pesada", minimizando a informação que é efectivamente necessário transmitir. O formato GSM (*Global System for Mobile Communications*) [Mouly93] (usado nos telemóveis) é especialmente vocacionado para modelação de voz e reduz a necessidade de transmissão para 13.2 kbit/s.

O LPC (*Linear Predictive Coding*) é o formato mais compacto, opera como um sintetizador de voz, quase como um robot que repete as nossas palavras, utiliza tão somente 5.6 kbit/s.

Todos estes formatos de codificação áudio, podem ainda, ser combinados com diferentes tamanhos de pacotes em que se codifica o som.

De acordo com o tempo de áudio contido em cada pacote, o *rat* oferece quatro opções diferentes para o tamanho dos pacotes de áudio a transmitir: 20 ms, 40 ms, 80 ms e 160 ms. Pacotes maiores aceleram a transmissão por significar menos *overhead*, no entanto as eventuais perdas são mais graves porque afectam a inteligibilidade do discurso. O *rat* utiliza por defeito pacotes DVI, contendo a duração áudio de 40 ms, cabendo ao utilizador, a decisão de alterar esse valor na interface gráfica.

4.3.1.2 Parâmetros de configuração do *rat*

O *rat*, na sua interface gráfica, permite ao utilizador ajustar os parâmetros que modelam as suas comunicações, no entanto esse ajuste é exercido com a aplicação já em execução, algo que pode ser tardio e que certamente não será a solução ideal. O *rat*

possibilita também, pré-parametrização por linha de comandos, regulando o seu comportamento desde o início da execução.

A sinopse básica para executar o *rat* é a seguinte:

```
rat [options] addr/port
```

Quando se inicia uma sessão no *rat*, fornecem-se necessariamente dois parâmetros, o endereço *multicast* (para o âmbito deste trabalho) e o porto, toda a restante configuração da ferramenta tem valores que são assumidos por defeito. Esses valores por defeito podem constituir um aparente facilidade na utilização da ferramenta, mas efectivamente resultam numa execução "cega" às condições existente e que pode por em causa o êxito da sessão.

É objectivo deste trabalho introduzir auscultação às condições existentes e adaptação na execução das aplicações, e o áudio é a componente mais crítica das CMMs. Assim é possível, que o *middleware* (ver **Figura 12**) reconfigure a aplicação para que esta se execute em condições a prestar o melhor serviço possível sem comprometer o objectivo magno, a comunicação.

Os parâmetros que o *rat* aceita são numerosos, vão desde o *tll* (*time-to-live*), às opções de segurança (encriptação dos dados), podem ser consultados em (<http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/documents/rat-man.txt>), no entanto o único parâmetro que nos interessa para reconfiguração, é o tipo de codificação áudio empregue. Fazendo adaptação desse parâmetro consegue-se transmitir áudio com qualidade em conformidade com as possibilidades aferidas na rede e sistema final. Eis a sintaxe e semântica do parâmetro.

```
-f c1/c2/./cn
```

Especifica o tipo de codificação usado na transmissão áudio. A ordem é: tipo de codificação primário, secundário, etc. Os valores que representam os tipo disponíveis são: *l16*, *pcm*, *dvi*, *gsm* e *lpc* (ver secção 4.3.1.1).

A forma como essa adaptação é processada será exposta no Capítulo 6 referente ao "Trabalho Desenvolvido".

4.3.2. *vic - videoconferencing tool*

Surgiu em 1995, posteriormente ao *nv* e *ivs*, um terceiro modelo para aplicações de empacotamento de vídeo, uma ferramenta de videoconferência, *vic*, *video conferencing tool* [McCanne95], desenvolvida na U. C. Berkeley / LBL. Este modelo retira ensinamentos dos dois que o precederam, focalizando especialmente a flexibilidade. É uma aplicação de *framework* extensível, orientada a objectos, que suporta:

- múltiplas abstrações de rede;
- *codecs* baseados em hardware;
- um modelo de coordenação de conferência;
- uma interface extensível;
- diversos algoritmos de compressão de vídeo.

O *vic* foi originalmente concebido como uma aplicação para demonstrar os protocolos *real-time* Tenet¹¹ e simultaneamente para suportar a arquitectura das emergentes "*Lightweight Sessions*" no Mbone [Jacobson94].

O sistema *vic*, dedicado à conferência de vídeo, deve ser utilizado em conjunção com uma ferramenta para a conferência de áudio, normalmente o *rat* (descrito na secção anterior).

O *vic* suporta diversos tipos de compressão vídeo, desde: MPEG, JPEG, NV, C-You-See-Me, Sun Cell, H.261, Intra H.261, H.263, etc. [McCanne97]. Esta pletora de formatos confere-lhe grande flexibilidade, já que um utilizador do *vic* está habilitado a juntar-se a qualquer videoconferência do Mbone pois, praticamente todos os formatos estão suportados.

Dos *codecs* de vídeo disponibilizados pelo *vic* (versão 2.8), os mais adequados para os propósitos da aplicação a desenvolver serão o H.263 (norma ITU H.263-0106) e o H.261. O primeiro para sessões com poucos recursos, por ser um formato muito compacto (requer baixa largura de banda), e com qualidade razoável. O H.261 para sessões com suficiência ou abundância de recursos, pois comporta-se de forma adequada às sessões de *e-learning*, no sentido em que é económico em termos de recursos de largura de banda, para transmissão de cenas pouco movimentadas (uma

¹¹ Grupo de investigação em comunicações por computador da Universidade da Califórnia.

palestra, ou mostrar um esquema). Proporciona boa qualidade da imagem como da animação.

O *NV* e o *JPEG* (norma ITU R.601) são formatos mais onerosos em termos de consumo de largura de banda. Poderão ser mais económicos em termos de carga do CPU, mas o recurso mais escasso no panorama actual dos utilizadores da Internet é a largura de banda.

Em [Frohberg96] pode encontrar-se, de forma mais alargada, a justificação para a escolha dos formatos de vídeo, trata-se de um estudo sobre o comportamento do tráfego vídeo, onde, entre outras contribuições, se compara o desempenho, em termos de largura de banda, dos diferentes formatos vídeo suportados pelo *vic*.

Em termos de desempenho, o *vic* precisa de 1.1 Mbps para atingir uma qualidade de transmissão de 30 *fps*¹² (imagens por segundo). Se quisermos emitir com a máxima qualidade de imagem permitida; 24 bits de cor, máxima resolução e 30 *fps*, o volume de informação gerado pode ser muito elevado, i.e., cerca de 27 megabytes apenas num segundo de transmissão [Crowcroft99]. A qualidade da transmissão pode ser ajustada na interface (ver **Figura 13**) ou por linha de comandos.

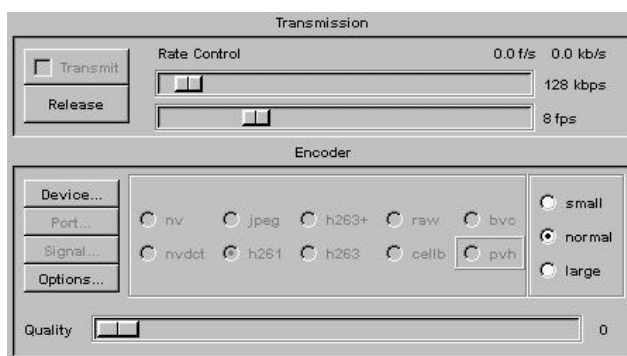


Figura 13 - Ajuste de parâmetros de transmissão na interface do *vic*

Tal como acontece com o áudio, o vídeo pode ser adaptado às condições de operação (rede/processamento), o *vic* dispõe de uma série de opções aplicáveis à configuração da aplicação antes de iniciar a transmissão. Essas opções podem afectar a qualidade da transmissão quer ao nível da cadência (taxa de imagens por segundo), do formato de codificação empregue e da resolução da própria imagem.

No *vic* executado por defeito, a parametrização não é casuisticamente ajustada, nem se torna fácil, para um mero participante numa videoconferência, fazer o ajuste avalizado. Os parâmetros que afectam a qualidade devem ser computados com

¹² a norma para emissões televisivas nos EUA, enquanto que na Europa é 25 *fps*.

acuidade, visando obter a melhor configuração em termos de garantia de um desempenho sustentado da comunicação. O trabalho a desenvolver procura solucionar este problema integrando as aplicações para CMM de forma adaptada a cada cliente (ver **Figura 12**).

Para informação mais alargada e obtenção do software, a *home page* do *vic* é a seguinte: <http://www-nrg.ee.lbl.gov/vic/>.

O *vic* suporta pré-parametrização, o que é essencial para conseguir a adaptação, vejamos o que pode ser ajustado.

4.3.2.1 Parâmetros de configuração do *vic*

O modo de funcionamento da aplicação de transmissão/recepção de vídeo, o *vic*, pode ser pré-programado segundo a seguinte sintaxe:

```
vic [ -A proto ] [ -B kbps ] [ -C conference ] [ -c dither ] [ -D
device ] [ -d display ] [ -f format ] [ -F fps ] [ -H ] [ -I channel ]
[ -K key ] [ -M colormapFile ] [ -m mtu ] [ -N sessionname ] [ -o
clipfile ] [ -P ] [ -s ] [ -t t1 ] [ -U interval ] [ -u script ] [ -V
visual ] [ -X resource=value ]
dest/port[:rport][/format/t1/nchan,dest2/port2[:rport2]/format2/...]
```

Os parâmetros representativos para este trabalho, no aspecto da adaptabilidade, são os seguintes, vistos em detalhe:

- B** ajusta o máximo de largura de banda disponível (*kilobits per second*).
- c** ajusta a paleta de cores, é dispensável nos casos onde abundam as condições de vídeo. Gray - para tons de cinzento
- f** usa a codificação de vídeo desejada, uma das seguintes (os mais usados):

h261	Intra H.261;
h263	H.263;
jpeg	Motion JPEG;
...	...
- F** ajusta o máximo de imagens por segundo (*frames per second*).

4.3.3. JMF - Java Media Framework

O JMF, instanciado no *JMStudio* é uma solução para media-conferência via Internet, desenvolvido pela *Sun Microsystems*, que assenta na plataforma Java e que concorre em potencialidades e qualidades com as aplicações anteriormente descritas, o

rat (áudio) e o *vic* (vídeo). Por absorver completamente a filosofia da Internet (sobretudo a independência da plataforma utilizada), e igualmente por demonstrar uma estabilidade superior à do *vic*, este *framework* representará em boa medida a plataforma de testes das actividades a desenvolver no âmbito deste trabalho.

O JMF fornece uma arquitectura unificada e *messaging protocol* para gerir a aquisição, processamento e entrega de media *time-based*¹³. O *framework* é concebido para suportar os tipos de conteúdos normalizados, tais como AIFP, AU, AVI, GSM, MIDI, MPEG, QuickTime, RMF e WAV.

A *home page* do JMF é a seguinte: <http://java.sun.com/products/java-media/jmf/>.

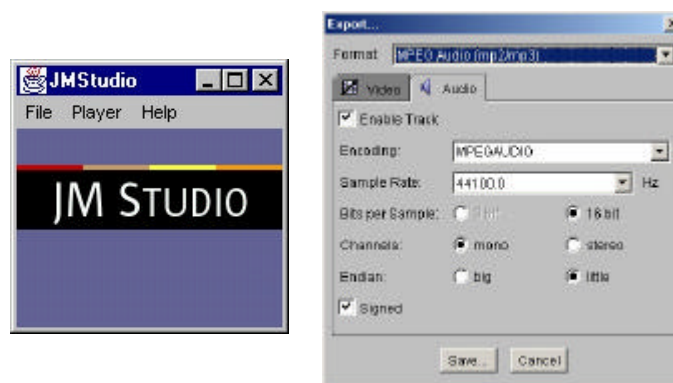


Figura 14 - Aspecto da configuração para emissão áudio no *JMF Studio*.

O JMF Studio é uma aplicação que permite exportar áudio e vídeo, usando ficheiros ou transmissão ao vivo. Permite igualmente abrir sessões RTP, com endereços *unicast* ou *multicast*. Faculta formatos de codificação para vídeo e áudio muito actuais e muito compactos, como o MP3, patente na **Figura 14**. Ultima-se a disponibilização do formato MPEG-4, que contribuirá para o melhoramento do desempenho geral, consequência da maior compressão do tráfego. Infelizmente não admite pré-parametrização, o que nos leva a utilizá-la somente no servidor de conferências, já que é impraticável, perante as directivas de usabilidade, integrá-la no sistema de EAD no lado cliente, excepto se o aluno a preferir, mesmo com o ónus de ter de a despoletar e configurar.

Com o JMF, é possível com facilidade, criar *applets* e aplicações que apresentem, capturem, manipulem e armazenem media *time-based*. As possibilidades de expansão estão totalmente abertas à comunidade, já que a filosofia Internet que norteia a distribuição desta ferramenta fomenta o auto-desenvolvimento aproveitando o *open-*

¹³ Informação que requiera entrega e processamento com tempo crítico.

source. Todas as vantagens que advêm da ferramenta ser implementada em linguagem Java (sobretudo a portabilidade) são extremamente apreciadas no âmbito da heterogeneidade de plataformas que se espera englobar.

Neste trabalho procedeu-se a adopção de *applets* que recorrem à API JMF [JMF-API99], proporcionando ao aluno uma recepção de áudio e vídeo de forma integrada no próprio *browser*. Assim, o aluno não tem necessidade de instalar, qualquer aplicação extra para assistir (só recepção) a palestras *on-line* ou visualizar material pedagógico a pedido. Basicamente trata-se de *applets* que executam o módulo "Player" da API JMF para objectos RTP, um para cada sessão *multicast*, áudio e vídeo. Este processo é descrito no Capítulo 6, secção 6.3.1.5.

4.3.3.1 Arquitectura de alto-nível do JMF

O JMF mimetiza o modelo básico do sistema de transmissão tradicional dos audiovisuais domésticos:

Câmara de Vídeo → Cassete Vídeo → VCR → Dispositivos de Saída (Monitor, Colunas Som)
(Disp. de captura) (Data Source) (Player) (Destino)

Uma *data source* encapsula o *stream* de media tal como uma cassete de vídeo e o módulo *player* provê processamento e mecanismos de controlo similares a um VCR (*Video-Cassette Recorder*). Capturar, transmitir e reproduzir áudio e vídeo com o JMF requer dispositivos adequados, tais como microfones, câmaras, altifalantes e monitores.

Seguidamente, na **Figura 15**, apresentam-se num modelo de camadas, os constituintes da arquitectura em análise. Destaque para a flexibilidade e extensibilidade inerente às diferentes APIs, requerida para suportar aplicações multimédia avançadas e futuras tecnologias media.

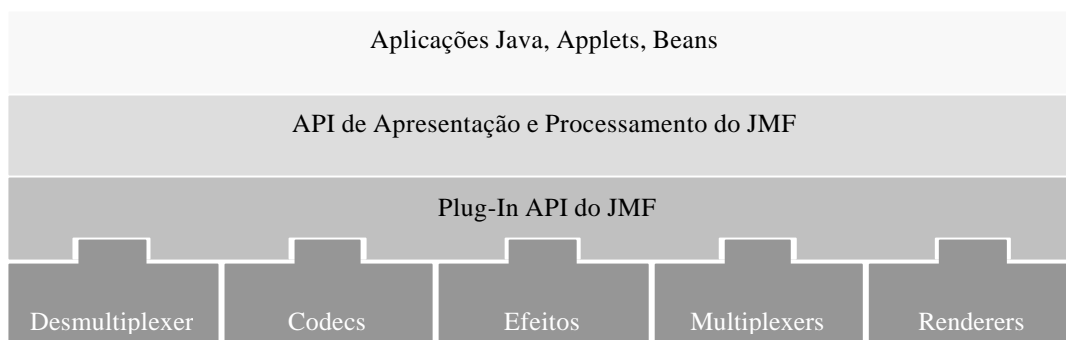


Figura 15 - Arquitectura de alto nível do JMF.

4.4. Ferramentas para espaço de trabalho partilhado - *Whiteboard (wb)* e *network text editor (nte)*

O *whiteboard (wb)* é uma aplicação *multicast*, bastante completa para trabalho colaborativo, como a própria palavra indica é um "quadro branco" onde cada participante pode contribuir, expressando as suas ideias recorrendo a grafismos e texto. Estão disponíveis, tal como num editor gráfico vulgar, várias ferramentas para desenho, vários tipos de traço, várias cores, polígonos, etc. Para complementar é possível adicionar texto. Os participantes podem editar todo o espaço de trabalho partilhado, adicionando ou apagando objectos. Mais detalhes e obtenção do software em: <http://www-mice.cs.ucl.ac.uk/multimedia/software/wb/>.

O *nte* é um editor de texto concebido para funcionar no Mbone, inicialmente desenvolvido por Mark Handley como parte dos projectos MICE e MERCI. É uma ferramenta de partilha de espaço de escrita. Cada participante pode ver e colaborar na elaboração de textos. O mesmo bloco de texto pode ser emendado e apagado por qualquer participante, excepto quando o autor o proteja de edição alheia, o que pode desvirtuar o sentido de trabalho colaborativo mas impede que participantes mais indisciplinados ameacem o trabalho dos demais. Mais detalhes do *nte* e respectivo *download* acessíveis em: <http://www-mice.cs.ucl.ac.uk/multimedia/software/nte/>.

4.5. Integrar as diferentes aplicações *multicast*

Com a necessidade de integrar as diversas aplicações *multicast* que foram desenvolvidas, para em complementaridade constituírem um ambiente virtual de comunicação hipermédia, foi necessário centralizar numa aplicação a interface entre os utilizadores/sessões/aplicações. Assim, as sessões destinadas ao consumo público no Mbone, são anunciadas na sua maioria através do *sdr (session directory)* [Handley95]. O *sdr* é uma ferramenta *multicast*, que lista as sessões activas e futuras, e permite aos utilizadores aderirem a essas sessões. Paralelamente, despoleta as aplicações envolvidas na transmissão da sessão sempre que o utilizador não manifeste a intenção de aceder apenas a algum tipo de media. Todo este cenário é visível apenas para os utilizadores

que se encontram dentro de um determinado âmbito, para o qual essas sessões estão a ser veiculadas.

As aplicações que normalmente constituem a base de uma sessão no Mbone, já descritas neste capítulo, são as seguintes:

- *rat* (*robust audio tool*) - audioconferência;
- *vic* (*videoconferencing tool*) - videoconferência;
- *wb* (*whiteboard*) - conferência gráfica;
- *nre* (*network text editor*) texto-conferência.

No sistema de EAD desenvolvido a integração das aplicações é feita de forma diferente, sem recurso ao anúncio de sessões pelo *sdr*. O *middleware*, que interage com uma base de dados de sessões calendarizadas, irá despoletar as aplicações *multicast* (*rat*, *vic*, *wb* e *nre*) de forma transparente e adaptada para o utilizador. No Capítulo 6, onde se apresenta o trabalho desenvolvido serão dados todos os pormenores.

4.6. Avaliação das aplicações para CMM (conferência multimédia *multicast*)

Os e-alunos que se pretende venham a utilizar o sistema de EAD a desenvolver, deverão ser capazes de lidar com as aplicações apresentadas neste capítulo, quer para comunicação áudio, vídeo e outras que se agregam como aplicações de espaço de trabalho partilhado. Essas aplicações foram, principalmente, concebidas pela e para utilização da comunidade de investigação em redes de computadores, e portanto não se aprimorou particularmente a usabilidade.

No projecto MERCI da UCL, já referido no início deste capítulo, foi realizado um estudo para avaliar a usabilidade das ferramentas *multicast*. A aplicação de videoconferência, *vic*, foi uma das avaliadas. Conclui-se que a usabilidade do *vic*, ao nível da interface e configuração das opções, era bastante confusa para novatos e mesmo para utilizadores experientes [Clark97]. Estudos realizados noutras ferramentas do Mbone, tais como o *rat* e o *sdr*, revelaram resultados similares [ClarkSasse97, Bouch97].

Os maiores óbices estavam radicados na interface, que não fora pensada para utilizadores leigos. Por outro lado, a inexperiência em participação em

videoconferência, poderia levar ao caos organizativo das muitas janelas dispersas pela área de trabalho, note-se que existirá por cada participante vídeo-activo uma janela com a recepção do seu sinal vídeo, a somar a estas estão: uma por cada ferramenta de espaço de trabalho partilhado (*wb*, *nte*) e ainda outra para a áudioconferência (*rat*). No total serão, possivelmente, mais de uma dezena de janelas a que será necessário atender.

Ao nível da interface, uma solução seria integrar de forma "sensível" todas as fontes de media, ajudando o utilizador a gerir a sua área de trabalho, mas a comunidade de investigação nesta matéria ainda tem muito trabalho pela frente.

A usabilidade das aplicações *multicast* é apenas uma faceta da sua avaliação, a qualidade da comunicação permitida por estas ferramentas é pelo menos tão importante. A qualidade de uma CMM é afectada por uma miríade de factores (ver **Figura 16**), alguns desses factores, sobretudo ao nível da qualidade da comunicação e usabilidade das aplicações envolvidas, constituem a motivação deste trabalho. Um maior desenvolvimento na temática da qualidade de serviço em aplicações *multicast* está presente no Capítulo 5.

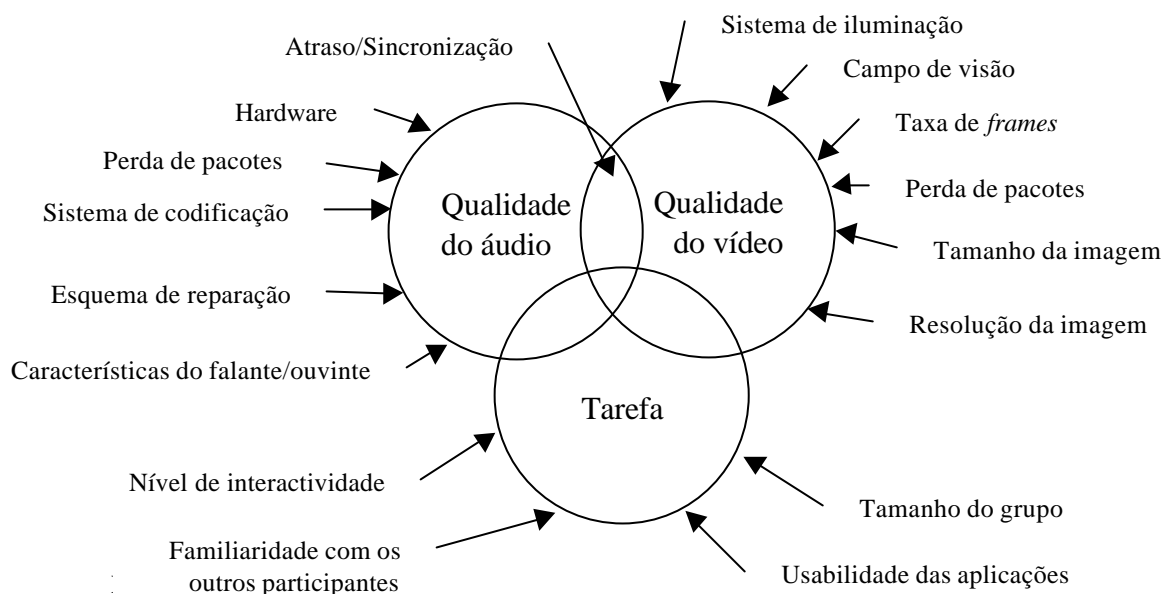


Figura 16 - Visão geral dos factores que influenciam a qualidade percebida numa CMM [Watson01]

Neste trabalho, ao nível das aplicações *multicast*, os objectivos são claros. Trata-se de integrá-las num sistema de ensino à distância baseado em WWW, favorecendo a sua usabilidade e introduzindo algum *middleware* que permita suprir algumas

deficiências constatadas, nomeadamente ao nível da adaptação aos recursos (rede e sistema final) disponíveis. Assim, as ditas aplicações serão pré-parametrizadas para actuarem de forma optimizada e sustentada.

Ao nível da usabilidade também serão dados alguns contributos. O e-aluno não necessitará de aprender pormenores técnicos das aplicações, estas serão executadas simplesmente premindo o botão principal do rato, quando o e-aluno está em condições de solicitar esse serviço, adicionalmente será computada, pelo *middleware*, a pré-configuração mais adequada para as condições verificadas.

No caso do e-aluno pretender apenas, assistir passivamente a uma e-aula, o sistema de EAD proporcionar-lhe-á essa experiência numa janela do seu *browser*, sem necessidade de recorrer a aplicações a correr noutras janelas, por intermédio de dois *applets*, um para receber o áudio e outro para o vídeo, que podem ser usados em conjunto ou por separado. Os pormenores de implementação são tratados no Capítulo 6, referente ao "Trabalho Desenvolvido".

No capítulo seguinte analisa-se a problemática da QoS para as aplicações descritas neste capítulo, procurando angariar soluções e prover modelos para garantir alguma QoS no sistema de EAD a desenvolver.

Capítulo 5

Qualidade de Serviço: a rede e as aplicações *multicast*

Com o advento das aplicações multimédia em tempo-real para comunicação em grupo, a concepção original da Internet revelou-se insuficiente. A necessidade de acomodar vários tipos de tráfego na ubiquidade do IP levou ao desenvolvimento de novos serviços de rede. As soluções passam pela introdução de uma nova filosofia que optimizasse a comunicação multiponto-a-multiponto, e por outro lado pela garantia da qualidade de serviço (*QoS - Quality of Service*)[Wang01].

Duas novas extensões foram então adicionadas ao IP. A primeira, já amplamente apresentada no Capítulo 3, versa sobre um novo modelo de comunicação multiponto-a-multiponto com a abstracção de grupo, denominado de *multicast* [Deering89]. A última,

que se pretende explicar neste capítulo, procura a introdução de novos serviços para suportar mais do que tráfego *best-effort*.

Duma perspectiva técnica, na literatura de *networking*, a temática central da *QoS* é então, a provisão de acesso previsível e coordenado aos recursos do sistema, envolvendo de igual modo os sistemas e a rede [Richards97]. Mais recentemente, o termo qualidade de serviço foi igualmente apreendido pela comunidade que desenvolve software e pelos seus utilizadores, usando-o de forma mais subjectiva, para caracterizar a utilidade e qualidade das aplicações. Por exemplo, a capacidade de uma aplicação se adaptar ao estado da rede pode ser considerado como um aspecto indirecto da sua qualidade [Miras02]. Ambas abordagens têm cabimento neste trabalho.

No caso específico do tráfego *multicast* IP, verificam-se problemas de escalabilidade com o crescente número de grupos activos que concorrem pelos meios de interligação disponíveis, e nesse sentido, vários estudos foram realizados para obviar o problema, destacando-se a utilização de reserva de recursos [Fourmaux98], *QoS* dinâmica [Acharya99], protocolos de adaptação *multicast* em redes activas [Yamamoto00] e *multicast* agregado [Gerla01].

O sistema desenvolvido, nomeadamente as aplicações que suportam o processo de ensino à distância, mormente os processos de *multicast* interactivo, como a videoconferência, vêm o seu eficaz funcionamento afectado pela concorrência pelos recursos de interligação disponíveis. No trabalho desenvolvido, mais do que reservar recursos, pretende-se preservar os recursos para os parâmetros críticos das aplicações. Adicionalmente, o sistema deverá ter um comportamento proactivo face aos recursos, sobretudo os da rede, prevenindo situações de sobrecarga.

Como o Mbone oferece as mesmas garantias de *QoS* que o próprio IP [Ferguson98], i.e., *best-effort*, então, se almejarmos obter *QoS* adicional deveremos enveredar pelas arquitecturas existentes e em desenvolvimento no IETF :

- *IntServ* ou *Integrated Services* (RFCs 1633, 2211, 2212, 2213);
- *DiffServ* ou *Differentiated Services* (RFCs 2474, 2475, 3260).

No campo das aplicações, o tratamento da qualidade de serviço, poderá revelar-se profícuo se atendermos ao facto de que num sistema de educação à distância, todo o acervo de material didáctico, desde sequências de vídeo, apresentações, imagens, etc., possam estar acessíveis em múltiplas qualidades, que possam satisfazer uma

comunidade heterogénea. O desenvolvimento do *site* e dos seus conteúdos, poderá então, ser orientado para o conceito de "conteúdos *web* dinâmicos" [Cheng99].

5.1. QoS "sobre" IP

O cenário actual das redes IP mudou. Hoje, o cenário de utilização das redes IP exige que "qualquer aplicação" possa correr com qualidade sobre o IP. Ou seja, a situação do IP actualmente é no sentido do "Tudo sobre IP" mantendo a premissa básica de projecto do "IP sobre Tudo" dos anos 70. De certa forma o paradigma mudou. A questão pertinente é a identificação das eventuais limitações do IP e procedimentos necessários para adequá-lo à nova realidade das redes.

Como veremos adiante, a qualidade de serviço em redes IP não é necessariamente resolvida com um único protocolo ou algoritmo. Na maioria dos casos e dependendo da necessidade da(s) aplicação(ões), um conjunto de novos recursos deve ser utilizado.

Os protocolos de transporte utilizados na Internet parece não se adequarem totalmente às necessidades das aplicações usadas neste trabalho, quer por serem demasiado penalizantes em situações anómalas (como seja o caso do TCP) quer por não incluírem os mecanismos imprescindíveis ao correcto funcionamento das aplicações de tempo-real. Por esta razão, surgiu complementarmente, a definição de novos suportes protocolares mais adequados a esta família de aplicações. Estes devem resolver com eficácia os problemas inerentes a este tipo de serviços como sejam: assegurar o isocronismo dos dados transportados, sincronização de fontes, adaptação a condições anómalas, controlo de atrasos, controlo do estado das ligações, etc.

As decisões arquitecturais tomadas na concepção do protocolo IP foram, na sua maioria, no sentido da simplicidade. Este paradigma de concepção impõe algumas restrições técnicas ao IP e, por consequência, restringiu as aplicações suportadas às aplicações com poucos requisitos de operação (por exemplo: aplicações de dados podem perder pacotes, permitem a existência de atrasos, ...).

Actualmente desenvolvem-se arquitecturas de QoS em IP que possam acomodar vários tipos de serviços.

No RFC 3052, que revê as arquitecturas de controlo de serviços, refere-se que as actuais arquitecturas para implementar QoS sobre IP não especificam se o serviço será

por aplicação ou por opção da camada de transporte. Tal facto, acarretará um impacto significativo em vários aspectos do serviço, nomeadamente em: controlo, garantias e cobrança. Um estudo sobre como deverá evoluir a arquitectura IP com suporte a QoS está publicado em [Huston00].

5.2. Parâmetros de Qualidade de Serviço (QoS) da rede

Para as futuras aplicações, especialmente as altamente interactivas e as que assentam em transferência de informação multimédia, é essencial que a QoS seja garantida em toda a extensão do sistema, incluindo a plataforma do sistema distribuído, o protocolo de transporte e a rede multiserviços. Protocolos de comunicação melhorados suportarão negociação de QoS fim-a-fim, renegociação e indicações de degradação de QoS. Será necessária coordenação sobre múltiplas conexões inter-relacionadas [Hutchison94].

A qualidade de serviço na Internet pode ser expressa pela combinação de factores impostos pela rede, tais como: largura de banda, atraso, *jitter*, fiabilidade e disponibilidade [Ferguson98]. Estes factores são comumente denominados de parâmetros de QoS e serão alvo de descrição mais apurada nas secções seguintes.

5.2.1. Largura de Banda

A largura de banda é a máxima taxa de transferência sustentada entre dois pontos terminais. Note-se que pode ser limitada, não só pela infra-estrutura física que corresponde ao caminho do tráfego na rede transitada, que provê o remanescente da largura de banda total, mas é também limitada pelos processos comunicativos concorrentes que partilham o todo ou parte do caminho fim-a-fim [Ferguson98].

A taxa de transmissão (largura de banda requerida) é o parâmetro mais básico de QoS. Influencia directamente a qualidade da transmissão, já que o número de bits que é "injectado" na rede, tem de ser ajustado em concordância pela aplicação. No **Quadro 6** mostra-se a taxa de transmissão tipicamente exigida por algumas aplicações.

Aplicação	Taxa de transmissão (Típica)
Aplicações Transacionais	1 Kbps a 50 Kbps
Quadro Branco (<i>Whiteboard</i>)	10 Kbps a 100 Kbps
Voz	10 Kbps a 120 Kbps
Aplicações Web (WWW)	10 Kbps a 500 Kbps
Vídeo (<i>Streaming</i>)	100 Kbps a 1 Mbps
Aplicação Conferência	500 Kbps a 1 Mbps
Vídeo MPEG [ISO/IEC92]	1 Mbps a 10 Mbps
Aplicação Imagens Médicas	10 Mbps a 100 Mbps
Aplicação Realidade Virtual	80 Mbps a 150 Mbps

Quadro 6 - Taxas de transmissão típicas para aplicações de rede [Martins99].

5.2.2. Atraso (*Delay*)

O atraso introduzido pela rede, corresponde ao lapso de tempo necessário, para que as unidades de informação da aplicação cheguem ao destino, desde a sua colocação na rede [Miras02].

Atraso é a combinação de: atraso de propagação (*propagation delay*), atrasos de processamento dos encaminhadores intermédios e atrasos variáveis de *queueing* nos *routers* até ao *host* destino. Demasiado atraso pode causar indisponibilidade da informação e ininteligibilidade em interacção multimédia, com consequências frustrantes para o utilizador.

O atraso de propagação corresponde ao tempo necessário para a propagação do sinal eléctrico ou óptico no meio utilizado (fibras ópticas, satélite, coaxial, ...) e é um parâmetro físico imutável.

Existem limites de tolerância ao atraso para as aplicações em foco neste trabalho, o **Quadro 7** sintetiza esses valores, vistos num único sentido, para o tipo de interactividade mais sensível, a áudio-conferência.

Atraso num sentido	Efeito na qualidade percebida
<100-150 ms	Atraso indetectável
150-250 ms	Ainda de qualidade aceitável, mas por ser próximo da fronteira, as pequenas variações podem produzir algumas falhas perceptíveis
Acima de 250-300 ms	Atraso inaceitável, a conversa normal é impossível.

Quadro 7 - Valores de referência para o atraso em áudio-conferência [Miras02].

5.2.3. Jitter

A variação no atraso ou *jitter* é usualmente devido ao armazenamento em filas de espera nos *routers*, durante períodos de tráfego intenso, ou esporadicamente devido a mudanças de encaminhamento devido a falhas, ou mesmo durante a actualização das tabelas de encaminhamento dos *routers* [Miras02].

Um *jitter* elevado provoca que os pacotes cheguem tarde demais ou fora de ordem, impossibilitando a sua descodificação, processamento ou visualização, o que equivale, em aplicações *real-time*, a considerá-los irremediavelmente perdidos.

Algumas aplicações toleram o *jitter*, outras fazem uso de técnicas de *buffering* para suavizar o efeito do *jitter*, evitando quebras no sincronismo [Sreenan00], no entanto existe um limite para todas (dependendo da tolerância ao *jitter* pela aplicação ou dos recursos de memória do *buffer*).

A **Figura 17** ilustra o efeito do *jitter* entre o envio de pacotes na origem e o seu processamento no destino. Observe-se que o *jitter* causa, não somente uma entrega com periodicidade variável (*Packet-Delay Variation*) como também a entrega de pacotes fora de ordem.

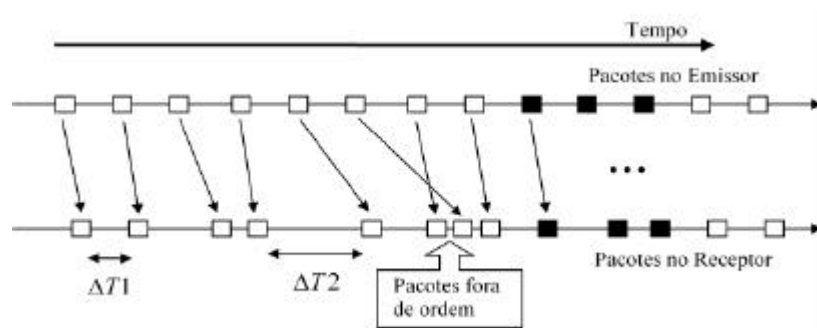


Figura 17 - Efeito do *jitter* para as aplicações [Martins99].

Na prática, a grande maioria das aplicações multimédia optam por utilizar o UDP ao invés do TCP pela maior simplicidade e menor *overhead* deste protocolo [Stevens94]. Nestes casos, o problema de sequenciação deve ser resolvido por protocolos de mais alto nível, normalmente incorporados à aplicação, como por exemplo, o RTP (*Real Time Transport Protocol*) descrito no RFC 1889. Detalhes do protocolo RTP podem ser consultados no Anexo I.

5.2.4. Fiabilidade ou perda de pacotes

A perda de pacotes é tipicamente o resultado da excessiva congestão na rede. A taxa de perdas corresponde ao rácio entre os pacotes que por múltiplas razões, apesar de serem transmitidos, se perderam desde a origem até ao destino e o total de pacotes transmitidos. É usual mensurá-la em percentagem de perda de pacotes.

A fiabilidade pode ser também um produto acessório do sistema de comutação, onde um sistema de comutação mal configurado ou com baixo desempenho pode alterar a ordem dos pacotes em trânsito, entregando pacotes ao receptor na ordem adulterada face à transmissão original, ou mesmo perdendo pacotes através de ciclos de *routing* transientes.

As perdas de pacotes em redes IP ocorrem principalmente em função de factores tais como:

- descarte de pacotes nos *routers* e *switch routers* (erros, congestionamento, ...);
- perda de pacotes devido a erros ocorridos na camada 2 (PPP - *Point-to-Point Protocol*, *ethernet*, *frame relay*, ATM, ...) durante o transporte dos mesmos.

De maneira geral, as perdas de pacotes em redes IP são um problema sério para determinadas aplicações como, por exemplo, a voz sobre IP. Neste caso específico, a perda de pacotes com trechos de voz digitalizada implica uma perda de qualidade eventualmente não aceitável para a aplicação. O que fazer em caso de perdas de pacotes é uma questão específica de cada aplicação, em [Bhattacharyya97] apresentam-se algumas soluções para controlar eficientemente o tráfego *multicast*.

Do ponto de vista da qualidade de serviço da rede, a preocupação é normalmente no sentido de especificar e garantir limites razoáveis (taxas de perdas) que permitam uma operação adequada da aplicação. No caso das transmissões *time-sensitive*, aquelas que mais interessam neste trabalho, os erros provocam distorção, que pode por em causa o entendimento no decurso de uma áudio ou videoconferência.

5.2.5. Disponibilidade

A disponibilidade é um aspecto da qualidade de serviço abordada normalmente na fase de projecto da rede e/ou no estabelecimento de um contrato de serviço. Em termos

práticos, a disponibilidade é uma medida da garantia de execução da aplicação ao longo do tempo e depende de factores tais como:

- disponibilidade dos equipamentos utilizados na rede privada (rede do cliente) (LAN, MAN ou WAN) e
- disponibilidade da rede pública, quando a mesma é utilizada (operadores de telecomunicações, ISPs - *Internet Service Providers*, ...).

As empresas dependem cada vez mais das redes de computadores para a viabilização de seus negócios (comércio electrónico, *home-banking*, atendimento *online*, transacções *online*, ...) e, neste sentido, a disponibilidade é um requisito bastante rígido. A título de exemplo, requisitos de disponibilidade acima de 99% do tempo são comuns para a QoS de aplicações *web*, aplicações cliente/servidor e aplicações de forte interacção com o público.

5.3. Parâmetros de QoS das aplicações

Da perspectiva do utilizador, os parâmetros de QoS da rede estão escondidos nos parâmetros de QoS ao nível da aplicação. O desempenho da aplicação depende da percepção da qualidade pelo utilizador e não advém do conhecimento técnico de como o serviço está implementado, nesta fase, a deterioração da qualidade pode ser imputável à rede mas também : ao sistema final, à câmara de vídeo usada, aos algoritmos de compressão de vídeo, ou ao microfone, por exemplo. Assim, os requisitos de QoS que afectam as aplicações podem ser vistos pelas seguintes métricas, que não coincidirão necessariamente com as da secção anterior:

Taxa de transferência

É a porção de largura de banda efectiva que a aplicação recebe da rede.

Latência

É o atraso fim-a-fim experimentado pela aplicação. A aplicação deve definir quantitativa e qualitativamente este requisito, por exemplo, "uma áudio-conferência deve decorrer com clareza".

Disponibilidade e continuidade de serviço

Descreve os requisitos de um serviço ininterrupto com qualidade aceitável. Existem vários factores que podem atentar contra a continuidade do serviço, entre eles, destacam-se:

- perda de informação;
- variabilidade dos atrasos;
- negação de serviço ou *Denial of Service (DoS)*;
- segurança.

Outros parâmetros mais difíceis de quantificar, mas não de menor importância, são os que se situam a montante e a jusante da própria aplicação no sistema final. Seguem-se alguns exemplos:

Falhas de equipamentos ou má configuração

Por exemplo, *codec* com compressão desajustada.

No caso das aplicações para ensino à distância, no caso da videoconferência, a resolução da imagem, o tamanho e até a iluminação podem ser factores que influenciam a qualidade [Miras02].

Incapacidade do Sistema Operativo

O SO não suporta correctamente a aplicação.

Defeitos no desenho da própria aplicação

Como é o caso da falta de sensibilidade e adaptabilidade às condições do meio (rede/sistema final) para aplicações *time-sensitive*. Por exemplo, uma aplicação de videoconferência pode enviar grandes quantidades de informação para a rede, mesmo quando esta se encontra congestionada.

Todos estes parâmetros terão importância na elaboração do modelo de QoS para a aplicação a desenvolver, sendo que o tópico da adaptabilidade será considerado capital.

5.4. Necessidade de QoS das aplicações envolvidas

Os requisitos das aplicações de áudio e vídeo interactivo, como as que são usadas no EAD, em termos de desempenho dos serviços de rede, são bastante exigentes, de modo a suportar a inteligibilidade e a interactividade. A ITU (*International Telecommunication Union*) produziu vários estudos e recomendações sobre a qualidade de comunicações: de voz (ITU-T-G.114), vídeo (BT.500.10) e de serviços sobre IP (ITU-T-Y1514). Se em termos de largura de banda, as necessidades oscilam entre as centenas de Kbps e as dezenas de Mbps, em termos de tolerância a perdas de pacotes, os limites situam-se abaixo dos 10% para o vídeo e são mais restritos para o áudio. O atraso também põe em causa a eficácia da comunicação interactiva. O gráfico da **Figura 18** é exemplificativo da influência destes requisitos de QoS.

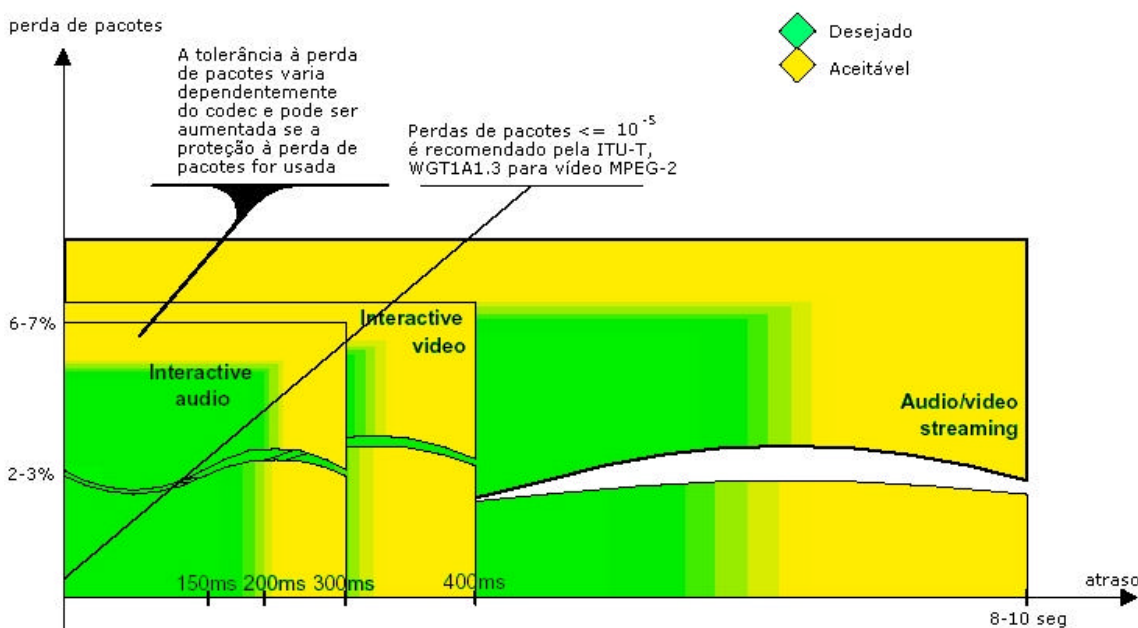


Figura 18 - Mapeamento dos requisitos de atraso e perda de pacotes para aplicações multimédia interactivas [Miras02].

Aplicar modelos de QoS em sistemas de difusão selectiva [Cui01], sobretudo estando em causa aplicações que debitam grandes quantidades de informação, com é o caso das conferências multimédia interactivas, tem-se revelado vantajoso [Biswas99, Campbell97, Fourmaux98].

As considerações que se seguem tentam identificar as exigências em termos de QoS das aplicações multimédia ilustrando algumas situações práticas.

Uma aplicação multimédia *offline* envolvendo, por exemplo, dados, gráficos e ficheiros com animação (vídeo, ...) não necessita de sincronização e, assim sendo, não necessita de "cuidados especiais", em termos de QoS da rede. No caso de dados correspondentes a uma animação que, em termos práticos, necessita de uma determinada taxa de transmissão, eventualmente carrega a rede, mas não exige atrasos, sincronização ou tempo de resposta. Este é um caso típico onde a necessidade de QoS se reduz a uma necessidade de taxa constante.

As aplicações que estão em foco neste trabalho, nomeadamente as aplicações interactivas para comunicação em grupo, necessitam da mais alta prioridade de transmissão comparativamente ao tráfego convencional, e cada aplicação é detentora de um nível de sensibilidade ao atraso e criticidade de missão, ver **Figura 19**.

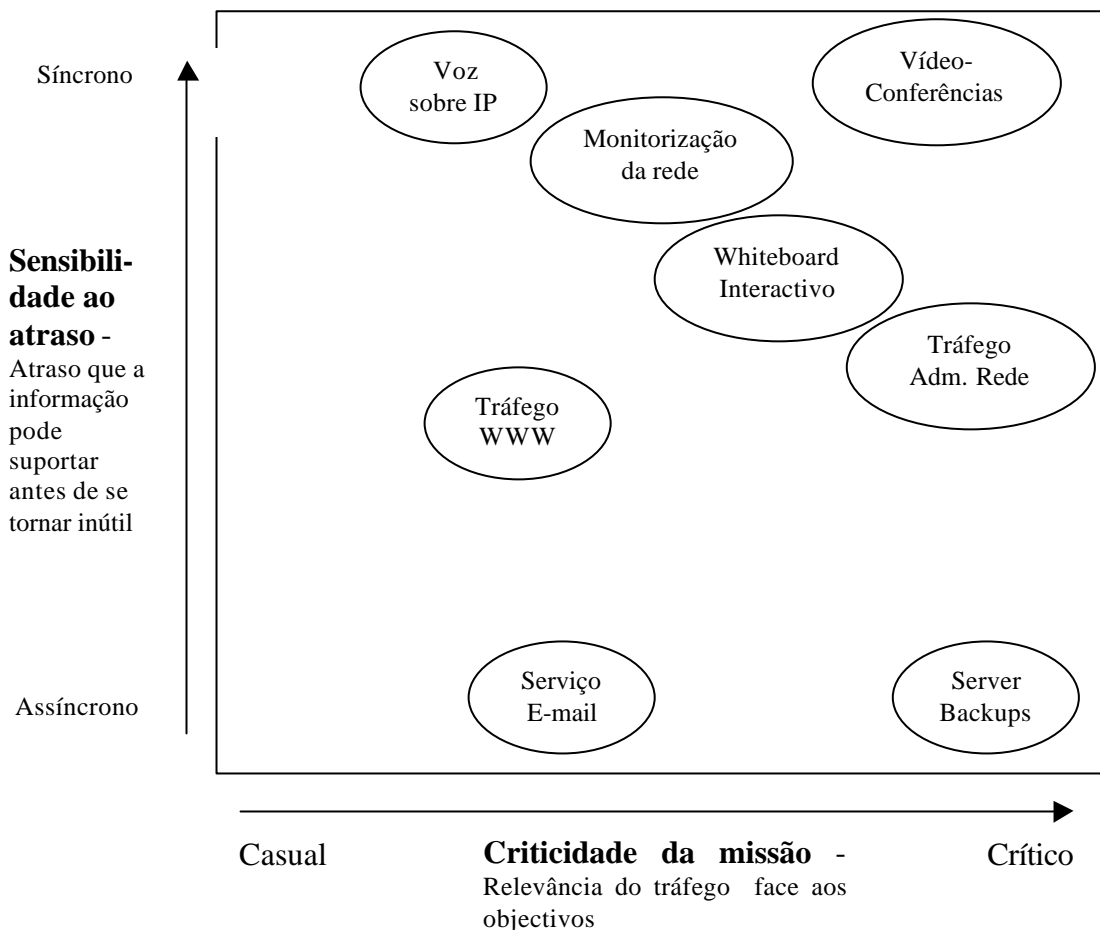


Figura 19 - Comparação entre sensibilidade ao atraso e criticidade do tráfego.

Por outro lado, para aplicações multimédia de conferência de áudio, as mais críticas, garantir apenas a taxa de transmissão não é suficiente. Neste caso específico, os

atrasos de comunicação e as perdas de pacotes influenciam a interactividade dos utilizadores e a qualidade da aplicação, algo que ficou patente no **Quadro 7**.

5.5. QoS - Alternativas técnicas

Actualmente, diferentes alternativas para prover e sinalizar serviços de comunicação em ambientes heterogéneos estão publicadas. Eis algumas como pistas de investigação complementar: o papel do emissor é fundamental na garantia de qualidade de serviço em ambiente heterogéneo, por isso, tal entidade deverá prover os conteúdos de forma adaptada às condições da rede [Ramanujan98]. Em [McCanne96] propõe-se que os receptores poderão ligar-se a grupos específicos para obter a qualidade de serviço desejada. Outras abordagens referem que o "esforço" pode ser introduzido no estabelecimento ao nível do transporte [Amir95] ou alternativamente nas *gateways* do nível de aplicação [Chawathe99].

Seguindo um princípio básico, a QoS deve ser configurável, previsível e passível de manutenção ao longo de todas as camadas arquitecturais para alcançar QoS fim-a-fim [Kurose93]. O fluxo atravessa os módulos fonte (e.g., CPU, memória, dispositivos multimédia, rede, etc.) e cada camada desde os dispositivos emissores de media, prosseguindo pela pilha protocolar da fonte, através da rede, chegando à pilha protocolar da entidade receptora e culminando nos dispositivos de reprodução finais.

A **Figura 20**, evidencia o âmbito da QoS na comunicação fim-a-fim.

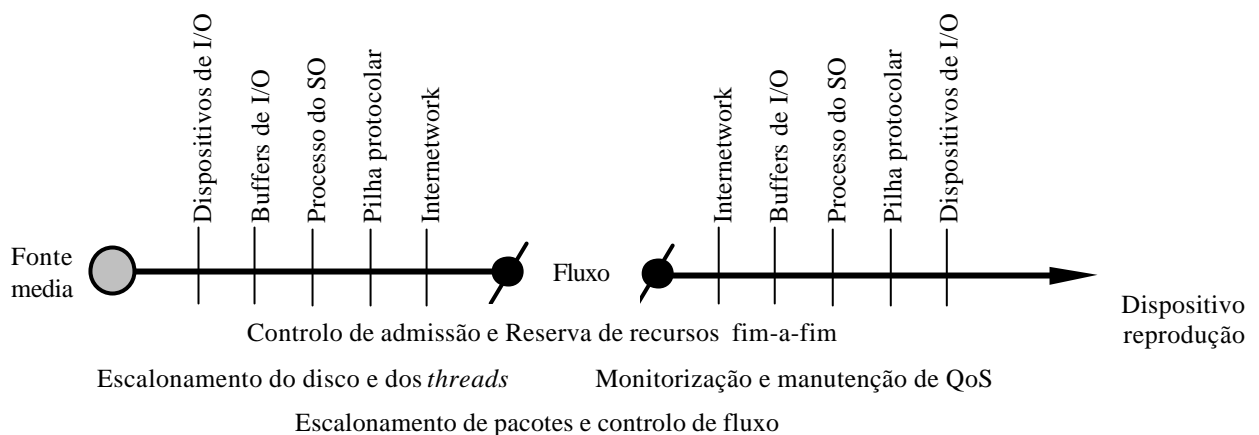


Figura 20 - Cenário de QoS fim-a-fim [Aurrecoechea98].

Para introduzir QoS ao nível da rede, o IETF propõe duas arquitecturas, baseadas em *frameworks* que compelem para a alocação de capacidades que suportem a garantia de recursos e a diferenciação de serviços. São denominadas de *Integrated Services (IntServ)* e *Differentiated Services (DiffServ)*, e provêm duas soluções distintas para alcançar QoS.

A arquitectura *IntServ* (RFC 1633), foi a primeira a desenvolver-se, no início dos anos 90, preconiza assegurar recursos para as aplicações, através de reserva de recursos para fluxos de aplicações individuais (*per-flow*), enquanto que a arquitectura *DiffServ* (RFC 2474, 3260), surgida mais tarde em 1997, usa uma combinação de estabelecimento de políticas de acesso, provisionamento e priorização de tráfego para conseguir uma diferenciação do serviço.

A arquitectura *IntServ*, designadamente o seu grupo de investigação, produziu valiosos contributos: novas arquitecturas de alocação de recursos, modelos de serviço e o protocolo de sinalização RSVP (*Resource Reservation Setup Protocol*), descrito no RFC 2205, que se assumiu como *standard* para a reserva de recursos ao longo do caminho (*path*) entre emissor e receptor. Vários trabalhos [Fourmaux98, Pullen99] associam a reserva de recursos ao *multicast*, foi a primeira via a ser explorada, actualmente, no seguimento do desenvolvimento da plataforma QBONE¹⁴ (Internet com QoS) para a Internet2, alguns investigadores estão a enveredar pela via *DiffServ* [Teitelbaum99].

A arquitectura *DiffServ*, produziu normas que possibilitam marcar os pacotes IP com QoS diferenciada. Empregando um conjunto de PHBs (*Per Hop Behaviours*) em cada *router*, onde cada PHB determina efectivamente, o tratamento de QoS que cada pacote recebe em cada encaminhamento. Esta norma descreve como marcar os pacotes IP para receberem PHBs apropriados, fazendo uso do DSCP (*Differentiated Services Code Point*), um campo do cabeçalho do pacote IP, constituído por 6 bits que permite 64 PHBs diferentes. No entanto, será necessário usar domínios *DiffServ*, onde estas políticas sejam implementadas consistentemente nos encaminhamentos pelos *routers*.

Dos PHBs existentes, o EF (*Expedited Forwarding*), está indicado para as necessidades específicas das aplicações usadas neste trabalho, isto porque, está orientado para uma classe de serviço que garanta entregas com requisitos próprios das aplicações *time-sensitive*, minimizando atrasos, perda de pacotes, etc.

¹⁴ <http://www.internet2.edu/qbone>

5.5.1. Integração de serviços em *multicast*

Num *framework* para serviços integrados [Wroclawski97, ShenkerI97, ShenkerII97, ShenkerIII97] a primeira função é fornecida pelos serviços de controlo da qualidade de serviço. A segunda função pode ser provida de várias maneiras, mas é frequentemente implementada por um protocolo de reserva de recursos como o RSVP [Braden97]. A reserva de recursos demonstrou ser uma solução válida para prover com QoS o tráfego *multicast* [Fourmaux98]. Em [Pullen99] pode encontrar-se um modelo de simulação com RSVP para tráfego *multicast*.

O RSVP foi projectado para ser usado com uma grande variedade de controlos de qualidade de serviços [Mankin97], e estas funções de controlo são desenvolvidas para serem usadas por uma grande variedade de mecanismos de configuração, associados às várias plataformas existentes (sistemas operativos e máquinas de arquitecturas específicas), devido a isto existe uma separação lógica entre as duas especificações. O RSVP adequa-se igualmente às comunicações *multicast*.

Quando diferentes membros do mesmo grupo *multicast* se filiam, nem todos dispõem das mesmas condições de acesso, pelo que nem todos poderão ser servidos com a mesma QoS [Wittman01]. Deste modo é incontornável o processo de *multicast* heterogéneo, onde cada receptor poderá requerer individualmente uma QoS determinada. Reforce-se que o RSVP está focalizado na sinalização, ou seja não providencia *data streams* heterogéneas. Se a solução passasse por aí então teríamos de recorrer a redes activas [Wittman98].

O esquema da **Figura 21** pretende representar as componentes do processo de reserva de recursos para um grupo *multicast* heterogéneo.

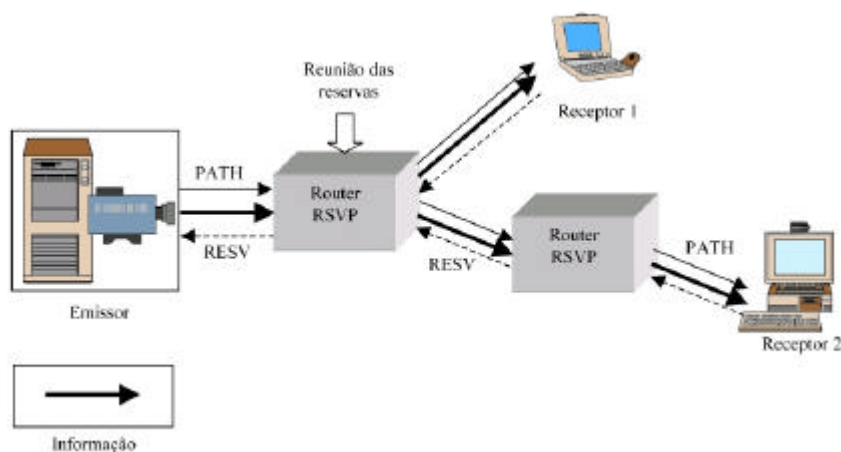


Figura 21 - Esquema para efectuar a reserva de recursos para o grupo *multicast* heterogéneo [Wang01].

A responsabilidade de modelar a quantidade de informação para cada receptor, mediante as suas limitações estará a cargo dos *routers* intermediários, assim será possível que um membro do grupo receba 5 *fps* enquanto que outro recebe da mesma sessão 25 *fps*.

5.5.1.1 Funcionamento do protocolo RSVP

O RSVP provê um procedimento, iniciado pelo receptor, para a reserva de recursos para um fluxo de dados com vários destinos (*multicast*) ou com apenas um destino (*unicast*). Este protocolo é usado para requerer uma qualidade de serviço específica, de uma rede de computadores, para o fluxo de dados de uma aplicação particular. O RSVP também é usado pelos encaminhadores para atender a pedidos de qualidade de serviço em todos os nodos que compõem caminho do fluxo, estabelecendo e mantendo este estado. Os pedidos RSVP vão desencadear uma negociação com cada nodo que faz parte do caminho dos dados, para que os recursos necessários sejam devidamente reservados.

A reserva de QoS é feita na ordem reversa [Liu98], ou seja, dos receptores ao transmissor. Essa solicitação de reserva repete-se até que chegue ao transmissor ou encontre um *router* com as mesmas necessidades. Tais reservas são implementadas através de dois tipos de mensagens: PATH e RESV (ver **Figura 21**).

- PATH: mensagens enviadas periodicamente pelo transmissor ao endereço *multicast*. Contém a especificação de fluxo (formato de dados, endereço fonte, porta fonte) e características de tráfego. Essa informação é utilizada pelos receptores para determinar o caminho de retorno ao transmissor e determinar quais os recursos que devem ser reservados. Os receptores devem pertencer ao grupo *multicast* a fim de receber mensagens PATH;
- RESV: mensagens geradas pelos receptores contendo parâmetros de reserva, como especificação de fluxo e de filtro. O filtro determina que pacotes no fluxo de dados devem ser usados no classificador de pacotes. A especificação de fluxo é usada no *escalador*, que procura satisfazer a necessidade do receptor.

O RSVP precisa da cooperação de outros protocolos para funcionar, um desses protocolos é o RTP (*Real Time Protocol*) (ver Anexo I), orientado ao transporte de tráfego multimídia.

Em resumo, o RSVP tem os seguintes atributos:

- faz reserva de recursos quer para , fluxos *unicast* e aplicações *multicast*;
- é simplex, i.e., só faz reservas para o fluxo de dados numa direção;
- o RSVP é orientado ao receptor, pois é este que inicia e mantém a reserva de recursos usadas por um determinado fluxo;
- mantém um *soft-state* (estado que carece de actualização frequente) nos *routers* e demais máquinas da rede, provendo suporte para mudanças dinâmicas nos membros de um grupo *multicast* e adaptação automática a mudanças na estrutura de *routing*;
- o RSVP não é um protocolo de encaminhamento, depende dos protocolos de *routing* actuais (ver Anexo I), suporta endereçamentos IPv4 e IPv6;
- o RSVP transporta e mantém parâmetros de controlo de tráfego e controlo de segurança na forma de "dados opacos" (inacessíveis);
- provê uma série de modelos ou estilos de reservas para atender a uma grande variedade de aplicações;
- provê um operação transparente através de *routers* que não o suportem;
- possui escalabilidade limitada, a solução pode passar por agregação de reservas (RFC 3175).

5.5.1.2 Estabelecimento de sessões *multicast* com RSVP

Ao nível de aplicação, uma sessão de cooperação está associada à utilização de um dado media. No caso dos dispositivos de informação utilizarem fluxos hierárquicos, à sessão de cooperação vão corresponder tantas sessões de transferência de informação quantos os sub-fluxos que constituem o fluxo hierárquico. Cada sessão de transferência é identificada por um endereço *multicast* X (identificador da sessão de cooperação) e por um porto de valor par Y.

Ao nível do sistema de comunicações, por cada sessão de transferência têm de ser estabelecidas:

- uma sessão RTP, constituída por uma parte de dados e uma de controlo. A sessão RTP é identificada pelo mesmo endereço *multicast(X) + porto(Y)* que lhe é passado pelo interface, sendo este endereço atribuído à transferência de dados. A transferência de informação de controlo é identificada pelo mesmo endereço *multicast* e pela porta com o número ímpar $Y+1$.
- duas sessões RSVP, uma referente à reserva dos recursos necessários à transmissão dos dados (*multicast(X) + porto(Y)*) e outra referente à reserva de recursos de controlo (*multicast(X) + porto(Y+1)*). A qualidade do serviço associada a uma sessão de dados é fornecida pela aplicação. No que se refere à sessão de controlo, esta apenas apresenta requisitos de qualidade referentes à largura de banda a disponibilizar para a transferência de dados de controlo.

No caso de uma videoconferência *multicast*, deverão ser criadas duas sessões RTP (áudio+vídeo), e para cada uma delas, mais duas sessões RSVP [Mendes98] (ver **Figura 22**).

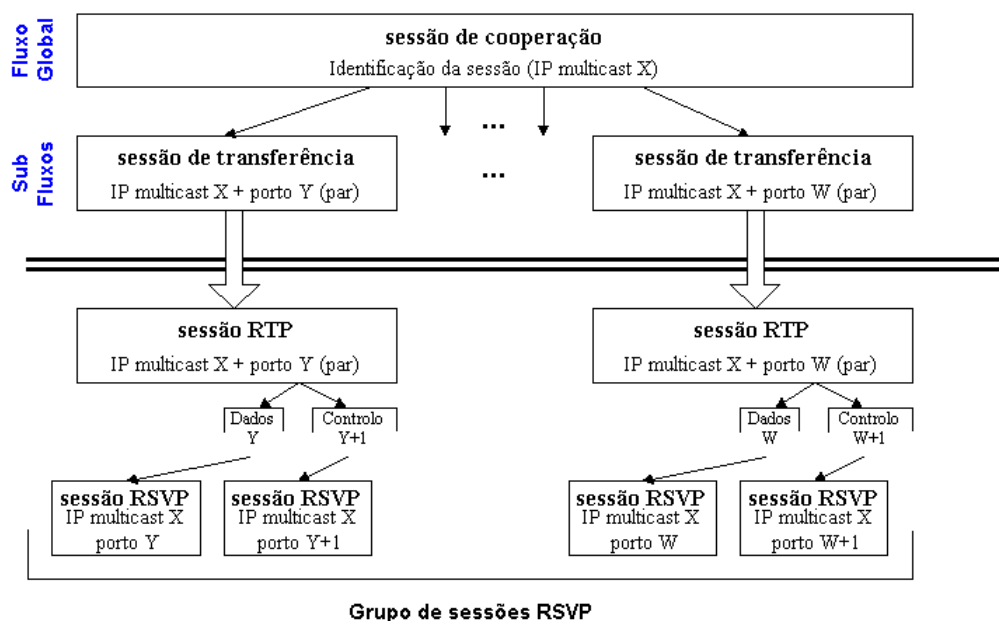


Figura 22 - Relação entre os conceitos de sessão [Mendes98].

Suportar um elevado número de reservas RSVP, ou sessões, pode significar um impacto negativo no desempenho dos *routers* [Ferguson98], portanto, a escalabilidade vê-se afectada. Consequentemente, na maioria das aplicações que requerem QoS, tem

de haver uma solução de compromisso na utilização dos recursos com mecanismos de obtenção de QoS.

São necessários mecanismos de segurança para verificar quem vai ter acesso ao serviço "de primeira classe", e no caso de conflito de requisitos, quem obterá os recursos? Como serão estabelecidas prioridades? Quem controlará este esquema? O RSVP ainda está em desenvolvimento para dar respostas a estas e outras questões.

Complementos e extensões do RSVP podem ser consultados nos RFCs: 2206, 2207, 2210, 2745 e 2746.

5.6. Relevância para o trabalho a desenvolver

A qualidade de serviço (QoS) em redes IP é um aspecto operacional fundamental para o desempenho fim-a-fim das novas aplicações (VoIP, multimédia, ...). Assim sendo, é importante o entendimento dos seus princípios, parâmetros, mecanismos, algoritmos e protocolos desenvolvidos e utilizados para a obtenção de uma QoS adequada.

O cenário de QoS fim-a-fim envolve uma multiplicidade de factores que concorrem para a degradação da QoS e pode ser consultado na pretérita **Figura 20**.

Todos os parâmetros de QoS, vistos neste capítulo, são variáveis que assumem, num ambiente heterogéneo como é a Internet, as mais diversas combinações, pelo que a solução para a aplicação a desenvolver deverá passar por dotá-la de alguma capacidade de adaptação aos parâmetros medidos, não só na rede como no sistema final. Assim, a QoS a providenciar ao sistema de EAD passará pelas duas perspectivas: a da rede e a da aplicação.

Na óptica da rede, monitorizam-se os seus parâmetros de QoS, que constituirão, juntamente com as propriedades do sistema cliente, os *inputs* para que o *middleware* regule proactivamente os recursos, por forma a assegurar maior escalabilidade e qualidade continuada para clientes que constituem um grupo *multicast* heterogéneo. Em situações de congestão da rede ou escassez de recursos, o *middleware* irá despoletar as aplicações num modo de "consumo mínimo", evitando dessa forma que o restabelecimento da QoS da rede seja hipotecado.

Na óptica da aplicação, o objectivo é preservar a qualidade de comunicação, assegurando os recursos para os parâmetros críticos. Por conseguinte, o *middleware*

actuará no sentido de modelar alguns parâmetros da aplicação que tolerem degradação de QoS, como sejam, o número de imagens por segundo, riqueza cromática, qualidade da imagem, largura de banda máxima utilizável, etc.

De entre as aplicações que integrarão o sistema de EAD, a adaptabilidade deverá aplicar-se somente às aplicações de voz e vídeo interativo, pois são aquelas que apresentam maior necessidade de QoS, e mais afectam os recursos disponíveis.

A forma de modelar os parâmetros de QoS das aplicações envolvidas neste trabalho resultou da conjugação dos resultados obtidos experimentalmente e das referências científicas nesta matéria. Por exemplo, os utilizadores de videoconferências, tipicamente requerem melhor qualidade de áudio que de vídeo [Bolot95]. É igualmente sabido que, o sucesso da comunicação numa videoconferência também pode depender de factores como, a cadência de *frames*, a qualidade da imagem, a resolução, o tamanho e a iluminação [Ahumada93, Miras02, Josifovski94].

Para finalizar o capítulo, apresenta-se o modelo de QoS, teoricamente ideal para uma aplicação com o cariz do ensino à distância, ressalvando que neste trabalho, por questões técnicas e de circunspeção será impossível implementar todo o modelo da **Figura 23**. Dos três pontos enunciados para arquitectura de QoS distribuída: mobilidade, adaptabilidade e heterogeneidade, será dada prioridade, fazendo juz aos objectivos, à adaptabilidade e heterogeneidade.

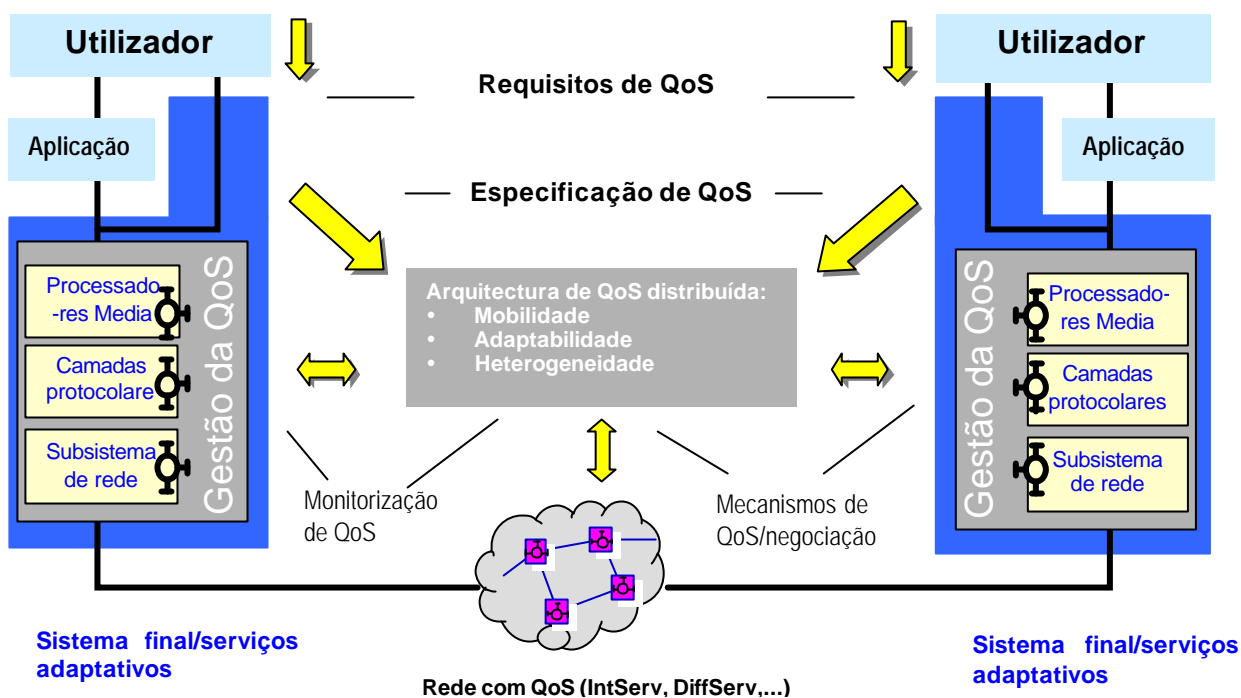


Figura 23 - Modelo de QoS para aplicações adaptativas multimédia *time-sensitive*.

Capítulo 6

Trabalho Desenvolvido

A inovação nas infra-estruturas telemáticas e a ubiquidade do serviço WWW, tornaram inevitável alargar a semântica das comunicações para níveis mais humanizados. Em particular, várias Instituições e Universidades de renome têm dedicado um enorme destaque e tributo ao desenvolvimento do ensino à distância com mediação pela Internet.

Com a proliferação das tecnologias multimédia, várias aplicações se candidataram a eleitas dos utilizadores de serviços de videoconferência pela Internet. O sistema IP/TV desenvolvido pela Cisco foi amplamente usado como plataforma eficiente no suporte a transmissões em formato MPEG-1¹⁵. Contudo, uma vez que assenta em tecnologia multimédia *Microsoft*, o sistema IP/TV não pode ser considerado

¹⁵ *Moving Pictures Expert Group*, norma 1 para armazenar e transmitir informação multimédia.

neutral quanto à arquitectura. Apesar de ser fornecida uma interface *Web*, faltam as aplicações complementares para o considerar uma ferramenta vocacionada para EAD.

As aplicações do Mbone, (*vic*) - *Video Conferencing System*, (*rat*) - *Robust Audio Tool*, *WhiteBoard* (*wb*) e *Network Text Editor* (*nte*), podem ser consideradas como o "canivete suíço" nestas lides, a gama completa de soluções desenvolvidas pela Universidade da Califórnia, Laboratórios Berkeley e UCL, já foi tratada no Capítulo 4, relativo às Aplicações *Multicast*. São factores abonatórios também, a multiplicidade de plataformas para as quais as aplicações estão produzidas bem como a enorme difusão nos meios académicos e organizacionais. Sendo ferramentas ainda em desenvolvimento, a estabilidade de algumas delas está aquém das exigências de criticidade de uma aplicação de medicina à distância, por exemplo, sobretudo o *vic*, seguramente pelo grande volume de tráfego que gera/processa, que pode atingir vários Mbps. Acresce também o facto de não possuírem um *plug-in* que permita correr em forma de *applet* um *player* para objectos RTP.

As limitações expostas podem ser suplantadas fazendo uso de ferramentas com a emergente tecnologia Java. A *Sun Microsystems* disponibilizou recentemente um *framework* para difusão selectiva de áudio e vídeo, encerrando todas as vantagens de portabilidade, próprias das aplicações Java, ideais para a heterogeneidade da Internet. Trata-se do JMF (*Java Media Framework*) instanciado no *JMStudio*, obviamente a "*webização*" das ferramentas também foi desenvolvida e permitirá coroar de simplicidade todo o processo de estabelecimento de uma sessão de *e-learning*.

Para este trabalho, o JMF funcionará como servidor de objectos RTP para os diferentes conteúdos das sessões *multicast*, capaz de lidar com objectos RTP/MPEG com eficácia e eficiência, assim, ficará a seu cargo a transmissão tutorial de áudio e vídeo.

O sistema está centrado no cliente, que eventualmente pode tornar-se emissor. A este, será proporcionada a alternativa de escolha das ferramentas *multicast* a utilizar, podendo optar por despoletar adaptativamente a dupla *vic+rat* para a sessão de videoconferência, ou bem fazer uso do *plug-in JMF*, este último, apenas para recepção, não suportando adaptabilidade na sua pré-parametrização.

Geralmente os serviços de EAD podem ser entregues de forma tríplice:

- síncrona (tempo-real);
- assíncrona (a pedido);
- híbrida.

Os sistemas de ensino à distância síncronos providenciam uma grande aproximação às condições reais de uma sala de aula, os conteúdos multimédia são apresentados em directo e ao vivo.

Os sistemas assíncronos oferecem acesso a arquivos didáctico-pedagógicos digitais recorrendo a tecnologias *web* e/ou *streaming*.

Sistemas híbridos permitem o complemento funcional para os serviços supramencionados. Estudos recentes, demonstram que os estudantes *on-line* preferem os serviços híbridos, com predominância para a componente assíncrona [telelearn].

No sistema de EAD desenvolvido poderá aceder-se a estes três tipos de serviços.

6.1. Introdução ao sistema adaptativo aplicado ao Ensino à Distância

O *framework* desenvolvido tem por finalidade disponibilizar um sistema adaptativo, onde diferentes elementos se configuram para fornecer um serviço integrado de transmissão de conhecimento. O ensino à distância é apenas a instanciação que serviu de base a este trabalho, no entanto o sistema pode adaptar-se com relativa facilidade para outros propósitos afins, desde o trabalho corporativo aos *helpdesks*. No mundo empresarial, como na educação, fidelizar a clientela passa por um estreito relacionamento com a mesma, no sentido de corresponder aos seus anseios, e esta correspondência passa necessariamente pela melhor forma de comunicação possível. Pretende-se com o sistema desenvolvido contribuir nesse sentido.

6.1.1. Funcionalidades

Sinteticamente, o sistema visa aproximar a mais avançada tecnologia de *groupware*, numa interface *web* de usabilidade profusamente desenvolvida, que permita aos agentes educativos usar as ferramentas *multicast* de forma transparente. Acrescidamente, é salvaguardada a qualidade de serviço (QoS) em serviços assíncros e síncronos por intermédio de automatismos de adaptabilidade que supervisionam o bom funcionamento do sistema. Conceptualmente, a gestão da QoS das aplicações poderá ser exercida como ilustrado na **Figura 24**.

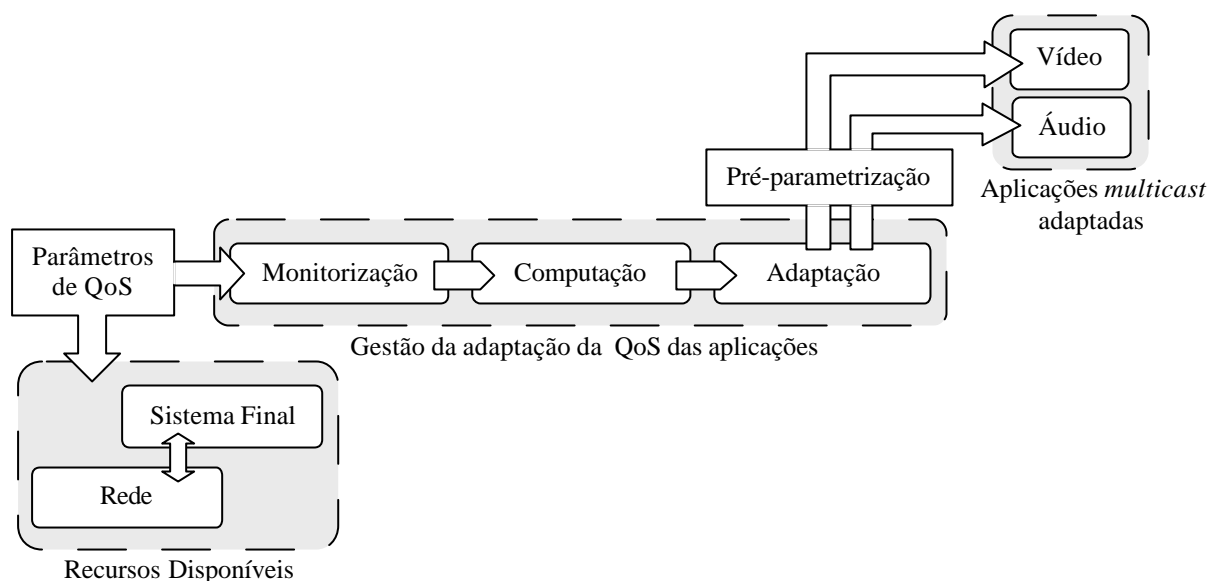


Figura 24 - Controlo da QoS das aplicações *multicast*.

O sistema desenvolvido está preparado para fornecer vários níveis de serviço:

- faculdade virtual, baseada numa interface *Web*, com usabilidade profundamente estudada e apurada, de modo a servir de ponto de partida e chegada aos conteúdos disponíveis, centralizando as demandas e organizando as respostas, conduzindo alunos e professores aos serviços disponibilizados de forma eficaz e eficiente;
- registo, autenticação e manutenção de estado dos agentes educativos, alunos e tutores mormente;
- gestão/manutenção/calendarização das sessões *multicast* originadas pelo sistema de educação à distância;
- acesso a material pedagógico assincronamente, como *multicast* de gravações de aulas, apresentações diferidas, documentação multimédia, etc.;
- *multicast* em tempo real de texto, áudio e videoconferências, para grupos de utilizadores dispersos na Internet, com acesso adaptativo para qualidade de serviço diferenciada nos clientes receptores e clientes receptores/emissores;
- disponibilização de ferramentas de desenvolvimento de relações de comunidade e espaços de trabalho partilhado como sendo, *whiteboard multicast*, fóruns e *chats* (textuais e gráficos) ou apresentação de *slides*.

6.1.2. Descrição da plataforma experimental

Trata-se de um sistema pioneiro na Instituição, integrado na rede de comunicações do IPB - Instituto Politécnico de Bragança, onde se realizaram a maioria dos testes que servem de base a este trabalho, espera-se que amadureça e se torne numa componente fundamental no futuro do ensino desta escola.

A apregoada heterogeneidade da Internet começa logo no sistema apresentado, de facto, o servidor Áudio/Vídeo RTP assenta num sistema *Windows 2000 Server*, o seu *hardware* define-se por um processador Intel Pentium III, 700 MHz e 256 MB de memória RAM. Incorpora um dispositivo de captura de vídeo especial para videoconferências, mais precisamente uma câmara de vídeo SONY EVID-31 com possibilidade de orientação controlada pelos participantes remotos.

O servidor de objectos *Web* (Apache) foi instalado num sistema Linux, versão Red Hat 7.0, com mysql/PHP 4. Este equipamento inclui um processador Intel Pentium II 333 MHz, e memória RAM com capacidade de 128 MB.

Os componentes de interligação da rede, onde se destaca o *router* Cisco 4000, o cerne das comunicações *multicast* do IPB, estão aptos para *multicast*, o *routing* para o Mbone encontra-se configurado pelo protocolo PIM-SM (ver Anexo I).

Na LAN do campus do IPB, plataforma de testes primária para este estudo, os actuais 4 Mbps disponíveis para a comunidade académica se ligar ao exterior (no nosso caso ao Mbone) são sempre escassos, principalmente quando o tráfego provém de aplicações de conferência multimédia. Em média, 50% dessa capacidade está permanentemente ocupada (ver **Figura 25**).

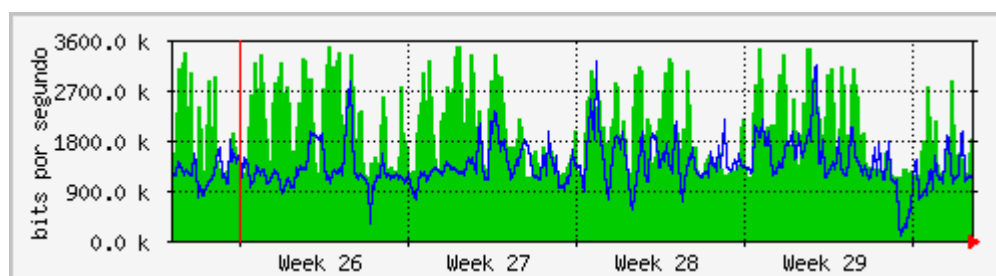


Figura 25 - Gráfico mensal (julho/02) do tráfego entre o IPB e a Internet (*router* da RCTS).

A **Figura 26** dá uma perspectiva esquemática do sistema na sua globalidade, onde são visíveis os componentes que constituem o lado servidor, a topologia de rede de alto nível bem como a visão das entidades clientes.

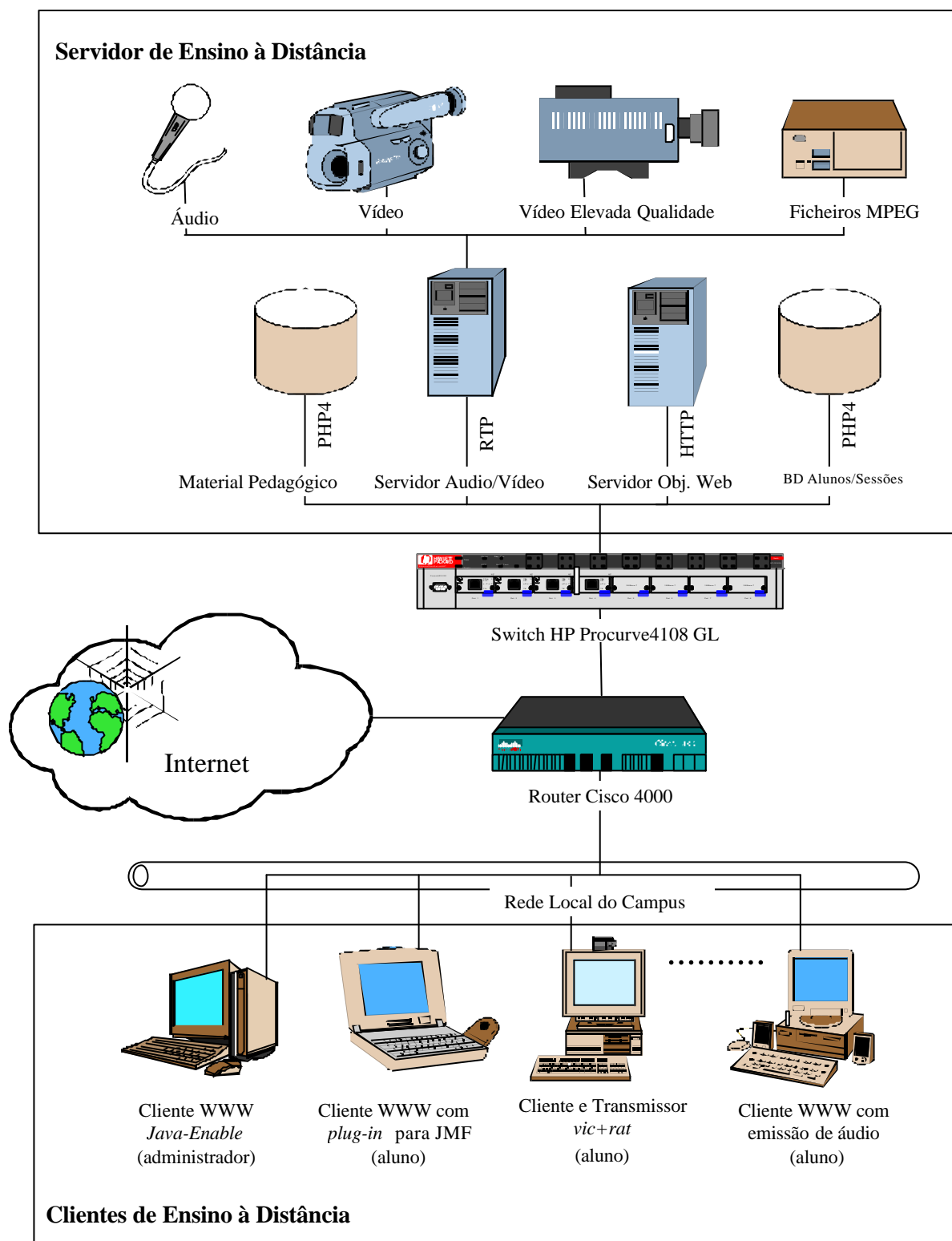


Figura 26 - Sistema desenvolvido para suporte de ensino à distância.

Ao nível da sub-rede onde a plataforma foi instalada (LAN do CIESA¹⁶), as ligações do *backbone* interno (entre *switch* e *router* de saída para a Internet), fazem-se a 1 Gbps. Do *switch* para os *hosts*, os troços mais lentos operam a 100 Mbps, limitados pelas placas de rede das máquinas e por alguns nós mais lentos (*hubs*).

O tráfego desta sub-rede caracteriza-se por um volume reduzido, resultante da navegação dos utilizadores do CIESA na Internet ou pontualmente da transferência de ficheiros de maior dimensão. Refira-se que em alturas de testes, 95% do tráfego detectado na rede é UDP/*multicast*.

Os sistemas clientes que serviram para testar o sistema são três, todos baseados em SO *Windows 2000 Professional*. Eis a descrição sintética do seu hardware:

- processador Pentium II, 333MHz, 196 MB RAM, Creative Web Cam e som;
- processador Pentium II, 700MHz, 256 MB RAM., Creative Web Cam e som;
- processador Pentium IV, 1700MHz, 256 MB RAM, Creative Web Cam e som.

Idealmente, deveria haver mais clientes com possibilidade de emissão (áudio/vídeo) mas, por falta de câmaras de vídeo não foi possível alargar a base de teste.

6.2. Arquitectura do sistema adaptativo de comunicação *multicast*

As aplicações de ensino à distância que assentam em videoconferência, ou aplicações de vídeo assíncrono, requerem ao serviço de rede *multicast*, condições que assegurem que os atrasos e as perdas experimentadas pelos pacotes de áudio/vídeo se situam dentro dos limites tolerados, por forma a manter uma qualidade aceitável na informação que é prestada ao utilizador.

Uma das soluções para lidar com este *status quo*, prende-se com a introdução de adaptabilidade e comportamento proactivo das aplicações [Li00, Ramanujan98].

A adaptabilidade da aplicação será exercida de forma independente da aplicação, introduzindo uma camada de *middleware*. Será necessário que os vários módulos (*applets* e *javascript*) que a constituem, em primeira instância, se possam inteirar das condições de QoS existentes em tempo de transmissão, bem como dos recursos à disposição (sobretudo a capacidade de processamento e recursos de memória

¹⁶ Centro Informático da Escola Superior Agrária de Bragança

disponíveis) nos sistemas clientes, e actue no sentido de regular de forma pró-activa as parametrizações das aplicações *multicast* a executar.

Na **Figura 27**, podem observar-se os elementos constituintes da arquitectura do sistema desenvolvido para comunicação em grupo *multicast*, com adaptação da QoS por *middleware*.

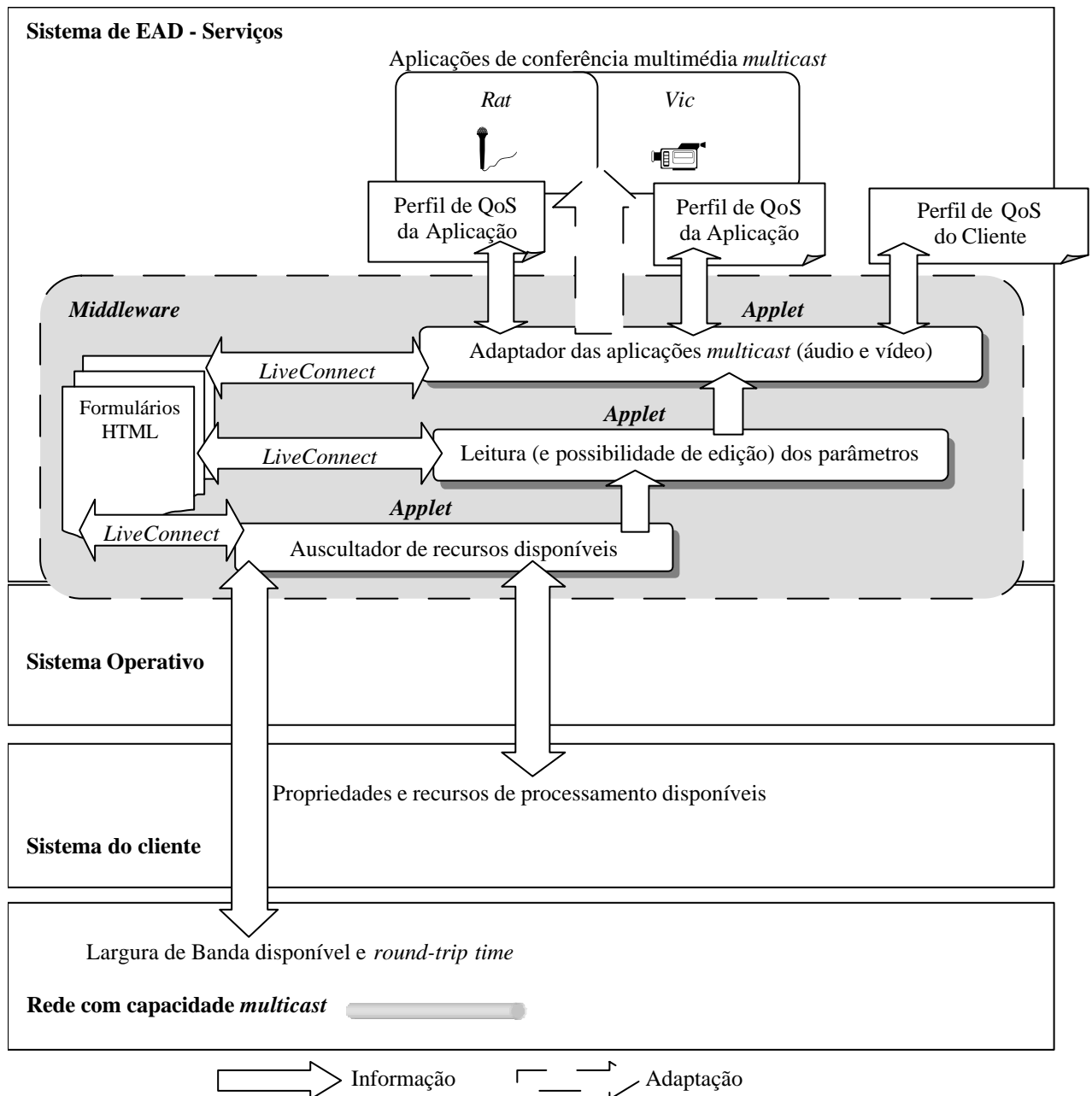


Figura 27 - Arquitectura do sistema adaptativo de comunicação *multicast* desenvolvido.

Como parâmetros que serão alvo de tratamento para fornecer um comportamento adaptativo das aplicações, destacam-se os seguintes:

- no caso do sistema cliente pretender tornar-se activo na emissão de áudio/vídeo, adaptar a qualidade da imagem/som que é transmitido em função das condições verificadas ao nível de comunicações/processamento/memória;
- no caso do vídeo, adaptar a qualidade da imagem e a taxa de imagens por segundo aos recursos de largura de banda disponíveis;
- em cenário de escassez de recursos optar pelos formatos de codificação mais compactos na transmissão de áudio e vídeo bem como limitar o consumo de largura de banda;
- em suma, fornecer ao utilizador de forma transparente, o melhor serviço *multicast* que o seu sistema/ligação podem alcançar sem comprometer a rede e a comunicação com o grupo.

A camada de *middleware* é o cerne da arquitectura, praticamente todos os fluxos de controlo do sistema são originados nesse módulo. Antes de se processar a adaptabilidade é necessário recolher dados, auscultando o sistema cliente. Compete à camada de *middleware*, avaliar as condições de interligação e processamento.

Os valores aferidos, em parâmetros de QoS da rede e do sistema considerados fundamentais, como sejam: a largura de banda disponível, o *round-trip time (rtt)*, a carga do processador e a memória disponível; são os *inputs* para o módulo que calcula o modo de adaptação que melhor servirá o sistema de EAD e o grupo *multicast*. Todos os mecanismos desenvolvidos para garantir a adaptabilidade serão explanados na secção 6.5.

A linguagem Java, pela sua perfeita adequabilidade às condições heterogéneas da Internet, foi eleita como ferramenta de desenvolvimento das aplicações necessárias. Deste modo, as *classes* (em forma de *applets*) poderão executar-se sem impedimentos de incompatibilidade em qualquer sistema final. O uso da tecnologia Java garante algumas propriedades consideradas fundamentais para o sucesso do sistema:

- independente da plataforma → adaptação à heterogeneidade da Internet;
- extensibilidade, garantindo actualizações em tempo de execução e de forma generalizada para todos os utilizadores;
- mobilidade (mais relevante nos desenvolvimentos futuros).

6.3. Descrição da interface *Web* e dos serviços disponibilizados

Por forma a efectivar a ponte entre os utentes e os serviços disponibilizados, procedeu-se à criação de uma interface WWW, que faz uso das mais variadas formas de exposição/interacção de conteúdos, a saber, HTML, *Javascript*, *applets* Java, bases de dados geridas por PHP, etc.

Serve esta interface para que o utilizador se sinta virtualmente, num espaço integrado de formação, desde a sua inscrição, aprendizagem, experiências extra-curriculares contextualizadas e avaliação.

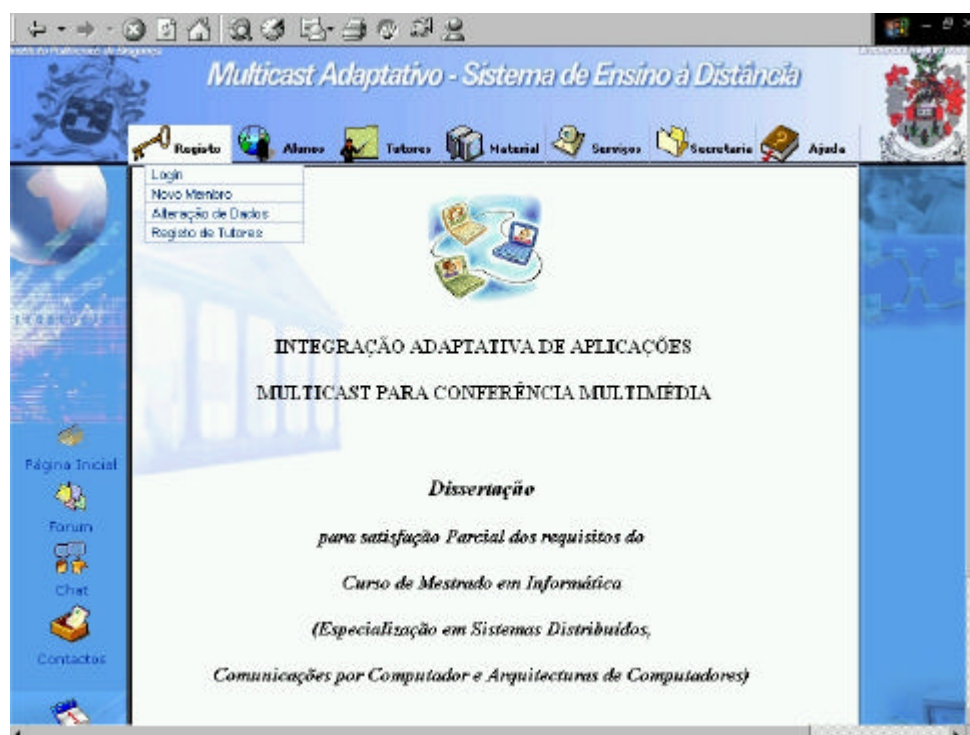


Figura 28 - Aspecto geral da interface *Web*.

O objectivo magno desta componente, é facilitar o uso da plataforma, e por inerência, contribuir para o desenvolvimento das aplicações *multicast* ao nível da usabilidade, tornando transparente para o utilizador a configuração e o lançamento das aplicações envolvidas. Encontra-se acessível no URL <http://www.esa.ipb.pt/multicast>.

Parte-se do princípio que o acto de navegar pela Internet se banalizará nos próximos anos e que qualquer indivíduo poderá fazê-lo sem exigência prévia de formação específica, pelo que o acesso aos serviços será o mais democrático possível, e tornará a tarefa de assistir a sessões *multicast* via Internet tão simples como ligar a televisão e seleccionar o programa desejado.

O utilizador poderá escolher a forma de aceder à informação em qualquer momento, optando por formas assíncronas (apresentações tipo transparências, forum de discussão, etc.) ou por forma de média em tempo real síncronas (momento áudio e videoconferência). Todas as variantes deverão estar acessíveis num mínimo de passos possível, evitando ao utilizador difíceis rememorações de actividades acessórias.

Basicamente, todas as funcionalidades têm um acesso directo, accionando apenas uma hiperligação, sendo que para aceder a algumas delas se deverá anteriormente abrir o menu correspondente, deslizando o rato pela barra de menus, de tipo *pull-down*, que encima o portal.

A área central está reservada à apresentação dos conteúdos e será a única que deverá sofrer mutações. O entorno, excepto a parte inferior, ficará como ambiente estável onde estão acessíveis os *links* para os serviços disponibilizados. Tal configuração, permite em qualquer instante, inverter totalmente o rumo da navegação sem necessidade de voltar a qualquer outra localização prévia, não interferindo com a liberdade e espontaneidade do utilizador.

Ao nível técnico, esta disposição funcional está baseada em novos paradigmas das versões mais recentes do HTML, concretamente o IFRAME, que permite situar uma janela de *browser* num qualquer lugar da nossa área de exposição e fazer com que essa janela se comporte como uma janela independente onde poderão ser carregadas outras páginas WWW.

Na já referida barra de menus poderemos encontrar, estruturadas por cronologia do processo educativo, todas as opções conducentes às boas práticas pedagógicas. Obviamente, esta versão não pretende ser meticulosa em relação a qualquer vicissitude do processo, mas sim ser palco do trabalho realizado para a dissertação de mestrado, congregando no geral todas as entidades e agentes envolvidos no processo educativo à distância e aproximando as tecnologias *multicast* ao utilizador nesta instanciação que é o *e-learning*.

6.3.1. Funcionalidades disponibilizadas

Segue-se a explanação do acervo de funcionalidades que estão disponíveis aos agentes envolvidos no processo educativo baseado em tecnologias de informação e multimédia.

6.3.1.1 Registo

Destituir um processo de aprendizagem de humanização, mesmo sendo esse processo não presencial, não se afigura como vantajoso, existem mais vantagens na constituição de uma comunidade que partilha objectivos, plataformas, conhecimentos, etc. O anonimato acaba por ser uma porta de fuga mais do que uma porta de entrada.

O processo de registo é facultativo, no entanto alguns serviços estão vedados aos utilizadores que não estão registados. Cada utilizador terá uma identificação perante o sistema que lhe permitirá aceder aos serviços para os quais está autorizado.

Para se registar basta aceder ao menu "Registo" e escolher a opção "Novo membro", seguidamente preencher um pequeno formulário (**Figura 29**) com os seus dados e clicar em "Registar" para que a base de dados os recolha.

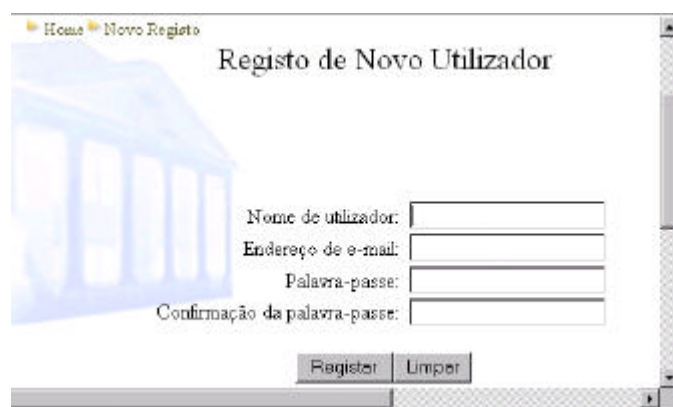
A imagem mostra uma janela de navegador com o título "Registo de Novo Utilizador". No canto superior esquerdo, há links para "Home" e "Novo Registo". O formulário contém quatro campos de texto rotulados: "Nome de utilizador:", "Endereço de e-mail:", "Palavra-passe:" e "Confirmação da palavra-passe:". Abaixo dos campos, há dois botões: "Registar" e "Limpar".

Figura 29 - Registo de novo membro da comunidade.

6.3.1.2 Serviços para alunos

Os alunos são o cerne da questão, é em torno deles que evolui todo o processo.

Depois de efectivar a sua inscrição como membro da comunidade, torna-se um potencial aluno, pois em qualquer momento, mediante a sua iniciativa, poderá integrar um grupo académico ou mais, participando em todos os eventos que a academia virtual propicie com afectação à área de actividade lectiva em que se matriculou.

Consultar o calendário das sessões *multicast on-line* agendadas, associar-se a essas mesmas sessões como participante passivo ou activo, inteirar-se dos eventos e

aquilatar os seus progressos (a avaliação não foi implementada), são algumas das opções que pontificam nesta secção.

Optou-se por criar uma base de dados para gerir a calendarização das sessões. Como é sabido, no Mbone, as sessões são publicitadas via *sdr* (*session directory*) que mantém actualizada a listagem das sessões presentes ou futuras. Esta aplicação funciona com base em informação, com um formato específico definido pelo protocolo SDP (*Session Description Protocol*) descrito em [Handley98], que foi rigorosamente respeitado neste trabalho, sendo que quando é criada uma sessão todas as informações relevantes para o SDP são guardadas.

Sessão	Descrição	Áudio	Video	Wboard	Texto	Início Sessão	Fim Sessão	Moderador
teste	primeiro teste	✓	✗	✗	✗	2001-07-02 18:23:46	2001-08-02 18:00:00	sergiold@ipb.pt
Aula n°1	Sessão sobre Multicast IP	✓	✓	✗	✗	2001-08-22 14:47:11	2001-09-02 14:47:11	sergiold@ipb.pt

Figura 30 - Serviços para alunos.

6.3.1.3 Serviços para tutores

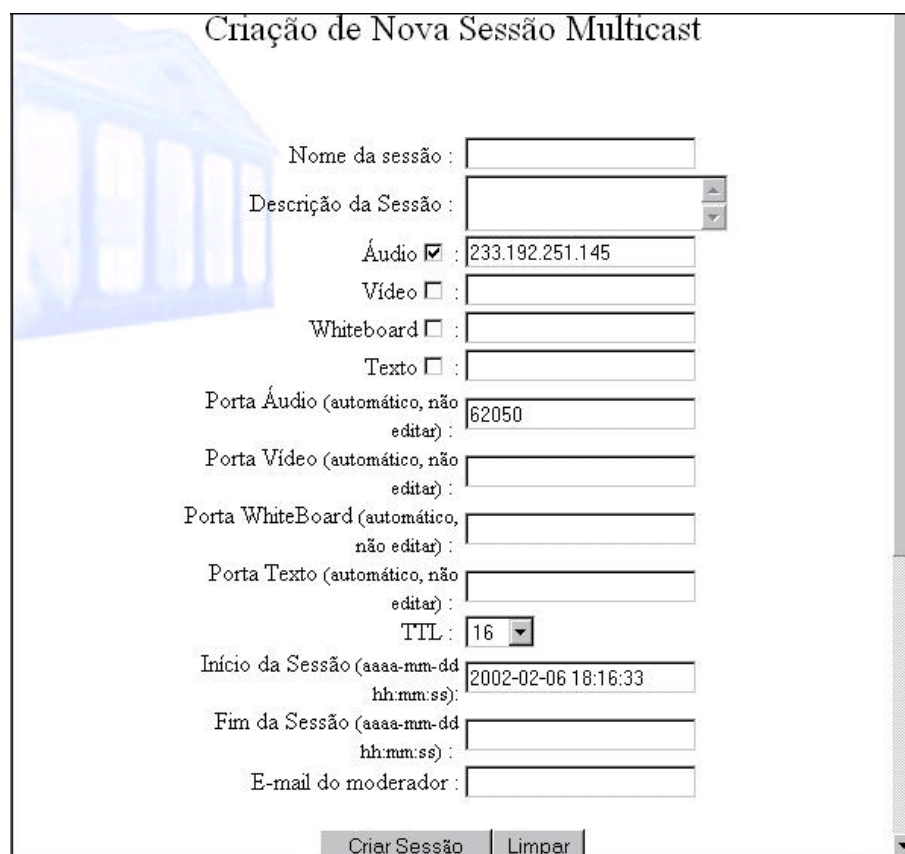
Este ensino estará melhor enquadrado na óptica tutorial, o indivíduo procura a sua própria formação e carece de um tutor que o ajude na sua tarefa, essa ajuda deverá ir no sentido de orientar o aluno sem recurso a qualquer imposição ou sub-alternização.

No modelo em apresentação, cabe ao tutor, no tocante à produção de conteúdos síncronos, decidir os conteúdos que melhor servirão os interesses do grupo académico, bem como sequenciá-los em sessões.

Sendo as sessões de conferência *multicast* a ferramenta mais poderosa deste tipo de ensino, pois coloca todos os tipos de média em convergência para a finalidade de

aprender, o tutor deverá saber gerir esses meios, convocando os argumentos adequados que podem ir do vídeo ao som, passando pelo texto, imagens e grafismos.

Criar uma sessão exige tão-somente o preenchimento do formulário *on-line* retratado na **Figura 31**.



O formulário, intitulado "Criação de Nova Sessão Multicast", contém os seguintes campos e controles:

- Nome da sessão : [campo de texto]
- Descrição da Sessão : [campo de texto]
- Áudio : 233.192.251.145
- Vídeo : [campo de texto]
- Whiteboard : [campo de texto]
- Texto : [campo de texto]
- Porta Áudio (automático, não editar) : 62050
- Porta Vídeo (automático, não editar) : [campo de texto]
- Porta WhiteBoard (automático, não editar) : [campo de texto]
- Porta Texto (automático, não editar) : [campo de texto]
- TTL : 16
- Início da Sessão (aaaa-mm-dd hh:mm:ss): 2002-02-06 18:16:33
- Fim da Sessão (aaaa-mm-dd hh:mm:ss) : [campo de texto]
- E-mail do moderador : [campo de texto]

Botões: Criar Sessão | Limpar

Figura 31 - Formulário para criação de sessões *multicast* (da responsabilidade do tutor).

O preenchimento dos pormenores técnicos é assegurado automaticamente pela aplicação, exigindo apenas ao tutor que assinala, nas caixas de verificação correspondentes, os media que pretende usar, encarregando-se a aplicação de conformar os parâmetros relativos ao endereço *multicast* e porta para as comunicações *multicast*. Este automatismo é logrado por um pequeno trecho de programação em *Javascript*, embutido na página, que "injecta" nas caixas de texto correspondentes os dados técnicos relevantes à ulterior realização da sessão. O mesmo é aplicável ao campo "data e hora" de início da sessão, que é automaticamente preenchido pela aplicação, mas de forma editável obviamente.

A moderação das sessões, visando a produtividade, será da responsabilidade do tutor, que deverá garantir que o grupo de trabalho se focalize no essencial e que não gracem os despropósitos.

Resulta de todo este processo de definição de características de uma nova sessão, a inserção final de um novo registo na base de dados, que constitui o repositório de dados do sistema de informação associado ao processo de EAD. Imediatamente, os alunos poderão consultar as sessões calendarizadas e associar-se a elas.

Aquando do momento da realização da sessão, o tutor deverá deslocar-se à sala de emissão, onde contará com o servidor *multicast*, ferramentas *multicast*, câmaras de captação de vídeo, microfones, etc.

6.3.1.4 Material

Aulas, conferências, palestras, vídeos didácticos, e outras fontes de conhecimento não perdem a sua eficácia se forem emitidas em diferido, podem ser organizadas de forma a responderem às necessidades dos alunos, e reproduzidas a pedido.

Qualquer material didáctico em formato digital pode ter cabimento nesta secção, alargando a base de conhecimento acessível aos alunos. Como já foi referido neste trabalho, demonstram estudos actuais que os conteúdos assíncronos são, pelo menos por agora, os predilectos dos e-alunos. As faltas às aulas, e as suas consequências nefastas em termos de estabilidade no processo, seriam potencialmente anuladas, pois os alunos poderiam sempre aceder ao material registado. A participação/interacção e discussão ao vivo estaria obviamente impossibilitada, mas não seriam privados da recolha dos ensinamentos proferidos na sessão pretérita.

O formato que se apresenta a seguir é o vídeo a pedido, onde o aluno poderá (re)visualizar as sessões em arquivo, controlando todo o processo e acedendo apenas aos excertos que lhe interessem.

Para reprodução utiliza-se um *plug-in* que revela bom desempenho e, por utilizar o JMF não exige mais aplicações ao e-aluno. O material é previamente importado do servidor e posteriormente reproduzido, desta forma a estabilidade está assegurada. Segue-se a exemplificação, ilustrada na **Figura 32**.



Figura 32 - Exemplo de acesso a vídeo a pedido.

6.3.1.5 Serviços de conferência

Nesta secção, os agentes educativos, poderão aceder aos recursos mais avançados da academia virtual, as sessões *multicast*. A adesão às conferências pode ocorrer em três perspectivas. Na perspectiva de participante apenas passivo, pontualmente activo ou inicialmente activo.

Na modalidade de participante apenas passivo, poderá ainda optar por assistir às sessões áudio+vídeo ou apenas áudio, num *player* do JMF, que se encarrega de receber e reproduzir o tráfego RTP áudio/vídeo, mas sem qualquer possibilidade de transmitir.

Participar passivamente por *plug-in*, reduz drasticamente a carga computacional necessária, isto porque, o sistema de EAD recorre a um *applet* em *Java* (*SimplePlayerApplet.class*), capaz de reproduzir tráfego RTP, que é importado do servidor, evitando assim a necessidade de abrir aplicações locais. Toda a recepção tem lugar na própria janela do *browser* e o utilizador terá um esforço mínimo em atingir esse estado de e-aluno. A limitação reside no facto do aluno apenas poder visualizar o tutor.

A **Figura 33** ilustra o resultado da transmissão RTP/vídeo que é directamente visualizável no *site*, via *applet*. O *applet* *SimplePlayerApplet*, utiliza o *framework* JMF (descrito na secção 4.3.3), requerendo portanto que o *package* (de *download* gratuito em <http://java.sun.com/products/java-media/jmf/2.1.1/download.html>) se encontre instalado no computador do e-aluno. O código *Java* correspondente ao *applet* *SimplePlayerApplet* pode ser consultado no Anexo II.

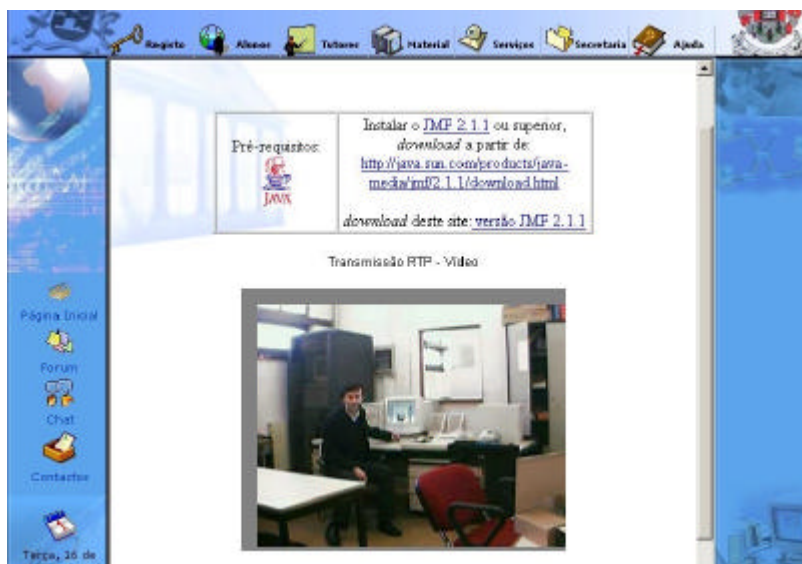


Figura 33 - Recepção de RTP/vídeo e RTP/áudio directamente no *browser*.

No caso da participação activa, o e-aluno deverá possuir as aplicações *multicast*, instaladas e inscritas na variável de ambiente PATH na sua máquina, estas serão invocadas e parametrizadas pela sistema de EAD aqui em exposição, tudo de forma transparente. Este é um processo mais exigente que a perspectiva passiva, pois necessita de possuir o hardware de emissão devidamente configurado e funcional sempre que queira emitir, algo que pode não ser trivial para a maioria dos participantes.

Se a opção for participar na conferência de forma pontualmente activa, ou seja, adere-se ao grupo com o intuito inicial de receber apenas os media, mas pretende-se ter a possibilidade de intervir num dado momento, então deverão utilizar-se as opções de conferência (Figura 34), nas quais são despoletadas as aplicações *multicast* (*rat*, *vic*, *wb* ou *nte*). Nesta modalidade as aplicações *multicast* são executadas simplesmente com os parâmetros básicos SDP, ou seja com as descrições da sessão e sem qualquer adaptação.

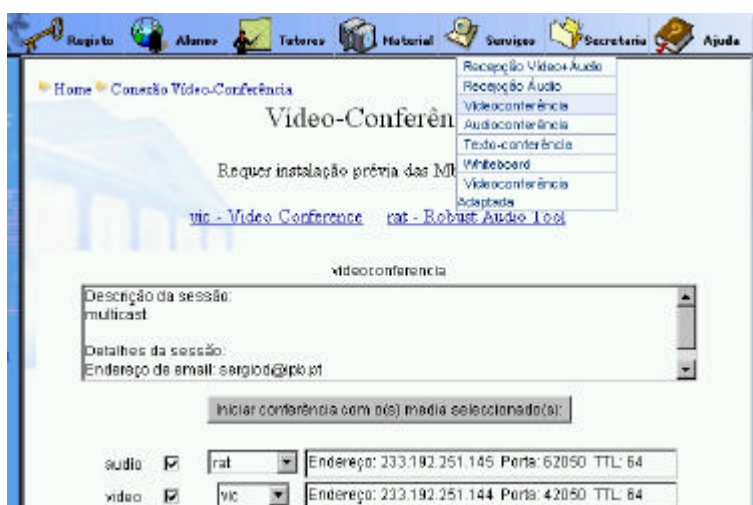


Figura 34 - Aceder a sessões *multicast* na perspectiva de participante pontualmente activo.

Por último, a terceira modalidade, permite aderir adaptativamente a uma videoconferência *multicast* de forma activa. Entra-se na sessão com as aplicações configuradas para emitir de forma adaptada aos recursos de interligação e processamento disponíveis. Módulos em *java* e *javascript*, veiculados com as páginas HTML, encarregam-se de auscultar o meio e o sistema cliente por forma a obterem a informação necessária para decidir a adaptação a aplicar. As aplicações que sofrem adaptação são as de áudio e vídeo. Mais adiante, a secção 6.5, trata este tema com o detalhe e rigor devidos. Para usar este modo, segue-se, na interface, o percurso que está retratado na **Figura 35**.

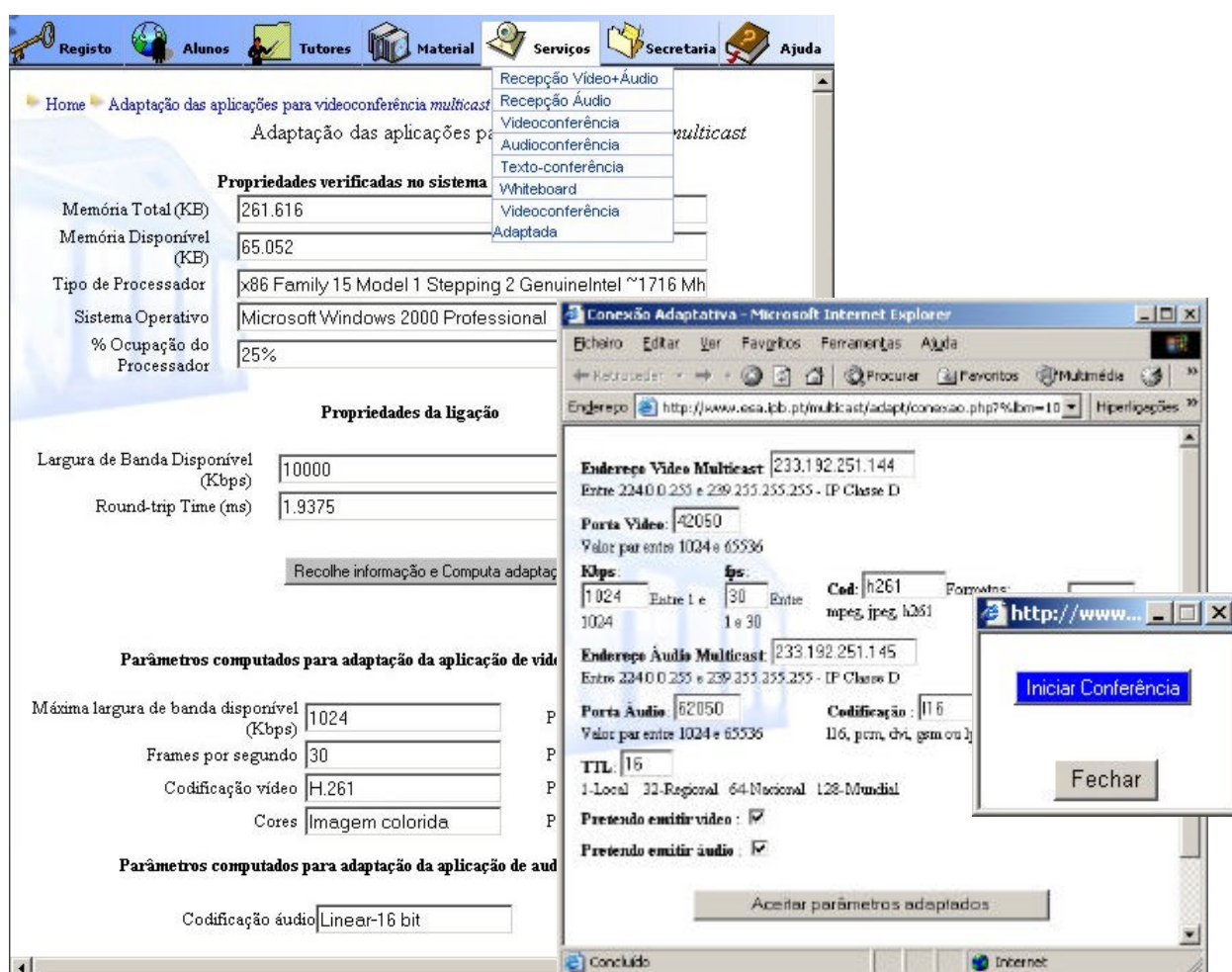


Figura 35 - Sequência de passos para participar numa sessão *multicast* com interacção e adaptação.

Como já foi anteriormente referido, o *middleware* que gere a adaptabilidade é baseado em módulos *Java*. Alguns deles, necessitam da autorização do sistema cliente (utilizador), para desempenharem funções que requerem privilégios que estão

normalmente vedados por questões de segurança, cabe aqui o exemplo do lançamento das aplicações *multicast* no sistema cliente. Pela razão apontada, o utilizador será "interpelado" por uma janela de autorização à qual deverá responder afirmativamente, ratificando a confiança no sistema servidor de EAD.

Depois de seleccionar e despoletar as aplicações *multicast* que pretende utilizar, o e-aluno passará a poder visualizar e/ou ouvir a conferência multimédia, uma panorâmica da mesma pode ser observada na **Figura 36**.



Figura 36 - Vídeo-conferência *multicast* multiparticipada.

6.3.1.6 Secretaria

Os serviços de secretaria não se inserem nas valências a que este trabalho tem de responder, no entanto os serviços administrativos são fundamentais, mas enquanto projecto experimental pode-se prescindir da sua contribuição, já que nesta fase do projecto não teremos alunos matriculados nem certidões a passar.

6.3.1.7 Ajuda

É um sistema de ajuda *on-line*, que fornece informações úteis a quem tem os primeiros contactos com o sistema de EAD. Futuramente as próprias sessões *multicast* poderiam servir um sistema de *helpdesk* mais personalizado, permitindo ao utilizador conferenciar com alguém que tivesse as funções de ajudar os e-alunos a explorar o sistema de EAD.

6.3.1.8 Fórum

Trata-se de um fórum de discussão baseado em *posts* hierarquizados cronologicamente, os participantes iniciam uma cadeia de discussão ou respondem a um *post* ou *sub-post* (ver **Figura 37**). É o modelo convencional, com maior difusão na Internet. Agrega facilidades de pesquisa e apresentação da informação.

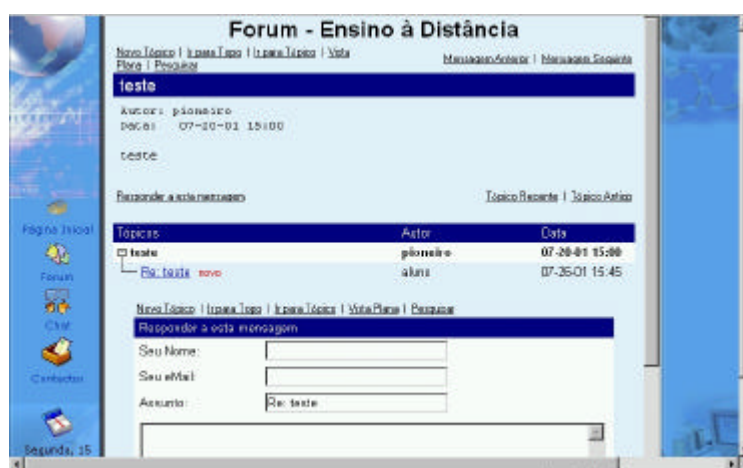


Figura 37 - Aspecto do fórum de discussão.

6.3.1.9 Chat multimédia *multicast*

Foi instalada uma aplicação em *Java*, servidora de *chat* multimédia (**Figura 38**), onde os participantes poderão trocar impressões, via texto ou mesmo grafismos através da interface *Web*, sem necessidade de qualquer configuração, basta identificar-se perante o grupo e correr o *applet* cliente.

Funciona baseado em tecnologia *multicast*, onde os participantes integram um grupo que partilha um endereço IP classe D e uma porta de comunicações.



Figura 38 - Aspecto do chat multimédia *multicast*.

6.4. Organização das sessões *multicast* e dos utilizadores de serviços

Uma outra função a desempenhar pelo sistema de ensino à distância, prende-se com a criação e publicitação de sessões *multicast* para servir os interesses pedagógicos dos grupos.

Obviamente a criação de sessões relativas a palestras, aulas, ou outros eventos pedagógicos, está ao cuidado dos tutores (dentro das suas disciplinas) ou do administrador do sistema. Aos alunos caberá o papel de aderir às sessões calendarizadas, podendo integrar o grupo *multicast* como membros passivos ou activos. Ainda assim poderão aceder a uma panóplia de serviços que estão ao alcance do seu nível de segurança. Para estabelecer esta hierarquia, torna-se necessário criar uma base de dados para diferenciar o tipo de acesso dos diferentes utilizadores.

Para implementar as resoluções preconizadas, optou-se por criar uma base de dados para cada entidade:

- **phpSecurePages** - a base de dados de diferenciação de acesso a utilizadores;
- **phpGestSessoes** - a base de dados de gestão e calendarização de sessões *multicast*.

Ambas bases de dados, residentes no servidor de objectos *web*, foram criados em *mysql* para Linux/Apache e incluem as seguintes tabelas.

PhpSP_users	phpGS_sesoes
iduser (Primary key)	idsessao (Primary key)
utilizador (Unique)	nomesessao (Unique)
email	Descricaosessao
palavrapasse	Ipaudio
nivelseguranca	Ipvideo
	Ipwb
	Iptexto
	porta_audio
	porta_video
	porta_wb
	porta_texto
	Ttl
	Iniciosessao
	Fimsessao
	Emailmoderador

Quadro 8 - Tabelas *mysql* para gestão de sessões e de utilizadores.

Estas estruturas de dados destinam-se à manipulação e manutenção via PHP4. O código PHP embutido nas páginas HTML permitirá salvaguardar os níveis de segurança diferenciados para os diferentes tipos de utilizadores que navegam pelo sistema. Mais concretamente, um requisito de segurança é verificado antes do *browser* apresentar qualquer conteúdo sujeito a regras de segurança, para além de ser um utilizador registado o sistema verifica se o seu nível de segurança é maior ou igual ao exigido por cada página a que tenta aceder.

Quanto aos utilizadores será necessário um processo de registo sucinto para que fiquem habilitados ao nível de alunos gerais, o facto de se inscreverem num determinado curso permite-lhes ascender o nível de segurança a que podem aceder, nomeadamente, faculta-lhes acesso a todos os conteúdos relativos ao curso em que se matricularam. Já os tutores terão a sua conta criada pelo administrador do sistema, que lhes conferirá um número superior de privilégios, tais como, a já aventada possibilidade de criação/alteração de novas sessões *multicast* no âmbito da sua leccionação.

Quanto à criação, manutenção e publicitação das sessões *multicast*, algumas páginas PHP4, com acesso condicionado aos utilizadores autorizados, são responsáveis por inserir/alterar/eliminar registos na tabela **phpGS_sesoes**. O conteúdo da tabela das sessões será tornado público à sub-comunidade relacionada com um determinado curso/disciplina, permitindo a adesão adaptativa aos utilizadores credenciados.

Os média utilizados para as sessões passam por: áudio, vídeo, *whiteboard* (quadro branco) e texto. Ao criador da sessão cabe decidir quais os média que quer envolver na conferência. Cada um destes recursos fará uso de um canal RTP independente, ao qual está atribuído um endereço IP *multicast* e uma porta livre (valor par entre 1024 e 65536). Idealmente estes quatro endereços *multicast* deveriam estar permanentemente atribuídos à nossa Instituição, no entanto dado o cariz experimental deste trabalho, a escolha dos endereços foi aleatória, dentro do espaço de endereçamento de âmbito administrativo, e não passou por requerimento à entidade internacional reguladora do espaço de endereçamento IP, a IANA.

A um nível mais técnico, refira-se que antes de proceder a qualquer inserção ou alteração de dados nas tabelas apresentadas, torna-se indispensável garantir a validação dos dados, impedindo situações que poriam em risco todo o sistema, como sejam o desrespeito pela gama de endereços IP da classe D, a correcção na definição das portas, um valor de *ttl* dentro da gama admissível, etc. Para que esta correcção de dados se verifique foi preciso embutir código *javascript* a montante de cada transacção para efectuar o *parsing* dos dados. Sem prejuízo do apresentado, optou-se nalguns casos pelo preenchimento automático de alguns campos pela aplicação, como seja: seleccionar um qualquer média origina o aparecimento do respectivo endereço *multicast* IP bem como a porta associada. Obviamente, tais automatismos não são uma imposição, a última "palavra" caberá sempre ao utilizador devidamente autorizado.

6.5. Módulo provedor de adaptabilidade

A adaptabilidade dos sistemas clientes quando integram o grupo ou se tornam emissores é um dos pontos fulcrais deste trabalho. Tal capacidade mimetiza a filosofia básica do processo *multicast*, a optimização da utilização dos recursos. As aplicações *multicast*, pela natureza e quantidade de tráfego que transaccionam podem contribuir muito negativamente para a agudização de situações de constrangimento e congestão da infra-estrutura de rede, mesmo ao nível da rede local. A adaptabilidade assegura que os excessos são evitados e que o tráfego que se injecta na rede terá boas probabilidades de atingir o seu fim, evitando situações de colapso e mantendo a conferência "viva".

O objectivo magno deste trabalho passa por garantir a adaptabilidade do lado da aplicação, ou seja, iniciar a aplicação de forma optimizada relativamente às condições

internas (RAM e CPU) e externas (largura de banda e *round-trip-time*) com o intuito de preservar os recursos para os parâmetros críticos (*e.g.*, inteligibilidade), adaptando a pré-parametrização a uma solução de compromisso computada e aplicada de forma transparente para o utilizador.

As aplicações *multicast* de domínio público, são iniciadas normalmente com os parâmetros por defeito, pois não têm a capacidade de avaliar as condições de QoS da rede ou do próprio sistema onde estão instaladas. Assim sendo, para uma correcta utilização tornar-se-ia necessário um procedimento prévio, muito tecnicista, para aproveitar ao máximo os recursos disponíveis na máquina e na ligação. Se se pensar em termos de um sistema de ensino à distância onde os alunos, como participantes mais representativos, não estão tecnicamente preparados para estas exigências, então torna-se fácil perceber a utilidade e o alcance deste módulo.

A adaptação das aplicações é computada em função do perfil de QoS determinado para cada novo membro *multicast* activo. O resultado da computação emanará um conjunto de pré-parametrizações (estratificadas por modos qualitativos) que ajustarão as aplicações para um funcionamento eficaz e eficiente no âmbito das conferências típicas do processo de ensino à distância. A secção 6.6 descreve a metodologia seguida na pré-parametrização adaptativa das aplicações (áudio/vídeo) envolvidas.

O módulo de *middleware*, provedor de adaptabilidade, é constituído por cinco componentes básicos (ver **Figura 27**):

1. *Applet* prospector de condições e calculador do modo de adaptação - *Auscultador.class*;
2. Formulário HTML para visualização das propriedades e recursos disponíveis no sistema;
3. *Applet* que lê os parâmetros do formulário para os passar ao módulo seguinte - *leParametros.class*;
4. Formulário HTML para visualização/edição da parametrização adaptativa e lançamento do módulo seguinte;
5. *Applet* de iniciação das aplicações *multicast* - *IniciaApAdaptada.class*.

Seguidamente descreve-se a funcionalidade destes componentes que funcionam em estreita colaboração e comunicando entre si pela tecnologia *LiveConnet*¹⁷.

¹⁷ Tecnologia descrita em *java.sun.com*

6.5.1. *Applet* prospector de condições

Inicialmente, este módulo é responsável por auscultar as condições inerentes à ligação do membro com o grupo, em termos de largura de banda disponível e *rtt - round trip time*. A obtenção destes dois parâmetros de QoS é facultada por um módulo *javascript* que figura na página *web* que integra este *applet*. O processo de cálculo da largura de banda disponível e do *rtt* baseia-se na possibilidade, via código *javascript*, de importar ficheiros de imagens do servidor *web* para a memória do sistema cliente, o propósito inicial da introdução desta possibilidade foi permitir que as imagens de uma página fossem pré-carregadas em memória para posteriormente serem apresentadas pela página ao mesmo tempo que o conteúdo textual. Utilizando esta técnica apenas para importar um ficheiro (de 100 KB, por exemplo) é possível calcular a taxa de transmissão com o servidor *web*, bastando medir o lapso de tempo decorrido no processo, para obter por regra matemática a taxa de bps transmitidos. Se se utilizar um ficheiro de dimensão muito reduzida (48 bits) é possível obter o *rtt*. Ambas medições são várias vezes repetidas para apurar por média móvel um valor mais fidedigno. Refira-se que a linguagem *java* permite efectuar uma emulação do processo de *ping* (apenas UDP) para o servidor, não se usou tal opção pelos custos em termos de segurança, pois obrigaria a ter uma porta UDP aberta para responder.

Através de um certificado digital, que assina digitalmente este *applet*, este módulo executará também funções que extravasam as limitações de segurança de um *applet* comum, como é o caso da criação de ficheiros no sistema cliente. Inicialmente, o *applet* recolhe indicadores que permitem avaliar as propriedades físicas do sistema cliente, mormente ao nível da qualidade e quantidade dos seus recursos de processamento e armazenamento. Com toda a informação obtida é possível elaborar uma avaliação do binómio sistema/ligação, com vista à computação da parametrização mais adequada às aplicações *multicast* a despoletar. Os parâmetros irão ser passados aos restantes módulos do *middleware* até chegarem ao *applet*, que os utiliza para pré-parametrizar a execução das aplicações.

Em termos técnicos, o *applet* *Auscultador.class*, assinado digitalmente e guardado no *cabinet file* *AuscultadorIE.cab* para o *browser* da *Microsoft*, ou ficheiro *AuscultadorNN.jar* para o caso do *browser* da *Netscape*, terá o seguinte comportamento:

- avaliará as condições de interligação, recolhendo a seguintes métricas: largura de banda disponível para a ligação cliente/servidor e *round-trip time*;

- utilizará as propriedades do sistema cliente disponíveis para os *applets*; para verificar qual o sistema operativo do cliente "os.name", e também a directoria "user.home";
- com os dados do ponto anterior executará um dos dois comandos seguintes, dependendo do tipo de sistema operativo detectado:
 - sistemas Windows

Executará o comando seguinte na pasta de residência da aplicação MSINFO32.EXE:

```
msinfo32 /report "user.home"/propriedades.tmp /categories +SystemSummary
```

O resultado é a gravação de um relatório das características do sistema cliente num ficheiro temporário, na directoria fornecida pela propriedade "user.info", que mais tarde será acedido pelo mesmo *applet* e dele extraídas as propriedades desejadas: memória total e livre, o tipo de CPU e carga do processador.

Por questões de diferentes versões do MSINFO32.EXE, o sistema foi desenvolvido para operar em SOs *Windows 2000* e *XP*.

- sistemas Unix (Linux, Sun Os e Solaris)

Será executado o comando:

```
top > "user.home"/propriedades.tmp
```

O resultado será a gravação num ficheiro temporário de uma "radiografia" ao desempenho actual do sistema, onde constam parâmetros que interessam para a futura parametrização adaptativa. O *applet* procurará nesse ficheiro dados para computação da adaptabilidade, seguidamente o ficheiro é eliminado.

- enviará para uma página HTML, mais concretamente para elementos de formulário nela presentes, os valores das propriedades relevantes, bem como os resultados da computação que originarão os parâmetros para iniciação adaptada das aplicações *multicast*. Para tal utiliza-se a tecnologia *LiveConnect*, que permite a comunicação *applets*« *javascript*.
- finalmente, chamará outro documento *web*, que contém o formulário prévio ao lançamento das aplicações de forma adaptada, essa página permite

opcionalmente, a utilizadores experimentados, realizar edições nos parâmetros. Os parâmetros são passados à página *web* despoletada, por inclusão no URL em forma de *query*. O seguinte excerto de código mostra como se procede.

```
try{
    getAppletContext().showDocument(new URL
("http://www.esa.ipb.pt/multicast/adapt/conexao.php?" + "%lbm=" + par_lb + "%fps=" +
par_fps + "%cdv=" + par_cod_v + "%cor=" + par_cor + "%cda=" + par_cod_a) , "_blank");
}
catch (MalformedURLException e) {
    System.out.println("URL malformado");
}
```

Uma outra contingência inicialmente prevista, que se prendia com o facto de, em casos graves de condições de transmissão, se poder subtrair a emissão vídeo, reduzindo a comunicação a áudio-conferência, foi descartada por se ter concluído que desvirtuava o processo pedagógico, provocando nos elementos do grupo que participavam como receptores uma sensação de perda de contacto e conseqüentemente redução drástica da atenção, o que interferia fatalmente nos objectivos equacionados.

6.5.2. Formulário HTML de visualização dos dados recolhidos

Neste formulário, o utilizador terá o *feedback* do início do processo de execução adaptada das aplicações *multicast* de áudio e de vídeo, podendo visionar os dados recolhidos e os resultados da computação para a adaptação de parâmetros de QoS das aplicações. Os primeiros dados a ser recolhidos são os da interligação, a obtenção dos dados do sistema desencadeia-se depois de accionar o botão presente no formulário para o efeito, que despoleta o *applet* anteriormente descrito.

6.5.3. *Applet leParametros.class*

Este é um *applet* muito simples (ver Anexo II) que, ao contrário dos outros dois que integram o *middleware*, não necessita de ser assinado digitalmente, pois não executa nenhuma operação que careça de privilégios suplementares. Basicamente

destina-se a permitir a comunicação entre duas páginas HTML, ou seja o formulário anterior e o que se segue. Através da análise da *query* que foi apensa no seu URL, pelo *applet Auscultador.class*, logra obter os parâmetros da adaptação. Estes apresentam-se, já na sintaxe usada nas linhas de comandos que pré-parametrizam as aplicações *multicast*.

6.5.4. Formulário HTML de visualização/edição da parametrização adaptativa

Este formulário é um passo importante que precede o início do estabelecimento da conferência, pois permite ao utilizador visualizar e aceitar a parametrização otimizada, automaticamente calculada, ou, em alternativa, provê também a liberdade de reconfigurar alguns parâmetros das aplicações *multicast* manualmente, contrariando a configuração fornecida algoritmicamente. Obviamente, as alterações são interditas aos parâmetros básicos de descrição da sessão. Nos parâmetros reconfiguráveis foi salvaguardada a correção dos dados, incluído código embutido (*javascript*) que efectua a validação dos dados alterados.

Este passo finaliza com a aceitação do início da conferência com as parametrizações e aplicações ratificadas.

6.5.5. Applet de iniciação das ferramentas *multicast*

O último componente do módulo provedor de adaptabilidade é outro *applet* (*IniciaApAdaptada.class*) que despoleta as aplicações *multicast* de forma adaptada às condições verificadas, recebendo os parâmetros do formulário anteriormente exposto via *LiveConnect*.

Este *applet* é invocado numa página gerada (por *Javascript*) pela página que inclui o formulário anterior. A geração do HTML inclui a passagem, ao *applet*, dos parâmetros necessários, desde a descrição da sessão aos parâmetros das aplicações onde se efectuará adaptação.

O seguinte trecho de código deste *applet* mostra a composição dos comandos antes de serem executados.

```

execcommand1 = "vic -t "+val_ttl+" -B "+val_lbm+param_cor+" -f "+val_cdv+" -F
"+val_fps+" -X transmitOnStartup="+val_video+" "+val_enderecov+"/"+val_portav;

execcommand2 = "rat -t "+val_ttl+" -f "+val_cda+"/dvi/gsm "+val_enderecoa+"/"+
val_portaa;

```

```

public void actionPerformed(ActionEvent e) {
    if( (e.getActionCommand()).equals("Iniciar Conferência")) {
        try{
            PrivilegeManager.enablePrivilege("UniversalExecAccess"); // necessário no NN
        } catch(Exception cnfe) {
            System.out.println("netscape.security.PrivilegeManager class not found");
        }
        try {
            Process proc1 = Runtime.getRuntime().exec(execcommand1) ;
            Process proc2 = Runtime.getRuntime().exec(execcommand2) ;
        }
        catch(IOException ieo) {
            System.out.println("Impossível abrir" + execcommand1 + " ou " + execcommand2);
        }
    }
}

```

A possibilidade de um *applet* executar aplicações no sistema cliente está dependente da autorização explícita do utilizador desse sistema. Assim torna-se necessário, tal como no primeiro *applet* do *middleware*, certificar e assinar digitalmente este código, indicando ao utilizador final que pode confiar na origem, no conteúdo e na actuação do *applet*. Este tema das questões de segurança dos *applets* será abordado na secção 6.6.1.

Fica desta forma concluído todo o processo de adesão a uma sessão, feito exclusivamente via *browser*, de uma forma muito mais facilitada e acessível ao comum utilizador.

6.6. Parametrização das aplicações *multicast* de áudio e vídeo

Com a parametrização das aplicações *multicast*, os clientes terão os seus sistemas adaptativamente configurados para quando desejarem tornar-se activos, almeja-se basicamente que as transmissões possam ser eficazes e eficientes.

A parametrização adaptativa das aplicações *multicast* é realizada computando todos os dados recolhidos, os internos (Ocupação do CPU, RAM disponível) e externos (Largura de Banda disponível e *round-trip time*), os valores apurados são as variáveis de uma fórmula que calcula os ajustamentos a imprimir à aplicação, previamente ao seu

lançamento. Esses ajustamentos vão no sentido de regular a taxa de imagens a emitir/receber (*frame rate*), a largura de banda máxima a utilizar e a qualidade das imagens (codificação e riqueza cromática), isto na aplicação de transmissão/recepção de vídeo, o *vic*. No áudio, pode-se simplesmente, modelar a qualidade do áudio, optando por formatos de codificação com melhor ou pior qualidade.

De acordo com o exposto na secção 4.3.2.1, a parametrização da aplicação de transmissão/recepção de vídeo, o *vic*, será pré-programada, por linha de comandos, usando as seguintes opções:

```
vic [ -B kbps ] [ -c dither ] [ -f format ] [ -F fps ] [ -X
resource=value ] dest/port
```

Os parâmetros representativos para este trabalho, no aspecto da adaptabilidade, são os seguintes, vistos em detalhe:

-B	Ajusta a utilização máxima de largura de banda disponível (kilobits per second).
-c	Ajusta a paleta de cores, é dispensável nos casos onde abundam as condições de vídeo. Gray - para tons de cinzento
-f	Usa a codificação de vídeo desejada, (apenas se utilizam os dois seguintes): h263 H.263 h261 Intra H.261
-F	Ajusta o número máximo de imagens por segundo (frames per second).

Para configurar o *rat*, utilizar-se-á apenas a opção "**-f**" relativa ao formato da codificação do áudio. Os formatos usados serão os seguintes: *lpc*, *gsm*, *dvi*, *pcm* e *l16*; já descritos na secção 4.3.1.1.

```
rat [ -f format ] dest/port
```

Em [Watson01] é possível encontrar, de forma completa e detalhada, todas as condições necessárias a uma conferência multimédia eficaz, desde o número de *fps* mínimos e desejáveis, à qualidade das imagens, abordando todos os elementos que impactam a qualidade da comunicação em grupo. Mesmo sabendo que com 10 *fps* se pode manter uma CMM "viva", são preferíveis os movimentos perfeitamente percebidos das 30 *fps* (máximo do *vic*), ou a riqueza de uma imagem de 24 bits de

cor aos tons de cinzento. Assim, vamos tentar encontrar uma solução de compromisso entre as condições disponíveis e a forma como são aproveitadas.

Para calcular o grau qualitativo de adaptação das aplicações criou-se uma fórmula (1), cujo resultado conduzirá a um dos modos de adaptação presentes na **Tabela 2**. A fórmula emergiu do trabalho experimental (ver secção 6.7) com sistemas cliente distintos e usando diferentes configurações das aplicações. Depois de verificações de proporcionalidade, e estabelecendo, basicamente um índice de disponibilidade e abundância de recursos para satisfazer o perfil de QoS de um novo membro do grupo *multicast*, chegou-se à fórmula (1). Esta, não incorpora qualquer influência de trabalhos relacionados, pois foi impossível encontrar tais trabalhos na literatura e publicações científicas da especialidade.

$$M = (\text{int}) (B/(\text{RTT}/2) + \text{ML}/P) *K \quad (1)$$

Onde:

M = Modo de ajuste seleccionado, a consultar na **Tabela 2**;

B = Largura de banda em Kbps;

RTT = *round-trip time* em ms;

ML = Memória livre em MB;

P = Taxa de ocupação processador;

K = 1/50 - trata-se de uma constante com o fim reduzir o resultado à escala que gradua os modos (1 a 5).

O resultado (M) da fórmula indexará um modo qualitativo de adaptação para as aplicações de áudio e vídeo. A estratificação dos modos qualitativos baseia-se numa escala de gradação da qualidade, sendo o escalão mínimo correspondente ao requisitos básicos para funcionamento eficaz das aplicações de conferência multimédia. Optou-se por cinco modos de adaptação, sendo um intermédio, o terceiro (que corresponde praticamente ao modo de iniciação por defeito das ferramentas *multicast*), com dois de qualidade superior e outros dois de qualidade inferior. A escala está baseada em estudos de percepção da qualidade do áudio e do vídeo [Watson01, Hollier97]. A **Tabela 2** mostra os parâmetros de qualidade (audio+vídeo) associados a cada modo qualitativo.

MODO	LARGURA DE BANDA MÁXIMA	TAXA DE FRAMES	CODEC VÍDEO	CORES	CODEC ÁUDIO
5	1 Mbps	30 fps	H.261	Sim	L16
4	512 Kbps	25 fps	H.261	Sim	PCM
3	256 Kbps	20 fps	H.261	Sim	DVI
2	128 Kbps	15 fps	H.263	Sim	GSM
1	64 Kbps	10 fps	H.263	Não	LPC

Tabela 2 - Tabela de modos de pré-parametrização das aplicações *multicast rat* e *vic* para emissão/recepção de áudio e vídeo.

Obviamente, o mínimo será sempre o modo 1, mesmo que a fórmula não atinja sequer esse valor. Da mesma forma o máximo será sempre 5, mesmo que a fórmula produza um valor maior. Cabe então ao *applet* (*Auscultador.class*) executar o algoritmo que põe em marcha o cálculo da fórmula e decidir o tipo de pré-parametrização otimizado. Por forma a minimizar erros de auscultação imprópria, recorre-se a várias auscultações (o sistema funciona com quatro medições para o cálculo da largura de banda e dez para o *rtt*) e considera-se a média móvel dos valores mensurados.

Quanto à obtenção do *rtt* e taxa de ocupação do processador, estes terão sempre um valor mínimo de 1 para evitar divisões por zero.

Para evitar incongruências, a largura máxima a autorizar à aplicação não deverá ser superior à detectada. Assim, no caso raro da fórmula produzir um modo de adaptação, que contenha uma largura de banda máxima maior que a disponível, o modo será ciclicamente decrementado, no pior dos casos até ao valor mínimo.

Já com a aplicação em execução e com a conferência a decorrer, o utilizador passa a controlar integralmente a mesma, ficando a seu cargo qualquer reconfiguração ou término da participação, inclusivamente porque há parâmetros que não são pré-configuráveis, por exemplo a resolução da imagem.

A adaptabilidade não é continuada no tempo de transmissão, ocorre apenas na iniciação das conferências, assim se as condições se alterem as aplicações *multicast* não se auto-adaptam. Desejavelmente as aplicações *multicast* deveriam possuir a capacidade de auto-adaptação, como aliás é referido no capítulo das conclusões e trabalho futuro.

Sendo o *Javascript* código *open-source*, consultável no *browser*, não se incluiu na dissertação, o código-fonte dos *applets* descritos apresenta-se no Anexo II.

6.6.1. Necessidades de segurança na execução de *applets* no sistema cliente

Como um *applet* é um programa executado no *browser/JVM*¹⁸ do sistema cliente, cabe a esse sistema cliente (pelo seu utilizador) autorizar/negar a execução de tarefas que requerem privilégios de segurança para além dos associados ao estatuto de *applet* convencional.

Apesar de utilizar ferramentas de assinatura de *applets* diferentes, ambos os *browsers* mais difundidos, o *Internet Explorer* (IE) e o *Netscape Navigator* (NN), permitem a execução de *applets* com privilégios de segurança aditivados, sempre que o utilizador aceda ao requerimento de autorização emitido pelo servidor. Este pedido acontece em forma de aceitação de um certificado digital, que assina o *applet*. A solicitação de autorização, é apresentada ao utilizador num janela onde constam a identificação do autor, a certificação da origem e os privilégios requeridos. Caso o utilizador aceite o certificado implicará a permissão de execução do *applet*. O certificado digital será guardado temporariamente na *CA Store* do *browser*, um ficheiro que colecta certificados provenientes da navegação na Internet por sites que os solicitem.

No caso do *browser* NN, são algumas dezenas de privilégios que se podem atribuir aos *applets*, adicionados através do método *enablePrivilege* da classe *PrivilegeManager*, presente no *package* `netscape.security`. Veja-se o exemplo de código seguinte, que garante ao *applet*, privilégios de leitura de ficheiros no sistema local:

```
PrivilegeManager.enablePrivilege("UniversalFileRead");
```

No trabalho presente houve necessidade de recorrer a alguns deles, nomeadamente: *UniversalPropertyRead*, *UniversalExecAccess* e *UniversalFileAccess*, explicados a seguir no **Quadro 9**. Esta listagem, não sendo exaustiva, procura exemplificar o aventado e expor os privilégios atribuídos aos *applets* usados neste trabalho.

<i>UniversalThreadAccess</i>	Permite a manipulação de <i>threads</i> não pertencentes ao <i>applet</i>
<i>UniversalThreadGroupAccess</i>	Permite a manipulação de grupos de <i>threads</i> não pertencentes ao <i>applet</i>
<i>UniversalExecAccess</i>	Permite a execução de sub-processos. É usado neste projecto aquando da execução das aplicações <i>multicast</i> despoletadas pelo <i>applet</i> , bem como na execução de comandos no sistema cliente que visam adquirir as suas propriedades físicas para posterior avaliação e utilização na adaptabilidade das aplicações <i>multicast</i> a utilizar. O resultado da execução desses comandos é guardado num ficheiro temporário que posteriormente será analisado
<i>UniversalExitAccess</i>	Permite terminar a sessão do <i>browser</i>
<i>UniversalLinkAccess</i>	Permite a ligação a <i>dynamically loaded libraries</i> (DLLs)
<i>UniversalPropertyWrite</i>	Permite ler e escrever as propriedades do sistema, emanadas pela JVM
<i>UniversalPropertyRead</i>	Permite a leitura das propriedades do sistema. É usado neste trabalho para obter algumas propriedades importantes para a computação dos parâmetros que irão determinar a adaptabilidade das aplicações <i>multicast</i>
<i>UniversalFileRead</i>	Permite a leitura de qualquer ficheiro no sistema de ficheiros local
<i>UniversalFileWrite</i>	Permite a escrita de qualquer ficheiro no sistema de ficheiros local
<i>UniversalFileDelete</i>	Permite a eliminação de qualquer ficheiro no sistema de ficheiros local
<i>UniversalFileAccess</i>	Permite a leitura, escrita e eliminação de qualquer ficheiro no sistema de ficheiros local. É utilizado no projecto aquando da leitura das propriedades guardadas em ficheiro temporário. Este ficheiro foi originado pelo <i>applet Auscultador.class</i> , por comandos executados selectivamente para cada sistema operativo. Depois de extraídas as propriedades necessárias à parametrização adaptativa, o ficheiro temporário é eliminado.
<i>UniversalMulticast</i>	Permite estabelecer <i>Multicast IP</i>
<i>SuperUser</i>	Permite todos os privilégios
<i>Os privilégios restantes podem ser obtidos no manual de JAVA da Sun Microsystems</i>	

Quadro 9 - Alguns métodos para incrementar os privilégios de *applets* no *browser NN*.

Já no caso do *browser IE*, trata-se do *package com.ms.security*, na classe *PermissionID* que inclui vários métodos para garantir privilégios adicionais aos *applets*. O **Quadro 10** contém uma descrição sumária de alguns dos métodos, logicamente os utilizados neste trabalho são os equiparáveis aos adiantados para o *browser NN*, nomeadamente os métodos: EXEC, PROPERTY e FILEIO.

CLIENTSTORE	Permite o acesso às APIs armazenadas pelo sistema cliente
EXEC	Permite a execução de comandos, aplicações no sistema cliente
FILEIO	Permite operações de I/O no sistema de ficheiros do sistema local
NETIO	Permite aceder às operações de I/O que ocorrem na ligação à rede
PRINTING	Permite aceder aos recursos de impressão do sistema local
SECURITY	Permite o acesso às APIs de segurança do JDK- <i>Java Development Kit</i>
MULTIMEDIA	Permite aceder aos dispositivos multimédia do sistema local
PROPERTY	Permite aceder às propriedades do sistema local
REFLECTION	Permite utilizar <i>reflection</i> APIs
REGISTRY	Permite aceder e editar o <i>Registry</i> do SO local
SYSTEM	Tipo especial que permite o acesso a todos os privilégios
<i>Os privilégios restantes podem ser obtidos no manual de JAVA da Microsoft.</i>	

Quadro 10 - Alguns métodos para incrementar os privilégios de *applets* no *browser IE*.

Os testes incidiram maioritariamente sobre a transmissão *multicast* de vídeo, por ser a mais exigente em termos de recursos, quer sejam de processamento ou de rede.

Nos esforços de investigação que foram conduzidos para complementar a consecução deste trabalho procurou-se testar em cenários díspares, com recurso a largura de banda abundante, mas também em condições de exiguidade de recursos.

6.7.1. Cenários experimentais e resultados

Para além dos cenários de disponibilidade de largura de banda já adiantados, definiram-se quatro cenários experimentais, relativos às condições de captação da imagem, para a utilização das aplicações *multicast* de vídeo. São eles:

- qualidade por defeito, cena pouco movimentada;
- qualidade por defeito, cena movimentada;
- qualidade máxima, cena pouco movimentada;
- qualidade máxima, cena movimentada.

Desta forma pode-se inferir o desempenho das aplicações nas várias situações.

Relacionaram-se também os modos de adaptação das aplicações *multicast* (ver **Tabela 2**) com os cenários experimentais supra listados.

Principiou-se por medir, em cada máquina usada nos testes, a carga média do CPU afectada numa transmissão áudio/vídeo para um grupo *multicast*. Efectuou-se a medição para cada um dos cinco modos de adaptação previstos. Os resultados estão presentes no gráfico da **Figura 40**.

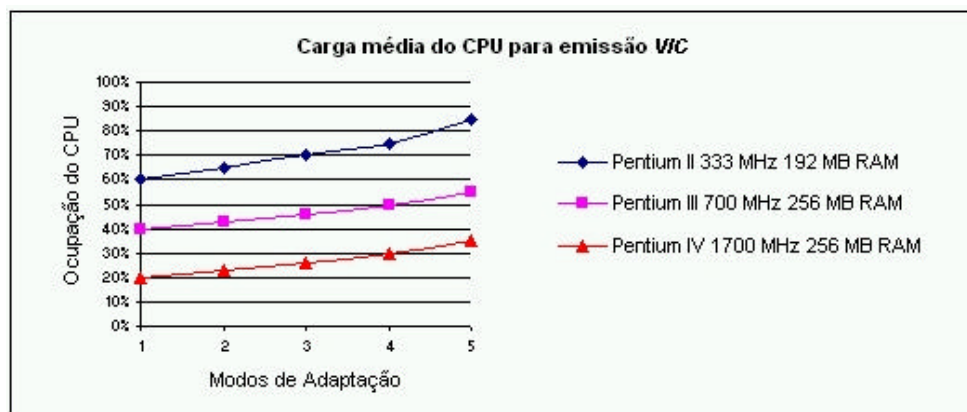


Figura 40 - Carga do CPU, verificada em diferentes máquinas, na transmissão de vídeo pelo *vic*.

Mediram-se também, as necessidades de largura de banda, para a transmissão de vídeo (a mais exigente) nos cinco modos de operação. Em cada modo testou-se particularmente para cada cenário no tocante à captação de imagens. Os resultados são visíveis na **Figura 41**.

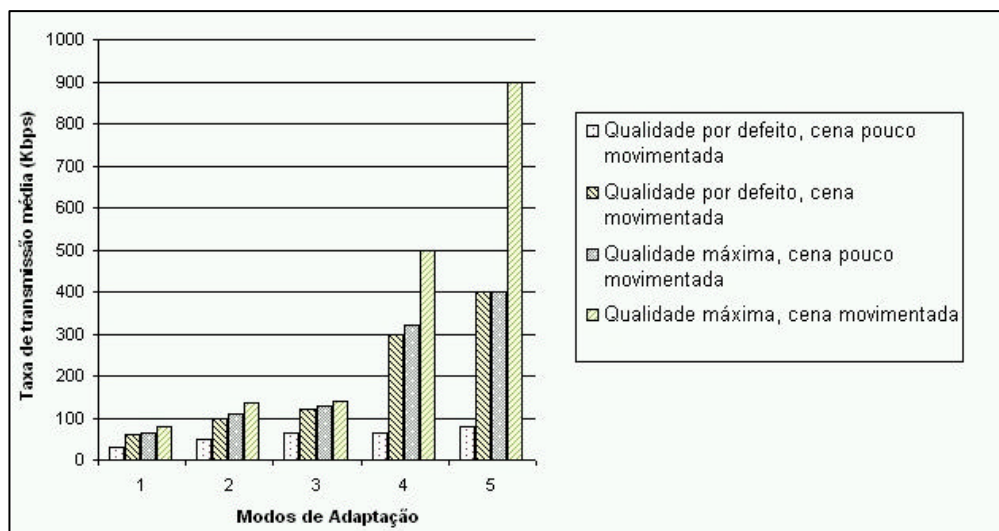


Figura 41 - Necessidades de largura de banda para a transmissão de vídeo pelo *vic* nos diferentes modos de adaptação e em diferentes cenários.

Os testes à escalabilidade das comunicações de um grupo *multicast* fizeram-se por simulação, considerando todos os membros como emissores de vídeo, para uma sub-rede com recursos limitados. Produziram-se duas versões de simulação, uma com utilização do sistema de adaptação desenvolvido e outra sem adaptação, utilizando portanto, a configuração por defeito das aplicações *multicast*. A justificação para estas simulações prende-se com o facto de não se possuírem câmaras de vídeo suficientes para equipar dezenas de clientes e efectuar um teste real.

Basicamente, os parâmetros introduzidos no cenário hipotético foram os seguintes:

- a largura de banda máxima é de 4 Mbps, plenamente dedicada ao tráfego *multicast*;
- o valor do *rrt* é proporcional ao número de membros do grupo *multicast*;
- adicionam-se membros até esgotar os recursos de largura de banda;
- os recursos de processamento são considerados elemento neutro.

Se não for usada a adaptação, o *vic* abre, por defeito, configurado para transmitir a 128 Kbps, portanto será esse o valor a decrementar na largura de banda disponível por

cada membro que adere ao grupo *multicast*, desprezando o seu próprio retorno. Vejamos os resultados, da simulação para a abertura do *vic* sem adaptação, na **Figura 42**.

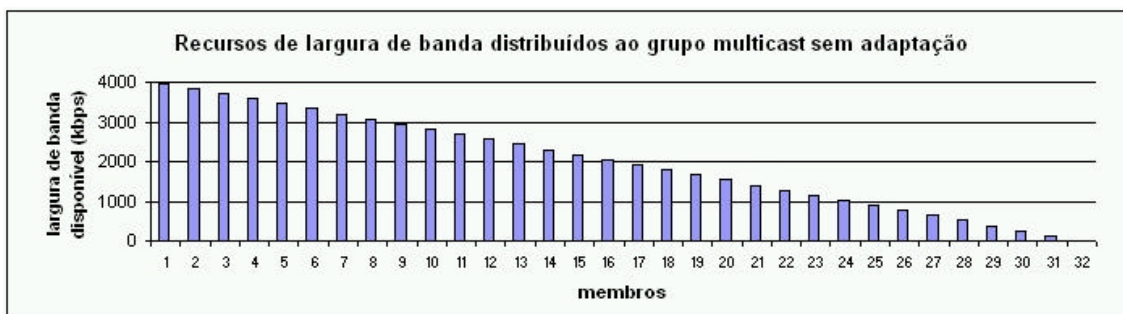


Figura 42 - Afecção dos recursos de largura de banda aos membros do grupo sem utilizar adaptação.

Trata-se de uma redução linear nos recursos disponíveis, o número de participantes atinge 32 antes dos recursos de interligação se esgotarem. A atribuição de recursos de interligação às aplicações não prevê a variabilidade da QoS da rede.

Seguem-se os resultados da simulação para a mesma sessão com utilização do sistema de adaptação desenvolvido (ver **Figura 43**). Assim, aplica-se a cada novo membro, um modo de adaptação ajustado, resultante da fórmula (1). Refira-se que se utilizaram os valores, de consumo de largura de banda, obtidos na experiência ilustrada pela **Figura 41**, no cenário de "Qualidade por defeito, cena movimentada". Isto por se considerar o mais representativo de uma videoconferência para ensino à distância.



Figura 43 - Escalabilidade do grupo *multicast* usando adaptação na qualidade de transmissão do *vic*.

Neste exemplo adaptativo, a escalabilidade foi melhorada em 19%, de 32 participantes passou a admitir-se 38. Os recursos foram distribuídos, até ao quinto participante, de forma a facultar-lhe óptima qualidade de serviço, algo que na simulação anterior não acontecia, levando a que fossem eventualmente desaproveitados. À medida que o grupo vai crescendo, o sistema adaptativo vai limitando os débitos dos novos

membros, a limitação torna-se mais rigorosa depois de 50% dos recursos estarem outorgados. A **Figura 44** mostra os resultados produzidos pela fórmula de adaptação (1) e os modos de pré-parametrização (ver **Tabela 2**) resultantes.

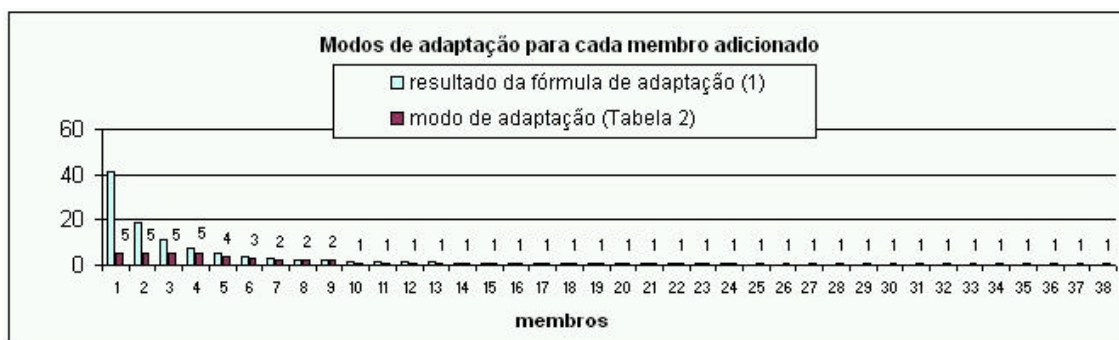


Figura 44 - Modos de adaptação para pré-parametrização das aplicações *multicast* dos novos membros.

Por último testou-se o comportamento do sistema em situação de alguma carga da rede, portanto com menos recursos e comparou-se com a falta de adaptação. A métrica que se utilizou foi o número de pacotes perdidos à medida que se adicionava um novo membro ao grupo *multicast*.

Mesmo com muita largura de banda disponível, o processamento exigido aos encaminhadores é muito avultado, logo notou-se que a perda de pacotes se incrementou consideravelmente após o segundo membro (com menos recursos de processamento) iniciar a transmissão sem recorrer à adaptação.

Num gráfico de utilização de largura de banda (**Figura 45**) consegue-se perceber claramente que o segundo emissor iniciou a actividade pelas 00:00h e terminou passadas duas horas. Nesse período de tempo a taxa de perda de pacotes manteve-se bastante elevada, sobretudo no segundo emissor, retornando a valores baixos depois de cessar a segunda fonte. Refira-se que ambos computadores estavam configurados com a mesma qualidade de transmissão, simplesmente o segundo (Pentium II) corresponde à máquina mais incapaz em termos de processamento e o primeiro (Pentium IV) à mais capaz. Assim, o primeiro consegue gerar cerca de 3 Mbps de tráfego e o segundo queda-se por metade desse valor.

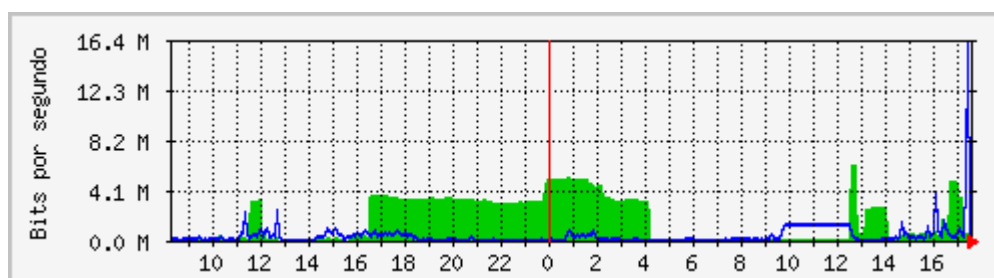


Figura 45 - Sobrecarregar a rede para efectuar testes de transmissão sem abundância de recursos.

No dia seguinte realizaram-se outros testes complementares.

Instalaram-se duas câmaras na máquina mais poderosa, e nessa situação conseguiu-se injectar nas rede cerca de 6 Mbps, como se pode constatar na **Figura 45**, no pico próximo das 12 horas. Se bem que a carga da rede se elevou substancialmente, foi impossível, com os meios disponíveis, levar a rede ao estado de sobrecarga ideal para realizar os testes de medição de perda de pacotes. Uma solução possível passaria por, utilizar uma ferramenta geradora de tráfego ou, possuir mais meios e com mais capacidade de geração de tráfego.

Medir o número de pacotes perdidos não pode ser visto com uma métrica 100% fidedigna no comportamento de aplicações, pois depende de muitos factores externos, como sendo: encaminhadores mal configurados, o tamanho dos *buffers* dos *routers*, a capacidade de processamento das fontes, e até mesmo da qualidade da cablagem. Ainda assim, recorreu-se ao próprio *vic*, que oferece a possibilidade de visualizar a estatística da transmissão RTP, onde se inclui o número de pacotes perdidos.

Pela análise dos pacotes perdidos nas transmissões RTP das fontes, concluiu-se que o formato *Motion Jpeg* revela uma taxa de perdas consideravelmente superior ao formato H.261. Por conseguinte, nenhum dos modos de adaptação usa o *codec jpeg* e, limitou-se a largura de banda máxima utilizada para 1 Mbps.

O sistema de EAD desenvolvido, adapta as aplicações de videoconferência para operarem, no máximo, a 1024 Kbps de vídeo mais 128 Kbps de áudio, logo evita excessos mas não abdica da qualidade. Simplesmente, as máquinas em teste utilizavam o formato *motion JPEG*, o mais oneroso para a rede. Com a mesma qualidade perceptível, o H.261, usado pelo modo de adaptação mais elevado, necessita menos de 1 Mbps para as cenas mais exigentes.

O formato H.261, que é usado na forma adaptativa, demonstrou, além de mais eficiência em termos de compressão da informação, uma menor proporção de pacotes

perdidos, o que em videoconferência significa imagens mais nítidas e com a animação mais realista.

Durante uma hora de medições, em termos de pacotes perdidos, nas condições referidas, verificou-se o comportamento ilustrado na **Figura 46** e **Figura 47**.

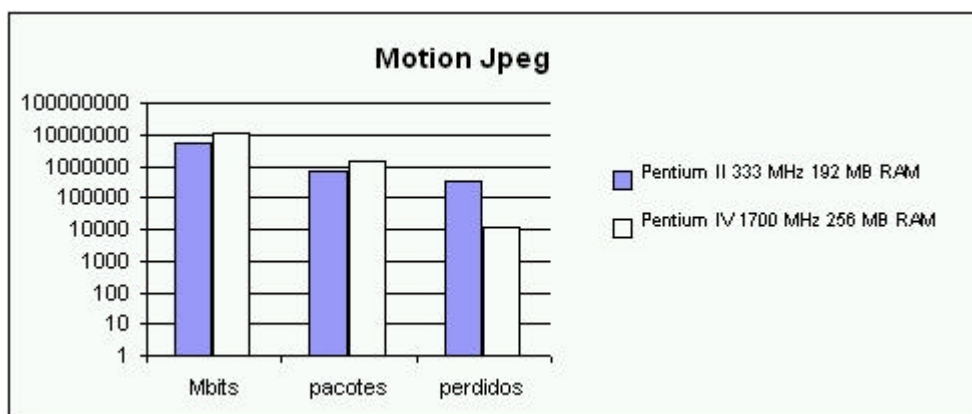


Figura 46 - Taxa de perda de pacotes de vídeo RTP no formato *motion jpeg*, medidos pelo vic.

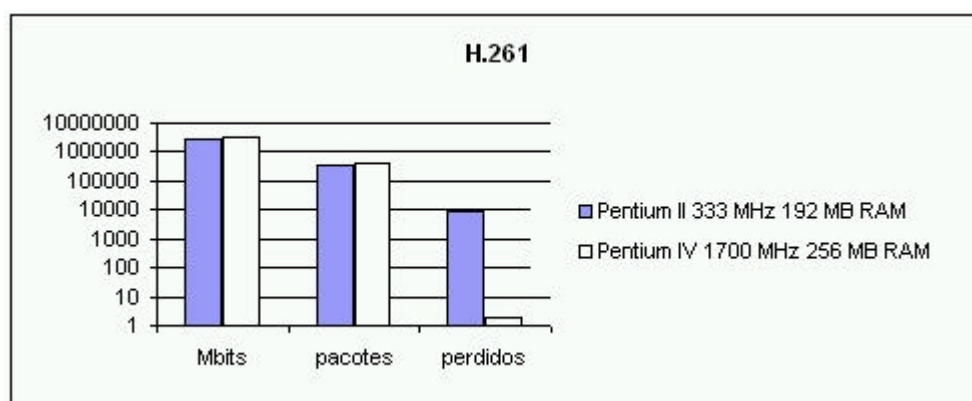


Figura 47 - Taxa de perdas de pacotes em emissão RTP no formato H.261.

Em termos de percentagem, para a máquina Pentium II, a transmissão no formato *Motion Jpeg* perdeu 30% de pacotes, algo grave numa videoconferência. No mesmo formato mas usando a máquina mais poderosa, a taxa de perda desceu para 2%.

Para o formato H.261, as perdas percentuais foram, para os dois sistemas, muito baixas. Cifram-se em 1% para a máquina com menores recursos e 0% para a outra.

6.7.2. Discussão dos resultados

Nos gráficos da **Figura 40** e **Figura 41** pode constatar-se que em termos de carga do CPU e necessidades de largura de banda, os modos de adaptação têm um comportamento adequado à sua ordem, i.e., a um modo de ordem maior corresponde maior dispêndio de recursos de processamento e interligação.

As simulações, apesar de não serem testes reais, demonstram que a escalabilidade também melhora, mas mais importante do que isso, talvez seja verificar que os elementos do grupo que aderem em condições de abundância de recursos podem beneficiar delas, enquanto que no caso da não adaptação não são aproveitados os recursos existentes. É certo que os membros incorporados mais tardiamente recebem poucos meios do sistema, mas a experiência nas sessões do Mbone demonstra que os grupos *multicast* tendem a ser reduzidos.

No caso de escassez de recursos o formato usado no modo adaptativo (H.261) também teve melhor comportamento. Constatou-se que para um cliente com menor capacidade de processamento, usando o formato *Motion Jpeg*, o número de pacotes perdidos crescia para além dos limites que a aplicação pode tolerar antes de perder a eficácia. Enquanto que, com o mesmo cliente, mas despoletando o *vic* na forma adaptativa, o modelo se encarregava de restringir as "ofensas" à rede, o que redundava em menor taxa de perdas. Adicionalmente, em situação de constrangimento da rede, adaptando o débito autorizado evita-se o agravamento de uma situação de congestão, e a rede recupera mais rapidamente.

Outros testes, de natureza subjectiva, poderiam ter sido realizados, mas tal envolveria testar o sistema com e-alunos reais, inquirindo-os acerca da qualidade percebida na utilização dos serviços disponibilizados, algo que não foi equacionado mas que traria certamente valor acrescentado ao trabalho.

Capítulo 7

Conclusões

O *multicast* é inevitável. A Internet dificilmente poderá escalar sem a optimização e conseqüente poupança de recursos que advêm desta técnica. Cria ainda, a possibilidade de desenvolver e utilizar novas e atractivas aplicações que as pessoas desejam. As comunicações *multicast* crescem em importância mas o seu desenvolvimento é lento, trata-se de uma mudança estruturante, certamente eficaz mas ao mesmo tempo difícil e morosa de implementar.

O obstáculo mais óbvio e significativo é a necessidade de ter *routers* com capacidade *multicast* na ligação fim-a-fim entre emissores e receptores, bem como efectuar a sua gestão. Este desejo torna-se mais fácil alcançar numa rede local, numa empresa por exemplo, com gestão centralizada da rede. Acontece porém que o *multicast* prolifera por toda a Internet. O desafio é muito maior se atendermos ao facto de que os

routers multicast não estão sempre e onde são desejáveis, sendo ainda necessário atravessar as zonas não *multicast* com os túneis, conferindo assim, apenas virtualmente, essa capacidade *multicast* a toda a rede. Como resultado, o Mbone é hoje uma malha intrincada de troços *multicast* e túneis cujo desempenho é de certa forma imprevisível.

As horas "mortas" na utilização dos recursos de interligação das LANs e consequentemente das WANs, estão cada vez mais rarefeitas, os congestionamentos são cada vez mais a norma. Todas as escaladas de largura de banda ao nível da infraestrutura de rede parecem diluir-se nas exigências exponenciais dos utilizadores e suas aplicações.

As elevadas necessidades de largura de banda, características de aplicações que suportam um sistema de ensino à distância credível, dificultam em ambientes de serviço de tipo *best-effort*, a complacente participação no processo de *e-learning*. A garantia de QoS em redes IP é a saída airosa que se vislumbra. No entanto, a não existência alargada de redes com QoS é ainda uma realidade. Neste cenário, a capacidade de adaptação das aplicações assume um papel relevante. O presente trabalho pretende ser, também nesta matéria, uma contribuição válida.

A comunicação em grupo é uma área vital da investigação informática, contextualiza quase tudo o que se possa produzir em termos de aplicações e comunicações. Sustentados pelas novas tecnologias comunicacionais, os portais electrónicos, de empresas e instituições de ensino, fornecerão serviços cada vez mais interactivos e colaborativos aos utilizadores. As aplicações emergem, os protocolos tentam acompanhar e aperfeiçoar-se. A solução parece ser passível do seguinte aforismo: *uma multiplicidade de protocolos com diferentes tipos de serviços*.

O ensino à distância via Internet parece estar já semi-instalado, importa aprimorar os meios, os processos e a filosofia. A certificação de qualidade dos cursos, metodologias e agentes deste tipo de ensino, é uma área que necessita de forte impulso para acompanhar esta revolução.

A ideia de ter um canal de televisão/multimédia próprio com a possibilidade de transmitir para todo o mundo, com meios acessíveis, é sem dúvida um poderio imenso, a liberdade toma formas inexistentes até agora.

7.1. Avaliação crítica do trabalho

O trabalho desenvolvido procurou integrar, na sua forma otimizada, um conjunto de soluções conformadas num sistema de ensino à distância baseado em tecnologia *multicast* com preservação de QoS por adaptabilidade.

A adaptação por *middleware* [Li98, Li00] é uma solução recente, que parece ajustar-se às características de heterogeneidade da Internet e aos requisitos das novas aplicações de comunicação em grupo. Operar em multiplataforma e ser independente das aplicações que controla são características de enaltecer. Porém, as aplicações *multicast* disponíveis, carecem de formas de minuciosa parametrização exterior que possibilite a modelação eficaz do seu funcionamento.

A adaptabilidade não é continuada no tempo de transmissão, ocorre apenas na iniciação das conferências, assim, caso as condições se alterem as aplicações *multicast* não se auto-adaptam. Desejavelmente as futuras aplicações *multicast* deveriam possuir a capacidade de auto-adaptação.

O projecto é uma realidade aberta à comunidade, já que, não se encontra em laboratórios inacessíveis aos potenciais interessados, mas sim democratizado, publicado na Internet, onde pode ser consultado e avaliado por todos.

7.1.1. Descrição genérica das características da plataforma

A academia virtual está acessível em <http://www.esa.ipb.pt/multicast>, disponibilizando de forma integrada e transparente para o utilizador (e-aluno e e-tutor), serviços avançados de CMMs (Conferências Multimédia *Multicast*), fornecendo serviços síncronos, assíncronos e híbridos de divulgação do saber, pretensamente para implementação de boas práticas pedagógicas.

Em [Nevin94] defende-se que o ensino, sobretudo ao nível superior, pode beneficiar com a inclusão de meios telemáticos e multimediáticos para *e-learning*. Um dos preceitos apontados para garantir o sucesso das aplicações de conferência em ensino à distância, prende-se com a simplificação das suas parametrizações. Para um utilizador comum, aceder a uma conferência multimédia deverá ser tão simples com ligar o televisor e seleccionar o programa. No sistema desenvolvido procurou-se, não somente

beneficiar a usabilidade mas também torná-la proactiva, havendo a preocupação de combinar a simplificação da interacção com a eficaz e eficiente utilização dos recursos.

O trabalho desenvolvido, mesmo nos serviços avançados de CMM interactiva e adaptativa, apenas exige ao utilizador que, depois da sua autenticação no sistema de EAD, siga a hiperligação relativa ao tipo de conferência desejada para assistir e/ou participar numa sessão agendada. Enfatizou-se portanto, a melhoria da usabilidade das aplicações envolvidas.

7.1.2. Principais resultados

O sistema de EAD desenvolvido, baseado em aplicações *multicast* adaptadas por *middleware*, mais concretamente o protótipo testado, apresentou bons indicadores de desempenho, que permitem validar o modelo de arquitectura seguido. A adaptação revelou-se vantajosa na sustentação da qualidade das comunicações dos novos membros activos de um grupo *multicast*, sobretudo em cenários de QoS imprevisível, quando aplicada com acuidade a aplicações *real-time* tolerantes. Obtiveram-se também, resultados positivos em matérias como a escalabilidade dos grupos e a preservação dos recursos para os parâmetros críticos das aplicações.

Ao nível da política de adaptabilidade, a intervenção focalizou-se na distribuição dos recursos disponíveis de forma "consciente", ou seja, perante as condições verificadas na ligação e na capacidade de processamento de um determinado novo membro do grupo *multicast*, o sistema configurará transparentemente as aplicações de conferência multimédia para funcionarem proactivamente. Num cenário de escassez de recursos disponíveis as aplicações serão pré-parametrizadas para obterem a QoS funcional mínima, caso existam maiores disponibilidades permitir-se-á às aplicações que usufruam de mais recursos beneficiando a qualidade percebida. Com estes procedimentos, a escalabilidade melhora e a QoS que os utilizadores obtêm terá maior probabilidade de ser sustentada.

Apesar das aplicações serem tolerantes à esporádica degradação da QoS, variando o nível de tolerância para cada media, há parâmetros de qualidade que não podem ser descontinuados, sob pena de destituir de eficácia a comunicação que é fundamental em qualquer processo de ensino. Por exemplo, 10 *fps* deverá ser o mínimo exigível para a transmissão vídeo, logo não fará sentido despende os recursos

disponíveis na obtenção da qualidade perfeita da imagem, se com isso, aniquilarmos a percepção do acompanhamento visual de uma experiência laboratorial.

Através da limitação da utilização de largura de banda, pela escolha de formatos de codificação adequados às necessidades específicas das conferências para EAD ou bem por limitação explicitada por parametrização, podem-se utilizar mais racionalmente os recursos, impedindo situações de exaustão e conseqüente interrupção da QoS, pondo em causa a eficácia do processo pedagógico à distância.

7.1.2.1 Contribuições

De acordo com os objectivos traçados inicialmente, esperava-se contribuir na implementação de práticas "ecológicas" no ambiente da Internet, i.e., utilizar de forma mais eficiente os recursos disponíveis e prevenir a sua exaustão.

As contribuições deste trabalho poderão resumir-se a:

- integração adaptativa de aplicações *multicast* para conferência multimédia interactiva;
- melhoramento da usabilidade das aplicações *multicast*;
- desenvolvimento e promoção do ensino à distância em Portugal;
- desenvolvimento de aplicações *multicast* "conscientes" da qualidade de serviço nos três quadrantes: rede, aplicação e máquina.

O sistema desenvolvido, ao integrar ferramentas de cooperação ou de espaço de trabalho partilhado, pode igualmente fomentar o espírito de colaboração e trabalho em grupo.

7.1.3. Avaliação das condições de desenvolvimento

O *middleware* que governa as formas de adaptação das aplicações é multiplataforma, composto por módulos em *Java (applets)* e *Javascript*. Preparar um *applet* que incorpore funcionalidades que requerem assinatura digital não é trivial, requer uma sequência extensa de procedimentos. Depois do código *java* estar pronto, deverá ser convertido em *class* (processo de *p-code*) para obter o *applet* executável, o passo seguinte é agregar as componentes num único módulo e compactá-lo (CAB para

IE e JAR para NN), posteriormente o *applet* deve ser assinado digitalmente e finalmente transferido para o servidor *web*. Um dos pontos menos favoráveis na programação dos *applets* que compõem o *middleware*, deriva do facto da depuração de erros em tempo de execução no *browser* ser muito rudimentar, simplesmente o *applet* falha e praticamente nenhum retorno é devolvido. Desta forma, é muito penoso superar situações de erros cuja origem é desconhecida, obrigando a rever todo o programa.

Apesar da apregoada compatibilidade entre *browsers* e código que suportam, verificou-se que o navegador NN e o IE não interpretam da mesma maneira o mesmo código, mesmo nas versões mais recentes. Isto obviamente, não é adjuvante para quem desenvolve aplicações para a *web*, pois necessita ter em consideração as duas plataformas e incluir código condicional. Por exemplo, o código *javascript* que gere os menus da interface do sistema corre perfeitamente no IE e apresenta deficiências no NN, isto devido às diferentes interpretações do paradigma de aplicação de *layers*.

O período de testes do sistema também foi complicado, sobretudo devido à falta de ferramentas de domínio público para monitorização especialmente vocacionadas para estudar os recursos envolvidos nas comunicações de um grupo *multicast*. É certo que existem algumas, o MultiMON (<http://www.merci.crc.ca/mbone/MultiMON/>) por exemplo, mas muito insipientes. Optou-se por utilizar uma ferramenta comercial, o MulticastMonitor, disponível em <http://www.multicastmonitor.com/>, que apesar de algo limitada permitiu obter alguns resultados satisfatórios.

7.1.4. Avaliação das condições de teste e utilização

A perspectiva dos "e-alunos" também seria primordial neste estudo, obter a relação dos seus anseios à partida e colectar as suas opiniões depois de testarem o protótipo teria sido de suma importância na concepção e desenvolvimento do sistema, é certo que as directivas seguidas derivam de estudos de conceituados pedagogos da nova era, mas nada substitui a experiência como gostava de eloquir Luis de Camões.

O sistema, apesar de definir que o servidor de conferências (e-tutor) é uno e localizado, pode contemplar tutoração repartida, por exemplo quando um tutor remoto é convidado a colaborar numa palestra. Quando tal acontece, a determinação das condições de ligação dos e-alunos, deveria ter como referência não somente o *site* da

academia virtual mas também o *site* do tutor remoto. Portanto, o cenário de tutoração distribuída levanta novos desafios para que a adaptação seja consistente.

As ferramentas *multicast* utilizadas, sobretudo o *vic* (seguramente pela quantidade de informação processada), demonstrou problemas de estabilidade, sobretudo quando executado em máquinas com baixo desempenho. Também relativamente às aplicações de conferência, tornou-se bastante difícil gerir a área de trabalho quando decorria uma conferência *multicast* envolvendo *vic*, *rat*, *wb* e *nfe*. De facto, são muitas janelas a debitem informação ao mesmo tempo e torna-se difícil acompanhar todos esses media.

7.2. Desenvolvimentos futuros

Alardear de ter uma solução ultimada é sinónimo de não ter compreendido a grandeza do domínio de aplicação aqui envolvido, de facto muito mais haveria a acrescentar, muitas mais teses se redactarão sobre esta matéria e os caminhos continuarão abertos, até mais, se é possível.

Em [Jun99] demonstra-se experimentalmente que, uma arquitectura de serviço de vídeo adaptativa implementando reserva de recursos contribui positivamente para o desempenho do serviço. Num sistema onde a reserva de recursos, neste caso específico a reserva de largura de banda, não seja incluída, verifica-se que a percentagem de *frames* perdidas é muito superior quando comparada com a mesma transmissão num sistema que implemente reserva de largura de banda.

No mesmo estudo, demonstra-se igualmente que a reserva de recursos em termos de CPU, opera comprovadas melhorias ao nível do desempenho da aplicação, tal acontece quando uma aplicação concorrente, de processamento intensivo, põe em causa a qualidade da transmissão. Saem beneficiados ambos os lados, cliente e servidor. Portanto poder-se-ia, em desenvolvimentos futuros do sistema, incluir aplicações *multicast* que efectuem reserva de recursos, como complemento da adaptabilidade e por forma a garantir a sustentabilidade.

A mobilidade também poderia ser uma boa forma de expandir o EAD, com a nova geração de telemóveis UMTS e a possibilidade de correr aplicações Java em aparelhos de computação móveis, possibilitaria a verdadeira omnipresença da escola virtual.

Relativamente à integração adaptativa das aplicações no sistema de EAD desenvolvido, idealmente a forma de exercer a adaptação das aplicações deveria ser dinâmica durante uma sessão, i.e, ser do estilo "*soft-state*", sendo actualizada periodicamente por forma a ajustar-se às variações de disponibilidade de recursos. Aproveitando os que entretanto vão sendo libertados ou por outro lado, redefinir para configuração menos "consumista" uma aplicação tolerante que opera numa rede congestionada. Isto obviamente dentro de parâmetros mínimos de garantia de QoS, pois assistir a uma videoconferência onde a imagem, apesar de perceptível, não tem qualquer estabilidade, pode ser desmotivante.

Faria sentido também, criar espaço para sessões informais entre os alunos, para fomentar o inter-relacionamento e o trabalho em grupo.

No tocante à base de dados que suporta o sistema de EAD, poderia implementar-se uma espécie de *bookmarks* no sistema de armazenamento de material pedagógico. Este, deveria permitir a cada aluno, como que fazer "anotações virtuais" no material didáctico digital, uma espécie de hiperligações que ficariam guardadas na base de dados.

Em termos da interface, poder-se-ia ter seguido o exemplo da ferramenta *Centra One* apresentada no Capítulo 2, onde se procedeu à integração das ferramentas de conferência num espaço de trabalho organizado, de modo a estarem sempre visíveis e controláveis.

A inclusão de comunicações *multicast* seguras e concomitantemente, formas de autenticação dos membros do grupo *multicast* também seria uma mais-valia para o trabalho, permitindo a criação de conferências seguras sempre que informação confidencial fosse intercambiada pelos participantes, *e.g.*, em reuniões de avaliação.

Acredita-se que futuramente, o tráfego *multicast* seja difundido por todos os ISPs, e que protocolos como o RSVP sejam o mais transparentes possível. Refira-se a título de exemplo, que as versões mais recentes do *Netmeeting* incorporam a reserva transparente de recursos, ou seja, se forem executadas em sistemas operativos *Windows 2000*, devidamente configurados para garantir QoS, desencadeiam a reserva de largura de banda. O processo será bem sucedido se todos os *routers* do *path* aceitarem a reserva e se os receptores estiverem autorizados a reservar recursos.

Áreas do conhecimento complementares como a compressão de informação acompanhada da segurança podem catalisar o desenvolvimento do Mbone, no que concerne à entrega em tempo real de, teoricamente, uma grande quantidade de

informação. Reportando-nos a este trabalho, a decisiva melhoria a implementar seria encontrar a forma de garantir que o sistema de EAD continuaria o seu processo de desenvolvimento e maturação, quer por inclusão de parcerias empresariais, de investigação avançada ou por congregação de ambas.

7.3. Inovar é preciso

Para que num país como Portugal se possa começar a esboçar a transição dos modelos comunicacionais tradicionais para as novas estruturas de transmissão e produção do conhecimentos é importante interiorizar a constatação de que os modelos educativos em vigor respondem de forma pouco eficaz às respostas que a sociedade espera do actual sistema. O desfasamento existente entre as necessidades sentidas, as expectativas geradas, e as realidades efectivas está a provocar o descrédito progressivo do sistema e, concomitantemente, dos seus agentes profissionais. O descontentamento social leva a que camadas cada vez mais extensas da população questionem o papel dos formadores e os termos da sua eficácia social.

É possível sustentar a afirmação que defende serem os novos modelos de comunicação telemáticos e multimediáticos potenciais instrumentos de acesso ao saber. Porém, a condição necessária para que esta mutação tecnológica e comunicacional aconteça, pressupõe a apreensão e aceitação por parte da classe docente em geral, do leque global de oportunidades decorrentes da aliança entre as infra-estruturas telemáticas e as novas estratégias multimediáticas de difusão do saber.

Desde já se sublinha a recusa liminar em conceptualizar os novos modelos de comunicação como a panaceia milagrosa capaz de resolver graves questões estruturais de que o Ensino em Portugal ainda enferma. Urge emanar novas metodologias pedagógicas a aplicar aos futuros educandos, tendo consciência da base tecnológica que, a cada dia que passa se vai alastrando e inundando as instituições, as escolas e até as residências particulares.

A revolução, que se pretende pacífica, necessita do somatório das vontades individuais dos agentes educacionais e, neste como noutros campos, comungo do princípio que afirma que o caminho se faz caminhando.

Referências

- [Abdel-Wahab93] H. Abdel-Wahab et al., A roadmap for collocation on the data highway, Second Workshop on Enabling Technologies: Infrastructure for Collaborative Entreprises, pages 9-18, April 1993.
- [Acharya98] A. Acharya, F. Ansari, M. Ott, H. Sanneck, Dynamic QoS for IP switching using RSVP over IPSOFACTO, C&C Research Labs, NEC USA, Princeton, 1998.
- [Ahumada93] A. J. Ahumada, C.H. Null Image quality: a multidimensional problem. In A. B. Watson, ed., Digital Images and Human Vision, MIT Press, Cambridge, MA, 1993.
- [Aleixo98] F. A. Aleixo, RSVP - Uma Implementação de Qualidade de Serviço sobre Tecnologia da Internet, Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1998.
- [Alfano96] M. Alfano, R. Sigle, Controlling QoS in a Collaborative Multimedia Environment, Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing (HPDC 5), New York, 1996.
- [Almeroth99] Kevin C. Almeroth, The Evolution of Multicast: From the Mbone to Inter-Domain Multicast to Internet2 Deployment, Department of Computer Science, University of California, 1999.
- [Amir95] E. Amir, S. McCanne, H. Zhang, An Application Level Video Gateway, in Proc. ACM Multimedia'95, San Francisco, CA, November 1995.
- [Aurrecoechea98] C. Aurrecoechea, A.T. Campbell and L. Hauw, A Survey of QoS Architectures, ACM/Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture, Vol.6 No.3, pg. 138-151, May 1998.
- [Balan02] R. K. Balan, J. P. Sousa, S. Y. Park et al, Adaptation: This Won't Hurt a Bit!, School of Computer Science, Carnegie Mellon University, 2002.
- [BallardieI97] A. Ballardie, Core Based Trees (CBT version 2) Multicast Routing, RFC 2189, September 1997.
- [BallardieII97] A. Ballardie, Core Based Trees (CBT) Multicast Routing Architecture, RFC 2201, September 1997.

- [Banerjee01] S. Banerjee and B. Bhattacharjee. Scalable Secure Group Communication over IP Multicast. In Proceedings of International Conference on Network Protocols, Nov. 2001.
- [Barcellos00] M. Barcellos, V. Roesler, M&M:Multicasting & Multimedia, Universidade do Vale do Rio dos Sinos – UNISINOS, Julho 2000.
- [Bhattacharyya98] S. Bhattacharyya, J. Kurose, D. Towsley, R. Nagarajan, Efficient Multicast Flow Control using Multiple Multicast Groups, IEEE INFOCOM'98, San Francisco, April 1998.
- [Berc96] L. Berc, W. Fenner, S. McCanne, RTP Payload Format for JPEG-compressed Video, July 1996, RFC 2035.
- [Biswas99] S. Biswas, R. Izmailov, and B. Rajagopalan, A QoS-aware routing framework for PIM-SM based IP-multicast. Internet draft: draftbiswas-pim-sm-qos-00.txt, June 1999.
- [Blake98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Services, RFC 2475, IETF, December 1998.
- [Bolot94] J. Bolot, Turletti, Wakeman, Scalable Feedback Control for Multicast Video Distribution in the Internet, INRIA Video Conferencing System (ivs), Institut National de Recherche en Informatique et an Automatique, 1994.
<http://www.inria.fr/rodeo/ivs.html>
- [Bolot95] Jean-Chrysostome Bolot, Hugues Crépin, Andres Veja Garcia, Analysis of Audio Packet Loss in the Internet, INRIA, France, 1995.
- [Bouch97] A. Bouch, Redesigning the RAT user interface. MSc thesis, Department of Computer Science, University College London, University of London, 1997.
- [Braden94] R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: An Overview, RFC 1633, ISI, June 1994.
- [Braden97] R. Braden, L. Zang, S. Berson, S. Herzog, S. Jamin, Resource Reservation Protocol (RSVP) - Version 1 Functional Specification, RFC 2205, September, 1997.
- [Bradner97] S. Bradner, Internet Protocol Multicast Problem Statement, September 1997. Work in Progress.
<draft-bradner-multicast-problem-00.txt>

- [Bunderson70] Bunderson, C.V., and Dunham, J.L., Research Program on Cognitive Abilities and Learning, Final Report, Computer Assisted Instruction Laboratory, The University of Texas at Austin, 1970.
- [Callahan96] John Callahan, Todd Montgomery, Brian Whetten, High Performance, Reliable Multicasting : Foundations for Future Internet Groupware Information, NASA/West Virginia University Software IV&V Facility, January 1996.
- [Canetti98] R. Canetti, B. Pinkas, A Taxonomy of Multicast Security Issues, November, 1998. Internet Draft.
<draft-canetti-secure-multicast-taxonomy-01.txt>
- [Canetti99] R. Canetti, P-C. Cheng, D. Pendarakis, J.R. Rao, P. Rhoatgi and D. Saha, An Architecture for Secure Internet Multicast, IBM, February ,1999. Internet Draft.
<draft-irtf-smug-sec-mcast-arch-00.txt>
- [CanettiGaray99] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas, Multicast Security : A Taxonomy and Efficient Authentication, INFOCOM'99, 1999.
- [Carvalho96] Paulo M. M. de Carvalho, Interaction of Periodic Bursty Traffic Flows in ATM Networks, PhD thesis, Computing Laboratory, University of Kent at Canterbury, December 1996.
- [Casimiro01] A. Casimiro, P. Veríssimo, Using the timely computing base for dependable QoS adaptation, Faculdade de Ciências da Universidade de Lisboa, Departamento de Informática, July 2001.
- [Casner92] Casner, S., Deering, S. First IETF Internet Audiocast, ACM SIGCOMM Computer Communications Review, San Diego - California, pp. 92-97, July 1992.
- [Chatterjee97] S. Chatterjee, J. Sydir, B. Sabata and T. Lawrence. Modeling applications for adaptive QoS-based resource management. In Proceedings of the 2nd IEEE High-Assurance System Engineering Workshop (HASE97), Bethesda, MD, 1997.
- [Chawathe98] Yatin Chawathe, Eric A. Brewer, System support for scalable and fault tolerant internet service. In IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing - Middleware '98, Lake District, UK, September 1998.
- [Chawathe99] Y. Chawathe, S. A. Fink, S. McCanne, E. Brewer, A Proxy Architecture for Reliable Multicast in Heterogeneous Environments, 1999.

- [ChenDing95] J. Chen, Y. Ding, Videoconferencing: A Comprehensive Look, <http://polarbear.student.harvard.edu/~jackchen/paperfin.txt>, 1995.
- [Cheng99] Irene Cheng, A. Basu, L. Chen, QoS Specification and Monitoring for Multimedia Delivery in TeleLearning, Department of Computer Science, University of Alberta, Edmonton, Canada, 1999.
- [Chu01] Yang-Hua Chu, Sanjay G. Rao, Srinivasan Seshan, Hui Zhang, Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture, ACM SIGCOMM, 2001.
- [Clark97] L. Clark, vic usability study. Internal Note 97/1, Dept. of Computer Science, University College London, 1997.
- [ClarkSasse97] L. Clark, and M. A. Sasse, Conceptual design reconsidered - The case of the Internet Session Directory Tool. Proceedings of HCI'97, Bristol, August 12-15, 67-84. Springer, 1997.
- [Cronbach57] L. Cronbach, The Two Disciplines of Scientific Psychology, *American Psychologist*, pp. 671-684, 1957
- [Crowcroft99] J. Crowcroft, M. J. Handley, and I. Wakeman, *Internetworking Multimedia*. UCL Press, 1999.
- [Cui01] Jun-Hong Cui, Aiguo Fei, Mario Gerla, Michalis Faloutsos, An Architecture for Scalable QoS Multicast Provisioning, Computer Science Department, University of California, 2001.
- [Deering89] S. Deering, Host Extensions for IP Multicasting , RFC 1112, , Stanford University, August 1989.
- [Deering91] S. E. Deering, Multicast Routing in a Datagram Internetworks, Ph. D. Thesis, Stanford University, December 1991.
- [Deering97] S. Deering, D. Estrin, D. Farinacci, M. Handley, A. Helmy, V. Jacobson, C. Liu, P. Sharma, D. Thaler, L. Wei, Protocol Independent Multicast-Sparse Mode (PIM-SM) : Protocol Specification, RFC 2117, June 1997.
- [Douglas93] S. G. Douglas, Digital soup, the new abcs of distance learning, *EduCom Review*, pages 22-30, July-August 1993.
- [Ellis91] C.A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware: Some issues and experiences. *Communications of the ACM*, 34(1):38-58, January, 1991.
- [Eriksson94] H. Eriksson, MBONE: The Multicast Backbone, *Communications of the ACM*, 37(8):54-60, August, 1994.

- [Estrin98] D. Estrin, Multicast Enabler and Challenge, Caltech Eathlink Seminar Series, April 1998.
- [Fenner97] B. Fenner, Internet Group Management Protocol - Version 2, RFC 2236, November 1997.
- [Ferguson98] Quality of Service in the Internet: Fact, Fiction, or Compromise?, P. Ferguson, G. Huston, INET'98, Genova, Switzerland, July 1998.
- [Finlayson99] Ross Finlayson, IP Multicast and Firewalls, RFC 2588, May 1999.
- [Floyd98] S. Floyd, V. Jacobson, L. Ching-Gung, S. McCanne, L. Zang, A Reliable Multicast Framework for LightWeight Sessions and Application Level Framing, In Proceedings of ACM SIGCOMM'95, San Francisco, April 1998.
- [Fourmaux98] Olivier Fourmaux, Serge Fdida, Multicast for RSVP Switching - An Extended Multicast Model, Laboratoire d' Informatique, Université Pierre et Marie Curie, Paris, 1998
- [Frederick94I] Frederick, R. Network Video (nv). Xerox Palo Alto Research Center <ftp://ftp.parc.xerox.com/net-research>
- [Frederick94II] R. Frederick, Experiences with real-time software video compression, In Proceedings of Sixth International Workshop on Packet Video (Portland, OR), September 1994.
- [Garg99] A. Garg, S. K. Kasera, R. Kumar and D. Towsley, Measurement of Join Latency on the Mbone, University of Massachussets and Microsoft Corporation, August 1996.
- [Gemmell97] J. Gemmel, Scalable Reliable Multicast Using Erasure-Correcting Re-sends, Technical Report, MSR-TR-97-20, Microsoft Research, Redmond, WA, June 1997.
- [Gemmell97] Jim Gemmel, Eve Schooler, Roger Kermode, A Scalable Multicast Architecture for One-to-Many Telepresentations, Air Force Office of Scientific Research F49620-97-1-0267, 1997.
- [Gerla01] Mario Gerla, Aiguo Fei, Jun-Hong Cui, and Michalis Faloutsos, Aggregated Multicast for Scalable QoS Multicast Provisioning, Department of Computer Science, University of California, 2001.
- [Gilder94] George Gilder, The Bandwidth Tidal Wave, Forbes ASAP, December 1994.

- [Goldberg96] M. Goldberg, S. Salaris, P. Swoboda, World Wide Web - Course Tool: an environment for building WWW-based courses. In: Proceedings of the Fifth International World Wide Web Conference. Paris, France, 1996.
http://www5conf.inria.fr/fich_html/papers/P29/Overview.html
- [Gopalan99] K. Gopalan, T. Chiue, Integrated Real-Time Resource Scheduling, Technical Report TR-56, Experimental Computer System Labs, Computer Science Department, SUNY at Stony Brook, June 1999.
- [Greenhalgh96] Chris Greenhalgh, Spatial Scope and Multicast in Large Virtual Environments, Technical Report NOTTCS-TR-96-7, Department of Computer Science, The University of Nottingham, September 1996.
- [Handley93] M. J. Handley, Using the UCL H.261 codec controller, December 1993.
http://www.cs.ucl.ac.uk/mice/codec_manual/doc.html
- [Handley95] M. Handley, SDR : Session Directory Tool, University College of London, November 1995.
- [Handley96] M. Handley, J.Crowcroft, C.Borman, J. Ott, The Internet Multimedia Conferencing Architecture, ConneXions - The Interoperability Report, June 1996.
- [Handley98] M. Handley, V. Jacobson, SDP : Session Description Protocol, RFC 2327, IETF, April 1998
- [Hanna98] Donald Hanna, Higher Education in a Era of Difgital Competition, Emerging Organizational Models. JALN Volume 2, issue 1, March 1998.
- [Hardman95] V. Hardman, M. A. Sasse & I. Kouvelas, Successful Multi-party Audio Communication over the Internet, Communications of the ACM, 1995.
- [Hein97] Mathias Hein, David Griffiths, Switching Technology in the Local Network - From LAN to Switched LAN to Virtual LAN, Thomson Computer Press, 1997.
- [Helbig97] T. Helbig, S. Tretter, D. Trossen, Combining CORBA and ITU-T120 to an Efficient Conference Service, IDMS97, Armstad, Germany, September 1997.
- [Hoffman96] D. Hoffman, V. Goyal, RTP Payload Format for MPEG1/MPEG2 Video, RFC 2038, October 1996.

- [Hollier97] M. Hollier, R. Voelcker, Objective performance assessment: Video quality as an influence on audio perception”, 103 rd Convention of the Audio Engineering Society, New York, September 1997, Preprint No. 4590.
- [Huitema95] Christian Huitema, Routing in the Internet, Prentice Hall, 1995.
- [Huston00] G. Huston, Next Steps for the IP QoS Architecture, RFC 2990, November 2000.
- [Hutchison94] Quality Service Management in Distributed Systems, D. Hutchison, G. Coulson, A Campbell, G. S. Blair, Computing Department, Lancaster University, U.K., 1994.
- [IEEE] IEEE/IEE Electronic Library Online
<http://www.ieee.org/ielonline>
- [Isso/IEC92] Isso/IEC JTC 1/SC 29/N 071. Coding of moving pictures and associated audio - for digital storage media at upto about 1.5 Mbits/s - Part1: Systems, Part2: Video, 1992 CD11172
- [ITUT93] Video Codec for audiovisual services at p*64kb/s, 1993. ITUT Recommendation H.261.
- [Jacobson94] V. Jacobson, SIGCOMM '94 Tutorial : Multimedia Conferencing on the Internet, August, 1994.
- [JMF API99] Java™ Media Framework API Guide, Sun Microsystems, JMF 2.0 FCS, November 19, 1999.
- [Joberto99] Joberto Martins, "Redes Corporativas MultiServiço - Caracterização das Aplicações e Parâmetros Básicos de Operação", 1999.
<http://www.jsmnet.com/slides/AnaliseRequisitos/index.htm>
- [Reed98] J. Reed - "Applications Design Team/Wired Learning URL"
<http://www.kn.pacbell.com/vidconf>
- [Josifovski94] Lj. Josifovski, S. Gievska, D. Davcev, Frame Rate Control in a Multimedia Distance Learning System, Faculty of Electrical Engineering , " St. Kiril&Metodji" University, Skopje, Macedonia, 1994.
- [Jun99] Kyungkoo Jun, Ladislau Boloni, David Yau, Dan C. Marinescu, Intelligent QoS Support for an Adaptative Video Service, Computer Science Department, Purdue University, October 1999.

-
- [Kassler00] A. Kassler et al., An Open Endsystem Architecture For Adaptable Multimedia Services With Qos Support, IST Mobile Summit 2000, Galway, Ireland, October 2000.
- [Kent98] Stephen Kent, Randall Atkinson, IP Authentication Header, RFC 2402, November 1998.
- [Kruus98] Peter S. Kruus, Joseph P. Macker, Techniques and issues in multicast security, Naval Research Laboratory, Washington, DC 20375, 1998.
- [Kumar95] V. Kumar, Mbone: Interactive Multimedia on the Internet, New Riders Publishing, 1995.
- [Kurose93] J. F. Kurose, Open Issues and Challenges in Providing Quality of Service Guarantees in High Speed Networks, ACM Computer Communications Review, Vol.32, No 1, January 1993.
- [Laudon95] K. C. Laudon, Ethical Concepts and Information Technology, Commun. Of the ACM vol. 38, December, 1995.
- [Lawhead97] P. Lawhead, The Web and distance learning: what is appropriate and what is not, ITiCSE'97 Working Group Reports and Supplemental Proc., 27-37, 1997.
- [Leland91] W. E. Leland and D. V. Wilson, High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection. In Proc. of IEEE INFOCOM'91, pages 1360-1366, Bal Harbour, FL, USA, April 1991.
- [Levine96] B.N. Levine, J.J. Garcia-Luna-Aceves, A Comparison of Known Classes of Reliable Multicast Protocols, Proceedings of the International Conference on Network Protocols (ICNP)'96, October 1996.
- [Levy96] P. Levy, S. Fowell, Internet-based professional development for networked learner support, Proceedings of the 5th European Distance Education Network (EDEN), France, 8-10 July 1996.
- [Lévy96] Pierre Lévy, Colective Intelligence: Mankind's Emerging World in Cyberspace, Paris, 1996.
- [Li98] Baochun Li, D. Xu, K. Naharstedt, J. Liu, End-to-End QoS Support for Adaptive Applications Over the Internet, Department of Computer Science, University of Illinois at Urbana-Champaign, 1998.

- [Li00] Baochun Li, Agilos: A middleware control architecture for application-aware quality of service adaptations, PhD thesis, University of Illinois at Urbana-Champaign, 2000.
- [Liu98] C. Liu “Multimedia over IP: RSVP, RTP, RTCP, RTSP”. 1998.
http://www.cis.ohio-state.edu/~jain/cis788-97/ip_multimedia/index.htm
- [Mahdavi97] Janish Mahdavi, Sally Floyd, The TCP friendly web site, Technical note sent to the end2end interest mailing list, January 8 , 1997.
- [Maly94] K. Maly, H. Adbel-Wahab, C.M. Overstreet, Ajay Gupta, Muthu Kumar and Rahul Srivastava, Issues in Scalling Multimedia Collaboration Tools for Remote Instruction, Computer Science Department, Old Dominion University, November, 1994.
- [Mankin97] A. Mankin, F. Baker, R. Braden, S. Bradner, M. O’Dell, A. Romanow, A. Weinrib, L. Zhang, Resouce ReSerVation Protocol (RSVP) - Version 1 Applicability Statement, Some Guidelines on Deployment, RFC 2208, September 1997.
- [Martinez00] Margaret Martinez, C.Victor Bunderson, Foundations for Personalized Web Learning Environments, ALN Magazine Volume 4, Issue 2, December 2000.
- [Martins99] Joberto Martins , Qualidade de Serviço (QoS) em Redes IP Princípios Básicos, Parâmetros e Mecanismos, JSMNet - Itelcon - UNIFACS Grupo de Redes de Computadores, Setembro 1999.
- [Mattelart96] Armand Mattelart, The Invention of Communication, University of Minnesota Press M Books, 1996.
- [Maufer97] Thomas Maufer, Deploying IP Multicast in the Enterprise, Prentice Hall, 1997.
- [McCabe98] James D. McCabe, Practical Computer Network Analysis and Design, Morgan Kaufmann Series in Networking, 1998
- [McCanne95] Steve McCanne, Van Jacobson, VIC : A Flexible Framework for Packet Video, ACM Multimedia, November 1995.
- [McCanne97] Steve McCanne, Martin Vetterli, IEEE, and Van Jacobson, Low-Complexity Video Coding for Receiver-Driven Layered Multicast, IEEE Journal on Selected Areas in Communications, vol. 15, no. 6, August 1997.

- [Mendes98] Paulo Mendes, Edmundo Monteiro, Nuno Guimarães, Interface de Controlo de Qualidade de Serviço para Aplicações de Trabalho Cooperativo, 1ª Conferência sobre Redes de Computadores - Tecnologia e Aplicações, CRC'98, Coimbra, Novembro,1998.
- [Metzler99] B. Metzler, T. Harbaum, R. Wittman, M. Zitterbart, Amnet: Heterogeneous Multicast Services based on Active Networking, Technical University Berlin, 1999.
- [Miller98] C. Kenneth Miller, Multicast Networking and Applications, Addison Wesley, 1998.
- [Miras02] Dimitrios Miras, Network QoS needs of Internet2 applications - A survey, I2-QoS Working Group, Computer Science Department, UCL, UK, May 2002.
- [Mittra97] Suvo Mittra, Iolus: A Framework for Scalable Secure Multicasting, Computer Science Department, Stanford University, in Proceedings of the ACM SIGCOMM'97, September 1997.
- [Moore96] M.G. Moore, G. Kearsley, Distance Education: a Systems View, Belmont (USA), Wadsworth Publishing Company, 1996.
- [Mouly92] Michel Mouly and Marie-Bernadette Pautet. The GSM System for Mobile Communications. Cell fz Sys, 4,rue Elisee Reclus, F-91120 Palaiseau, France, 1992.
- [Moy94] J. Moy, Multicast Extensions to OSPF, RFC 1584, March 1994.
- [Neumann98] P.G. Neumann, Risks of E-Education, Communications of the ACM Vol. 40. No. 10, pp.136, 1998.
- [Nevin94] A.I. Nevin, K.A. Smith & A. Udvari-Solner, "Cooperative Group Learning and Higher Education", In J.A. Thousand, R.A. Villa & A.I. Nevin, Creativity & Collaborative Learning, Maryland: Paul H. Brookes Pub. Co., pp. 115-127, 1994.
- [Partridge98] Craig Partridge, Gigabit Networking, Addison Wesley, 1998.
- [Paxson95] V. Paxson and S. Floyd, Wide area traffic: The failure of Poisson modelling, IEEE/ACM Transactions on Networking 3(3), pages 226-244, 1995.
- [Paxson97] Vern Paxson, End-to-End Routing Behavior in the Internet, IEEE/ACM Transactions on Networking, vol. 5, no. 5, pp. 601-615, 1997.

-
- [Pullen99] M. Pullen, R. Malghan, L. Lavu, G. Duan, J. Ma, H. Nah, *A Simulation Model for IP Multicast with RSVP*, RFC 2490, January 1999.
- [Quinn97] Bob Quinn, *Internet Multicasting, Killer network apps that aren't network killers*, Dr. Dobb's Journal, October 1997.
- [Quinn01] Bob Quinn, Kevin Almeroth, *IP Multicast Applications : Challenges and Solutions*, RFC 3170, September 2001.
- [QuinnShute98] Bob Quinn, Dave Shute, *Windows Sockets Network Programming*, Addison Wesley, 1998.
- [Radoslavov00] P. Radoslavov, D. Estrin, et al., *The Multicast-Address-Set-Claim (MASC) Protocol*, RFC 2909, September 2000.
- [Ramanujan98] R. S. Ramanujan, K.J. Thurber, *An Active Network Based Design of a QoS Adaptative Video Multicast Service*, Architecture Tecnology Corporation, Proceedings of the 1998 World Conference on Systems, Cybernetics, and Informatics (SCI'98), 1998.
- [Reinhardt95] A. Reinhardt, *New Ways to Learn*, Byte, Vol 20, No.3, pp. 55-71, 1995.
- [Richards97] Mapping User Level QoS from a Single Parameter, A. Richards, G. Rogers, M. Antoniadis, V. Witana, CSIRO Telecommunications and Industrial Physics, Australia, University of Technology, Sydney, 1997.
- [Bhatti99] S. Bhatti, G. Knight, *Enable QoS Adaptation Decisions for Internet Applications*, Computer Science Department, University College of London, 1999.
- [Schmidt99] Brian K. Schmidt, *An Architecture for Distributed, Interactive, Multi-Stream, Multi-Participant Audio and Video*, Technical Report N°:CSL-TR-99-781, Stanford University, April 1999.
- [Schulzrinne96] H. Schulzrinne, *RTP Profile for Audio and Video Conference With Minimal Control*, RFC 1890, January 1996.
- [SchulzrinneCasner96] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, RFC 1889, January 1996.
- [Semeria97] C. Semeria, T. Maufer. *Introduction to IP Multicast Routing*, July 1997. Internet Draft, <draft-ietf-mboned-intro-multicast-03.txt>

- [Shenker95] Scott Shenker, Fundamental Design Issues for the Future Internet, Xerox Palo Alto Research Center, 1995.
- [ShenkerI97] S. Shenker, C. Partridge, R. Guerin, Specification of Guaranteed Quality of Service, RFC 2212, Sept 1997
- [ShenkerII97] S. Shenker, J. Wroclawski, General Characterization Parameters for Integrated Service Network Elements, RFC 2215, September 1997.
- [ShenkerIII97] S. Shenker, J. Wroclawski, Network Element Service Specification Template, RFC 2216, September 1997.
- [Skillicorn96] D. B. Skillicorn, Using Distributed Hypermedia for Collaborative Learning in Universities. Computer Journal, Vol. 36, No. 6, pp. 471-482, 1996.
- [Sousa97] P. Sousa, Integração de Aplicações de Tempo Real Tolerantes na Arquitectura WWW, Tese de Mestrado, Universidade do Minho, Setembro 1997.
- [Sreenan00] Cormac J. Sreenan, J. Chen, P. Agrawal, B. Narendran, Delay Reduction Techniques for Playout Buffering, IEEE Transactions on Multimedia, vol 2, n.2, June 2000.
- [Steeple96] C. Steeples, C. Unsworth, M. Bryson, P. Goodyear, S. Fowell, P. Levy, C. Duffy, Technological Support for Teaching and Learning: Computer-Mediated Communications in Higher Education (CMC in HE), Computer & Education, 1996.
- [Stevens94] W. Richard Stevens, TCP/IP Illustrated - The Protocols, Vol. 1, Addison-Wesley, 1994.
- [Tanenbaum96] Andrew S. Tanenbaum, Computer Networks, Third Edition, Prentice Hall Internacional, Inc., 1996.
- [telelearn] TeleLearning - NCE Research Report Series
<http://www.telelearn.ca>
- [Teitelbaum99] Benjamin Teitelbaum, Susan Hares, Larry Dunn, Vishy Narayan, et al, Internet2 QBone Building a Testbed for Differentiated Services, 1999.
- [Tennenhouse97] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden, A Survey of Active Network Research, IEEE Communications Magazine, vol. 35, no. 1, pp. 80-86, Jan. 1997.

- [Thaler02] D. Thaler, D. Estrin, D. Meyer, Border Gateway Multicast Protocol (BGMP) : Protocol Specification, June 2002, Work in Progress,< draft-ietf-bgmp-spec-03.txt>
- [Thomas96] Stephen A. Thomas, "Ipng and the TCP/IP Protocols - Implementing the Next Generation Internet", Wiley Computer Publishing, 1996.
- [ThomasCarswell96] P. Thomas, L. Carswell, J. Emms, M. Petre, B. Poniatowska, B. Price, Distance Education Over the Internet. ACM SIGCSE/SIGCUE Proceedings, SIGCSE/SIGCUE Conference on Introducing Technology into Computer Science Education, Barcelona, ITiSCE 96, 1996.
- [Update99] Update, News and trends for telelearning practitioners and researchers, TL - Network of Centres of Excellence, Spring 1999.
- [Wang01] Zheng Wang, Internet QoS, Architectures and Mechanisms for Quality of Service, Bell labs, Lucent Technologies, MK Publishers, 2001.
- [Watson01] Anna Watson, Assessing the Quality of Audio and Video Components in Desktop Multimedia Conferencing, PhD thesis, Department of Computer Science, UCL, March 2001.
- [White99] S. White, A. Gupta, J. Grudin et al, Evolving Use of a System for Education at a Distance, Microsoft Research, Redmond, April 1999.
- [Wittman98] R. Wittman, M. Zitterbart, AMnet : Active multicasting network. In proceedings of ICC'98, June 1998.
- [Wittman01] R. Wittman, M. Zitterbart, Multicast Communication - Protocols and Application, M. Kaufmann Publishers, 2001.
- [Wroclawski97] J. Wroclawski, The use of RSVP with IETF Integrated Services, RFC 2210, September 1997.
- [Yamamoto00] L. Yamamoto, G. Leduc, An Active Layered Multicast Adaptation Protocol, Proceedings of IWAN'2000, Tokyo, Japan, October 2000.
- [Yoakam98] M. Yoakam, "Distance Learning: An Introduction" http://www.ihets.org/distance_ed/ipse/fdhandbook/dist_lrn.html
- [Yoon00] J. Yoon, A. Bestavros, I. Matta, SomeCast : A Paradigm for Real Time Adaptative Reliable Multicast, Computer Science Department, Boston University, 2000.

Anexo I

Protocolos de Suporte a Aplicações *Multicast*

O IPv4 suporta *multicasting*, usando a classe D, cada endereço de classe D identifica um grupo de *hosts*. Disponibilizam-se 28 bits para identificar grupos, portanto mais de 250 milhões de grupos podem coexistir. Quando um processo envia um pacote para um endereço de classe D, desencadeia-se uma tentativa *best-effort* para entregar esse pacote a todos os membros do grupo endereçado, mas não são averbadas quaisquer garantias. Deverá pois existir uma panóplia de protocolos, que se encarregue de sanar as necessidades que cada aplicação suscita, desde a gestão de grupos à segurança, passando pela fiabilidade, etc.

Protocolos de controlo

Cada endereço classe D identifica um grupo de *hosts*, 28 bits estão disponíveis para identificação de grupos, portanto mais de 250 milhões de grupos podem existir ao mesmo tempo. A gestão destes grupos está a cargo de um protocolo, o IGMP que irá ser descrito nesta secção e que é vagamente análogo ao ICMP [Tanenbaum96].

IGMP - *Internet Group Management Protocol*

A entrega *Multicast* IP é selectiva: apenas estações interessadas são supostas receber tráfego dirigido a um dado grupo. Almejando implementar essas árvores de distribuição selectiva, que apenas atingem os membros do grupo, torna-se básico que os membros devem informar os *routers* onde estão, e que grupo(s) lhe(s) interessa(m) [Maufer97].

Como é sabido, os grupos *multicast* são dinâmicos, i.e., a constituição dos grupos é variável, o *status* de cada grupo deve ser conhecido por quem tem de entregar os pacotes *multicast*, os *routers*.

O IGMP - *Internet Group Management Protocol*, permite às estações agregarem-se e abandonar grupos *multicast*. Enviando um relatório de associação ou parceria (*membership report*) ao *router* de vizinhança imediata, uma estação informa o *router* que deseja formar parte de um grupo *multicast*. Os *routers* transmitem periodicamente mensagens com interrogações de parceria (*membership query*) para determinar quais os "host groups" que têm membros nas suas redes directamente conectadas [Semeria97].

A versão 1 deste protocolo, o IGMPv1, foi especificado no RFC-1112, Apêndice 1.

Essas mensagens de interrogação IGMP (ver **Figura 48**) são endereçadas a todos os *hosts* (224.0.0.1) e possuem um TTL (*Time To Live*) IP de 1 para limitar a sua transmissão apenas na sub-rede onde tiveram origem e não são encaminhadas por qualquer outro *router multicast*. Um *host* responde com um *membership report* para cada grupo ao qual pertence. Para limitar o número *membership reports*, cada estação inicia uma espera de tempo aleatório depois de ter recebido o *membership query*. As estações "auscultam" o meio tomando conhecimento dos relatórios de parceria enviados ao *router*; se um relatório é submetido para o grupo ao qual a estação pertence o seu

tempo de espera expira, e cancela o seu relatório para o grupo. Este mecanismo assegura apenas um *membership report* é gerado por cada grupo. Baseado nas informações das constituições dos grupos fornecidas através do IGMP, os *routers* estão capacitados para determinar que tráfego *multicast* (se houver algum) se deve encaminhar para as redes interligadas [Semeria96].

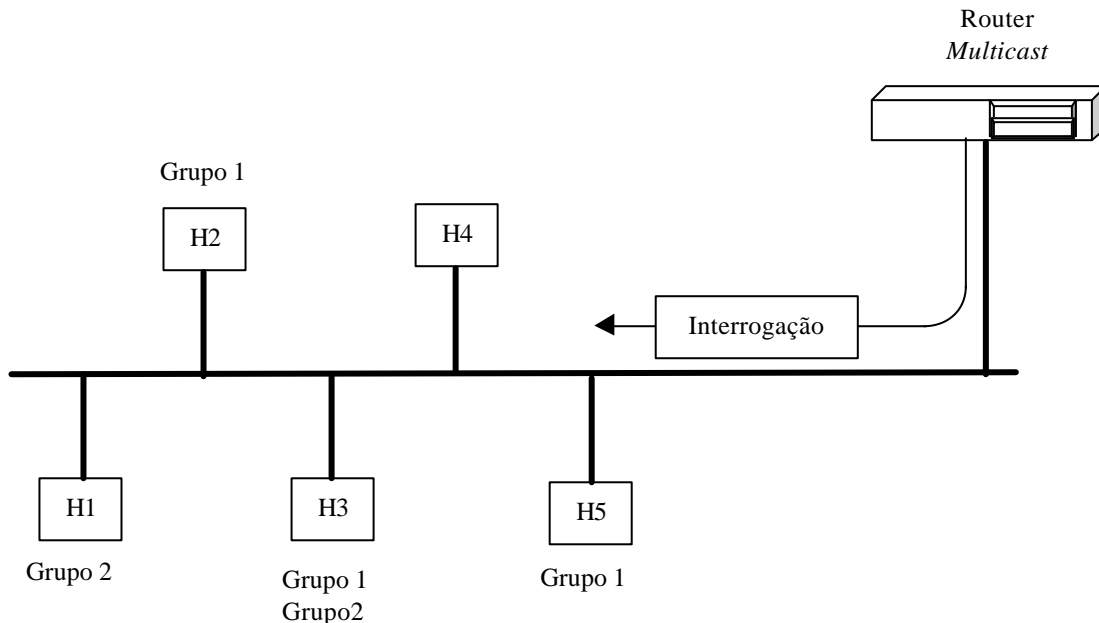


Figura 48 - *Internet Group Management Protocol* - Mensagem de interrogação [Maufer97].

Quando o software aplicativo pede ao software de rede da estação para esta se juntar a um grupo *multicast*, uma mensagem IGMP é enviada ao *router* mais próximo (se o *host* não for já um membro do grupo). Ao mesmo tempo, o endereço *multicast* de classe D do grupo ao qual se junta é mapeado como um endereço de baixo nível e a interface da rede é programada para aceitar pacotes para esse endereço [Crowcroft96]. Por exemplo, se uma estação passa a integrar um grupo num interface *Ethernet*, os 23 bits mais baixos do endereço de classe D são mapeados aos 23 bits mais baixo do endereço *Ethernet*. Resultante desta filtragem de endereços *multicast* por hardware, um *router* não necessita manter uma lista detalhada das estações que pertencem a cada endereço de grupo, mas apenas esse membro, pelo menos, do grupo, está presente na sub-rede à qual se encontra vinculado.

Uma das fraquezas da primeira versão do IGMP era a latência demasiado elevada associada com o término de sessões *multicast*. Depois do último membro de um grupo *multicast* numa sub-rede ter abandonado o grupo, os outros *routers* não são

imediatamente notificados para deter a propagação de tráfego para o grupo. Esta demora era causada pelo IGMP esperando até que várias interrogações indicassem que não restavam membros na sub-rede, de um grupo em particular. No entanto, indesejavelmente, tráfego desnecessário seria encaminhado para a sub-rede. O custo deste envio inútil podia ser elevado, particularmente num segmento da Internet com largura de banda estrangida.

A versão 2 do IGMP [Fenner97], apresenta alguns refinamentos que ajudarão a reduzir o *overhead* do protocolo. As mensagens de interrogação dirigidas a grupos específicos (*Group Specific Query Message*) permitem ao *router* interrogar grupos específicos nas redes onde estão directamente vinculados em vez de serem forçados a interrogar todos os grupos indiscriminadamente. Começando com a versão 2, o término de uma sessão *multicast* já não é feito de forma passiva. O último *host* de uma sub-rede a deixar o grupo *multicast*, transmite uma mensagem de saída de grupo (*Leave Group*) ao *router* na qual é indicado qual o grupo abandonado. Depois de verificar a partida com uma mensagem de interrogação dirigida a esse grupo específico, o *router* notifica outros *routers* para cessarem o encaminhamento de tráfego para a sub-rede dirigido ao grupo.

A versão 3 do IGMP vai mais longe na redução do *overhead*. A largura de banda será conservada pela mensagem *Group-Source Report* que permitirá às estações receber tráfego de fontes específicas de um grupo *multicast*. Em versões prévias do IGMP, o tráfego de todas as fontes tinha de ser encaminhado para uma sub-rede mesmo se as estações estivessem apenas interessadas em receber tráfego de fontes específicas. As mensagens *Leave-Group* apresentadas em primeira instância pela versão 2 foram também aperfeiçoadas para permitir às estações largar um grupo inteiro ou para especificar a fonte a que queriam renunciar [Semeria97].

Atendendo a que as versões recentes do IGMP podem reduzir o tráfego desnecessário, optimizando a utilização deste protocolo, deve ser favorecida a sua utilização em detrimento das anteriores.

Pelos métodos acima mencionados, os *routers multicast* estão habilitados a manter, por interface, uma tabela actualizada contendo os grupos cujo tráfego tem interesse para a sub-rede, pelo que após a recepção de pacotes *multicast*, os *mrollers* sabem para que interfaces os pacotes devem ser encaminhados.

Protocolos de *routing*

Os protocolos de *routing multicast* são utilizados pelos *mrollers* para descobrir caminhos de entrega (árvores), que possibilitam o encaminhamento de datagramas *multicast* através da rede. O encaminhamento *multicast* é feito com base na informação derivada dos protocolos de *routing multicast*. Nesse sentido, os protocolos de *routing* desenvolvem a função de actualizar as tabelas de rotas para a entrega de pacotes entre dois ou mais pontos, no momento em que ocorrem problemas num segmento da ligação, ou um novo *host* deseja associar-se a um grupo *multicast*.

A escolha dos protocolos inseridos nesta resenha foi definida de acordo com a filosofia utilizada para a construção das árvores de *routing*. Existem três técnicas para a construção de árvores: SPT (*Shortest Path Tree*), RPT (*Rendezvous Point Tree*) e uma técnica híbrida, utilizando inicialmente a árvore RPT e podendo migrar para SPT.

A árvore SPT é construída de acordo com o menor caminho entre a origem e os destinos. Por outro lado a árvore RPT é construída tomando como base um *router* central que recebe todas as mensagens e as envia para os destinos de um grupo.

Desta forma, abordar-se-á o *Multicast Extensions Open Shortest Path First* (MOSPF) por utilizar a árvore SPT, o *Core Based Tree* (CBT) por utilizar a árvore RPT e o PIM-SM, *Protocol Independent Multicast - Sparse Mode* por utilizar a técnica híbrida.

Os protocolos de encaminhamento *multicast* podem dividir-se segundo a forma como se processa a integração nos grupos: associação implícita ou associação explícita. O **Quadro 11** edita essa distinção. Uma outra divisão pode ser estabelecida se atendermos ao tipo de árvore de distribuição que é construída.[Maufer97].

Protocolos de Encaminhamento <i>Multicast</i>				
Associação Explícita (<i>Explicit-Join</i>)			Associação Implícita (<i>Implicit-Join</i>)	
MOSPF	CBT	PIM-SM	PIM-DM	DVMRP

Quadro 11 - Classificação de protocolos de encaminhamento *multicast* em função do tipo de associação aos grupos.

Além disso, os protocolos de encaminhamento *multicast*, dividem-se geralmente entre duas soluções: modo denso (*dense mode*) ou disperso (*sparse mode*).

Como o nome sugere, o modo denso lida com grupos *multicast* densamente distribuídos através da rede. Este tipo de protocolos cria árvores de entrega, fazendo uso de uma técnica denominada "*flooding*" ou inundação, para propagar informação ao longo dos *routers* da rede. Em modo disperso considera-se que os membros possam estar dispersos na vasta topologia da Internet.

DVMRP - *Distance-Vector Multicast Routing Protocol*

Existem dois protocolos proeminentes em ambiente densos, são eles: o DVRMP e o MOSPF, *Multicast Open Shortest Path First*. Ambos são bastante próximos dos protocolos funcionalmente similares em *Unicast* e integram as normas IETF. O protocolo DVRMP está descrito no RFC 1075.

O DVMRP, ou *Distance-Vector Multicast Routing Protocol*, é um protocolo baseado em vector de distâncias, tal como o RIP (*Routing Information Protocol*). Os pacotes inundam toda a rede de acordo com uma árvore inversa de menor caminho. A grande vantagem de DVMRP é a contenção de tráfego *multicast* através de um esquema de *prune* (poda), como descrito a seguir.

A árvore de distribuição *multicast* para um par (*fonte, grupo*) é inicialmente determinada através de uma inundação (do primeiro pacote transmitido ao grupo), e abrange toda a rede. As sub-redes folha que não possuem assinantes do grupo são "podadas" da árvore *multicast* enviando uma mensagem PRUNE ao seu "pai" (*upstream*). O processo ocorre recursivamente, e qualquer encaminhador que não possua "filhos" assinantes é igualmente podado. A árvore resultante não possui ramos onde não existam um ou mais *hosts* assinantes, evitando transmissões desnecessárias.

Entretanto, o estado de *pruning* nos *routers* expira periodicamente (*soft state*), ocasionando nova inundação e posterior poda. Outros ramos podem ser recolocados na árvore antes da próxima inundação através de *grafting*.

MOSPF - *Multicast Open Shortest Path First*

Este protocolo de *routing multicast* está descrito no RFC 1584.

Os *routers* implementados com MOSPF possuem uma imagem da topologia da rede, de acordo com as designações do protocolo de *routing Open Shortest Path First* (OSPF - protocolo *unicast*), especialmente construído para distribuir informações da topologia entre *routers* de um mesmo sistema autónomo.

Os *routers* MOSPF fazem uso do protocolo IGMP para monitorar as redes ligadas directamente a eles, permitindo assim, que sejam criadas as tabelas de *routing*, onde são mantidos os membros de um determinado grupo e especificado o *router* local que será responsável pela entrega dos pacotes *multicast* a esses membros.

Para que os encaminhadores possam manter as tabelas de *routing* é escolhido um *router* MOSPF, chamado *Designated Router* (DR), que envia mensagens IGMP à rede, perguntando se existem membros de um determinado par (Grupo, Fonte) e espera por mensagens IGMP de resposta, as quais servem para actualização das tabelas de *routing*.

O DR é responsável ainda, pela transferência via *flooding* das informações existentes nas tabelas aos outros *routers* no domínio OSPF, assegurando com isso, que todos os pacotes originados remotamente possam ser transmitidos a outros membros de um determinado par (Grupo, Fonte). Isso é possível pela criação da árvore SPT entre uma fonte de um grupo e seus demais membros.

Cada árvore SPT é construída a pedido, quando o primeiro pacote atinge um membro do par (Grupo, Fonte) destino. Após a construção da árvore, inicia-se o processo de *prunning* (poda) dos ramos que não são necessários para a transferência dos pacotes. Se algum *router* desejar receber os pacotes após ter sido “podado”, ele deve notificar os seus vizinhos através de um pedido de “reenxerto” [Moy94].

O MOSPF é um protocolo baseado no “estado de *link*” (*link-state*): cada *router* possui uma base de dados descrevendo o estado de todos os *links* da rede. Quando um *router* detecta a mudança no estado de um *link*, faz um *broadcast* de uma mensagem de actualização para todos os demais *routers*, fazendo com que as réplicas da base de *links* sejam actualizadas de acordo. O MOSPF apenas adiciona um novo tipo de anúncio, para identificar a localização de membros de grupos. Cada encaminhador pode então usar o mesmo algoritmo para computar, de forma local e independente, a árvore de menor caminho.

Considerando que a rede está dividida em sistemas autónomos, e sistemas autónomos em domínios, o MOSPF permite *multicast* intra-domínio, inter-domínio, e entre sistemas autónomos.

Os protocolos DVMRP e MOSPF têm uma vantagem comum: aproveitam a infra-estrutura de *routing unicast* existente. Por outro lado, essa vantagem pode ser encarada como uma desvantagem: são dependentes de protocolos de *routing unicast* específicos.

CBT - Core Based Tree

O protocolo CBT (RFCs: 2189 e 2201) foi concebido com a intenção principal de aumentar a escalabilidade apresentada nos protocolos que constróem árvores de *routing* para cada fonte *multicast*. A abordagem consiste em implementar uma única árvore de *routing* para cada grupo, que será utilizada por todas as fontes do grupo.

As fontes enviam os seus dados para o *router* central (também chamado de *Rendezvous Point* - RP), como se estivessem enviando pacotes *unicast*. Quando estes pacotes chegam ao *router* central, são disseminados aos demais encaminhadores pertencentes ao grupo, que por sua vez usam o identificador de grupo (endereço *multicast* IP do grupo), como um índice para encontrarem nas tabelas de *routing* as interfaces a quem devem entregar o pacote.

O CBT constrói e mantém uma árvore de distribuição *multicast* somente para alcançar redes que possuem membros associados a um grupo. Para alcançar este objectivo, os *hosts* expressam o seu interesse em juntar-se a um grupo, através de respostas a perguntas IGMP feitas pelo DR.

O *router* ao receber as respostas IGMP dos *hosts* ligados directamente a ele, gera uma mensagem requisitando aos seus vizinhos o estabelecimento de um caminho para a entrega das mensagem *multicast* vindas do RP. Os seus vizinhos (*routers*) actualizam as suas tabelas de *routing* e, devolvem uma mensagem confirmando o estabelecimento do caminho entre o RP e o *router* requisitante.

Estas etapas servem para definir a árvore de *routing multicast* relacionada ao grupo especificado e ajustar as tabelas de *routing multicast*, definindo o grupo, a interface de chegada e as interfaces de saída, conhecidas no CBT como "pai" e "filhos" respectivamente. As notificações de *prunning* (poda) são definidas pelos *routers*

"filhos", indicando que não possuem interesse em receber pacotes de um determinado grupo [BallardieI97, BallardieII97].

PIM - *Protocol Independent Multicast*

O grupo de trabalho da IETF, *InterDomain Multicast Routing* (IDMR), é responsável pelo desenvolvimento do PIM, *Protocol Independent Multicast*, como o nome sugere este protocolo terá bivalência para as duas modalidades, modo denso ou disperso. Assim, será possível optar por PIM-SM (*Sparse Mode*) ou PIM-DM (*Dense Mode*). Estes dois protocolos serão tratados com maior detalhe, em sucessivas subsecções próprias.

No que a este trabalho concerne, o *router* da nossa rede (Cisco 4000) está configurado para usar o protocolo PIM-SM, por razões que se prendem com a realidade actual na utilização desta tecnologia a nível local e nacional, da escassa largura de banda disponível, bem como pelos propósitos de investigação que estão subjacentes.

PIM-SM - *Protocol Independent Multicast - Sparse Mode*

Os problemas de escalabilidade inerentes ao Mbone, extrapolando para uma eventual ubiquidade do *multicast*, podem ser obviados recorrendo a protocolos, que tal como o PIM-SM, evitam que os *routers* que não explicitem o seu interesse na informação *multicast* de determinado grupo, sejam poupados.

O PIM-SM usa a abordagem do CBT²⁰, no sentido de definir para cada grupo o conceito de RP [Semeria97], uma espécie de "ponto de encontro" central, onde os receptores se encontram com novas fontes. O precursor de um grupo designa um RP primário e uma pequena lista ordenada de RPs alternativos. No caso de existirem vários encaminhadores PIM na mesma LAN, o *router* com IP mais elevado é escolhido como DR (*Designated Router*), este é responsável por enviar mensagens *join/prune* via RP em consequência do comportamento dos interligados. Cabe-lhe adicionalmente, determinar univocamente o RP para o grupo. O RP actua como um servidor que controla a

²⁰ *Core Based Trees*

composição do grupo, que recebe todos os pacotes endereçados ao grupo e os distribui de acordo com uma árvore *multicast* tradicional.

Os *hosts* que desejam fazer parte de um grupo, mandam mensagens explícitas de associação aos *routers* que estão no caminho do RP escolhido. Para isso, cria uma entrada (*,G), por onde serão enviados os pacotes. Quando não existirem mais membros locais conectados ao grupo, o DR obtém essa informação através do IGMP. Se não houver *downstream*, i.e., elementos que necessitem de pacotes do par (Grupo, Fonte), a entrada (*,G) é removida. Quando um *host* inicia a transmissão de pacotes de dados *multicast* para um grupo, inicialmente, o seu DR entrega cada pacote ao RP, encapsulando-os em pacotes *unicast*. O RP desencapsula os pacotes e envia-os para os membros do grupo, pela árvore RP-Tree.

Para aumentar a eficiência do protocolo, quando uma fonte está a transmitir um número significativo de pacotes de dados, os encaminhadores com receptores locais podem juntar-se a esta fonte específica, pela árvore SPT, criando a entrada (S,G) (*fonte, grupo*).

A política recomenda, inicialmente, que seja feita a troca para a árvore SPT, após receber um número significativo de pacotes de dados, de uma fonte particular, durante um intervalo específico de tempo. Para realizar essa política, o *router* monitoriza os pacotes de dados gerados pela fonte, obtendo a taxa de dados produzida, otimizando a tomada da decisão sobre que árvore de distribuição será usada.

Para que um novo membro possa associar-se a um determinado grupo, no qual o estado de *prune* tenha sido estabelecido, é necessário que o estado actual seja erradicado, possibilitando que os pacotes cheguem até aos novos membros [Deering97]. Por questão de eficiência, o PIM/SM não funciona necessariamente com um esquema de árvore partilhada. Quando uma fonte gera tráfego superior a um limite pré-definido (*threshold*), o PIM/SM no *host* pára de encaminhar pacotes ao *Rendezvous-Point*, adoptando ao invés, uma árvore de menor caminho (SPT) baseada no emissor (ou seja, nele mesmo).

A **Figura 49** mostra como se processa a transmissão *multicast* pelo protocolo PIM-SM.

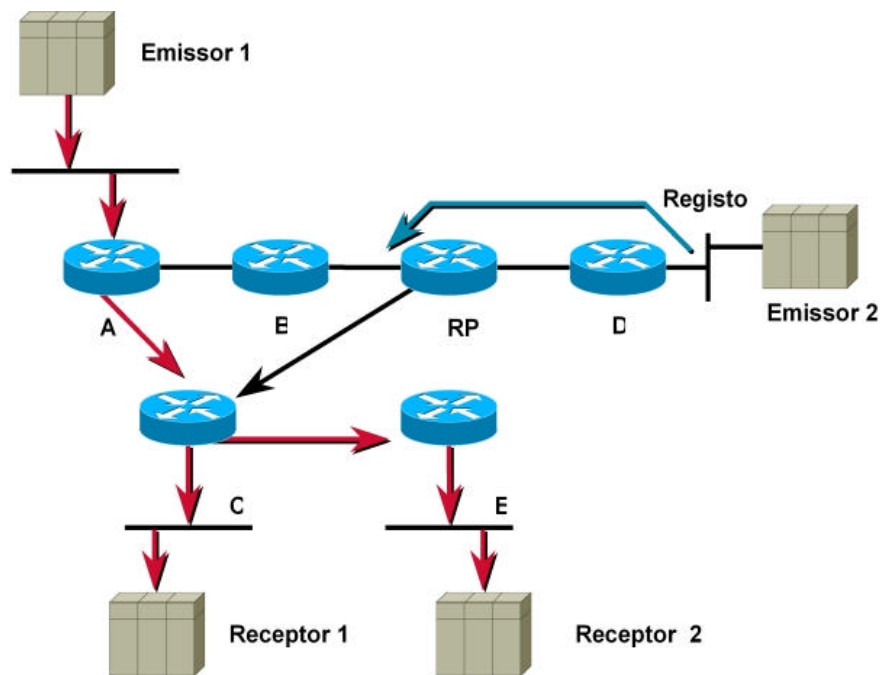


Figura 49 - Exemplo dos mecanismos de controlo de comunicações de grupo, no protocolo PIM-SM. O emissor 2 pretende transmitir para o grupo através da árvore partilhada.

PIM-DM - *Protocol Independent Multicast - Dense Mode*

O PIM-DM destina-se a redes com alta densidade de membros. Neste tipo de configuração, faz sentido inundar a rede com pacotes e limitar o tráfego através do esquema de *prune* e *graft*. A diferença para o DVMRP é que PIM/DM não depende de um esquema único de *routing unicast*, operando com RIP e OSPF, entre outros.

O protocolo PIM/DM não é adequado para redes com membros dispersos, pois as inundações geram desperdício de largura de banda, como tal, tem importância diminuída no âmbito deste trabalho.

BGMP - *Border Gateway Multicast Protocol*

Este protocolo destina-se a *routing multicast* inter-domínio. O BGMP [Thaler02] constrói árvores partilhadas para grupos *multicast* activos, e permite aos domínios receptores construir ramos de distribuição inter-domínio, específicos para cada fonte, quando necessários. Implementado sobre conceitos derivados dos protocolos CBT e

PIM-SM, o BGMP requer que cada grupo *multicast* seja associado com uma única raiz (no contexto BGMP é referida como sendo a raiz do domínio). Contudo, no BGMP, a raiz é um domínio inteiro, em vez de um único *router*.

O protocolo BGMP assume que em qualquer instante, diferentes gamas do espaço de endereçamento da classe D estão associados (e.g., com MASC [Radoslavov00]) com vários domínios. Cada um desses domínios torna-se então a raiz das árvores partilhadas dos domínios para todos os grupos na sua gama. Os participantes *multicast* irão receber, geralmente, um melhor serviço *multicast* se o alocador do endereço do iniciador da sessão seleccionar endereços do espaço parcial do seu próprio domínio, causando portanto que a raiz do domínio seja local para pelo menos um dos participantes da sessão.

Quando um receptor se junta ao um endereço de grupo *multicast* específico, o *router* da fronteira através da raiz do domínio gera uma mensagem de *Join* específica para o grupo, que é então encaminhada de *router* de fronteira em *router* de fronteira em direcção à raiz do domínio. As mensagens de *Join* e *Prune* são enviadas via conexões TCP entre pontos BGMP, e o estado do protocolo BGMP é actualizado por mensagens *KEEPALIVE* enviadas periodicamente sobre o TCP.

Como já se referiu, o BGMP usa o TCP como protocolo de transporte, assim elimina a necessidade de implementar fragmentação de mensagens, retransmissão, confirmação e sequenciamento. Utiliza o porto 264 para estabelecimento das suas conexões. Este porto é distinto do porto BGP para fornecer independência ao protocolo e, distinguir facilmente os pacotes dos dois protocolos (e.g., em classificadores de pacotes, utilitários de diagnóstico, etc.).

Protocolos de transporte

Transporte de tráfego multimédia

As aplicações multimédia necessitam de mecanismos eficientes de transporte para a informação dos vários media, que como se sabe é dependente do tempo. O protocolo de transporte em tempo-real, RTP (*Real-Time Transport Protocol*) [SchulzrinneCasner96], foi concebido para responder a este desiderato. Não assegura a

entrega dos dados atempadamente, relegando para as camadas de serviços inferiores a consecução dessas garantias de QoS. Inversamente, o RTP fornece mecanismos para incluir estampilhas temporais, identificando diferentes tipos de carga, numeração do sequenciamento, e monitorização da entrega da informação. Isto é tipicamente implementado no topo do UDP, mas outros protocolos básicos de comunicações podem ser igualmente usados.

O RTP define primariamente a sintaxe da transmissão da informação, são usados perfis para descrever as semânticas, i.e. como interpretar os campos dos cabeçalhos dos pacotes. Consequentemente, os serviços RTP não estão completamente especificados, possibilitando que cada aplicação individualizada possa ajustar o protocolo para cumprir os seus requisitos. Por exemplo o perfil áudio e vídeo [Schulzrinne96] lista vários valores para o campo tipo de carga, que indica diferentes esquemas de codificação, preconiza o uso de estampilhas temporais e números de sequência, especifica a inserção apropriada de marcadores para indicar pontos significativos contidos numa *stream*, etc.

É também especificado um protocolo de controlo, o RTCP (*Real Time Control Protocol*), este é usado pelas aplicações para monitorizar a qualidade de serviço que a rede está a conseguir entregar, bem como para disseminar informação que concerne aos participante numa sessão de comunicação. Em concordância com o anteriormente exposto, o RTP é um protocolo de utilidade reconhecida para aplicações que processam a transmissão de informação multimédia num domínio específico e independentemente do formato, acrescentando o facto de poder monitorizar e reagir à alteração das condições da rede.

O protocolo RTP - *Real Time Protocol*

O protocolo RTP foi consensualmente proposto como protocolo de transporte de suporte às aplicações de tempo-real. O RTP não efectua nenhuma reserva de recursos, estando isso, caso se entenda, a cargo das entidades protocolares inferiores.

Descrição do protocolo

De uma forma muito geral poder-se-ão salientar os seguintes pontos da especificação do protocolo RTP:

- fornece mecanismos de transporte fim-a-fim adequados para aplicações que operem na transmissão de dados em tempo-real;
- independente do protocolo de transporte usado (caso este seja implementado sobre outro protocolo de transporte tradicional) e das camadas de rede que o suportam. Na realidade deve ser usado um protocolo de transporte que implique um mínimo de controlo adicional e um mínimo de ocupação de largura de banda, como é o caso do UDP;
- no que diz respeito à camada de rede, poder-se-á dizer que está vocacionado para cenários de operação sem a necessidade de reserva de recursos por parte dos elementos da rede. No entanto, e caso esse serviço seja fornecido (por exemplo pelo RSVP), o RTP continuará a desempenhar o papel de protocolo de transporte embora sem a vertente adaptativa;
- proporciona mecanismos de comunicação em ambientes *multicast* ou *unicast*;
- engloba um protocolo de controlo adicional (RTCP), que permite controlar o estado das conexões. Neste canal transitarão todo um conjunto de informações de estado de cada uma das entidades participantes.

Na especificação da utilização do protocolo RTP ficou definido que na transmissão simultânea de vários media, cada um deles deveria ser transmitido num canal RTP independente. Entre outras vantagens, uma delas poderá ser a flexibilidade introduzida no serviço, permitindo, por exemplo, que um dos participantes opte unicamente pela recepção de um único *medium*. Assim, e se tomarmos como exemplo a comunicação baseada em som e imagem entre dois indivíduos, existirão quatro canais RTP independentes, dois para cada participante, correspondendo cada um deles ao transporte de som e imagem, como se pode observar na **Figura 50**.

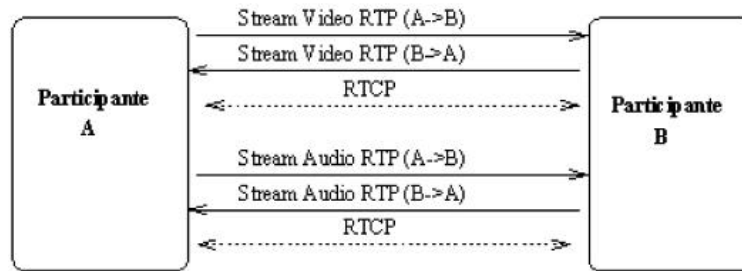


Figura 50 - Comunicação entre duas entidades usando o protocolo RTP [Sousa97].

Como já tinha sido referido, o RTP está vocacionado para a comunicação em ambientes multi-participante, suportados por mecanismos de *multicast*. Assim sendo um grupo de utilizadores poder-se-á juntar a uma dada sessão RTP e nela participar activamente desempenhando o papel de mero receptor quer também de transmissor de informação. O RTP foi desenhado com o intuito de operar sobre mecanismos de *multicast* de uma forma a minimizar o tráfego da sessão, muito em especial no que se refere à informação de controlo, como adiante se explicará quando se abordarem algumas questões relacionadas com o canal RTCP.

Na **Figura 51** está esquematizada uma sessão RTP onde participam quatro entidades, das quais três são simultaneamente emissoras/receptoras de tráfego RTP, enquanto uma quarta desempenha unicamente o papel de emissora. Desta forma todos os receptores²¹ de tráfego RTP terão obrigatoriamente que relatar a qualidade com que estão a receber as tramas das várias entidades emissoras, para todos os participantes da sessão. Através deste mecanismo as entidades geradoras tomam não só consciência de como o seu tráfego está a chegar aos diversos destinos, mas também como o tráfego gerado por outras entidades está a chegar a esses mesmos destinos. Deste modo, todo o grupo tem a percepção do estado de todos os participantes envolvidos na sessão.

Num ambiente multi-participante uma ligação lógica RTP é identificada por um endereço de rede (correspondente a um dado grupo *multicast*), e por um par de portas protocolares, que multiplexam os canais RTP e RTCP.

²¹ entende-se por entidades receptoras aplicações que se filiaram num dados grupo *multicast* e estão aptas a receber pacotes de rede que lhe são dirigidos.

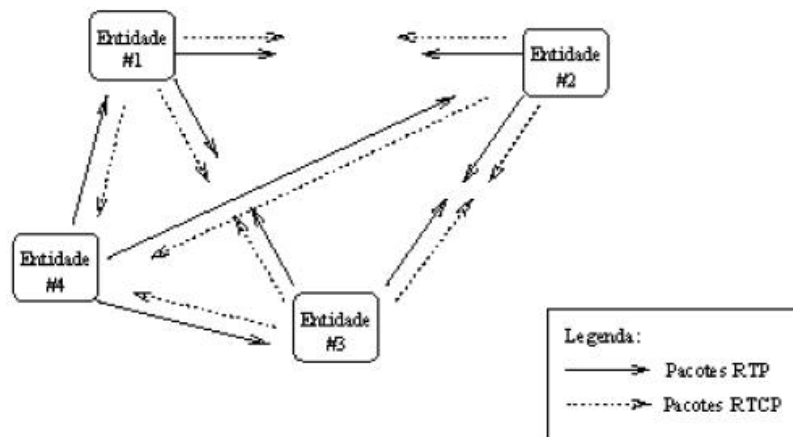


Figura 51 - Utilização do RTP em comunicações *multicast* [Sousa97].

Formato de um pacote RTP

Um pacote RTP está dividido em cabeçalho e na parte dos dados. No cabeçalho figura informação sobre o tipo de dados transportados (mais propriamente o tipo do *medium* e sua codificação²²) bem como informação útil para funcionalidades inerentes aos próprios mecanismos do RTP. Na **Figura 52** é apresentado o formato de um pacote RTP, sendo de seguida dadas algumas indicações acerca da utilidade desses campos.

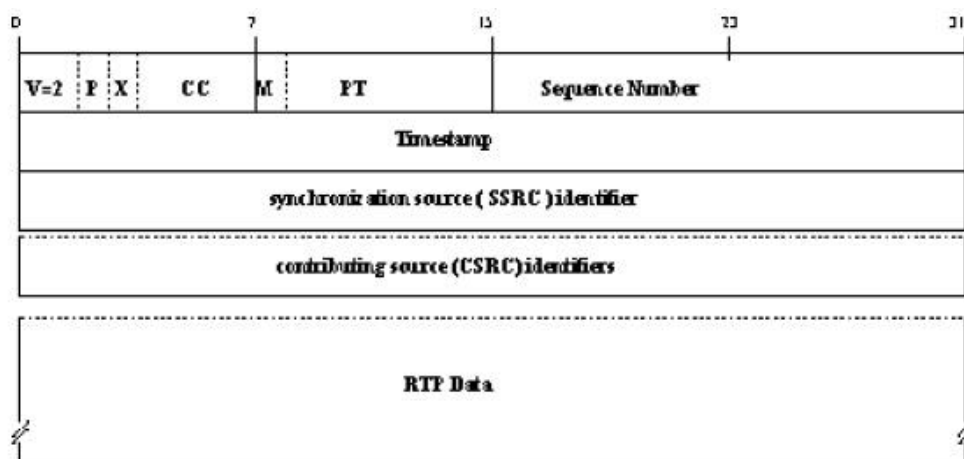


Figura 52 - Formato de um pacote RTP [Sousa97]

²² referidos como *payload type* no âmbito do protocolo RTP.

- *Version (V)* : Este campo define a versão do protocolo RTP que está a ser usado.
- *Padding (P)* : Bit que quando activado indica que o pacote contém informações adicionais que não fazem parte da codificação do *medium* transmitido.
- *Extension (X)* : Bit que quando activado indica que existe um cabeçalho adicional para ser interpretado mediante o tipo de conteúdo transportado no pacote (identificado pelo *payload type*).
- *CSRC count (CC)* : Indica o número de fontes de contribuição que operaram sobre este canal RTP. Estes indicadores são usados aquando da participação dos denominados *mixers*²³ que são sistemas intermediários que combinam e tratam as *streams* RTP originadas por outros sistemas.
- *Marker (M)* : Interpretado consoante o *medium* transportado e a sua codificação. Por exemplo, no caso de transporte de vídeo, poderá significar qualquer coisa como o fim de uma imagem que faz parte da sequência transmitida.
- *Payload Type (PT)* : Identificador único que terá correspondência para um dado *payload format*, que definirá como a aplicação deverá tratar a informação recebida. No RTP a um dado *payload type* está associado um *payload format* que identifica como um determinado *medium* se encontra encapsulado no pacote RTP e como deve ser interpretado pela aplicação. Já se encontram especificados alguns formatos em [Berc96, Hoffman96, Schulzrinne96].
- *Sequence Number* : Identificador do número de pacote RTP. Este número sofrerá incrementos de uma unidade por parte do emissor sempre que for emitido um novo pacote RTP. O número de sequência dos pacotes permite o cálculo do número de pacotes esperados e recebidos por parte das entidades receptoras bem como uma estimativa das perdas verificadas.

²³ entidades intermediárias com capacidade de alterarem o conteúdo dos pacotes RTP. Definidos no RTP para funcionarem como conversores de formato.

- *Timestamp* : Estampilha temporal que reflecte o instante em que o primeiro octeto de informação transportado no pacote RTP foi processado pela entidade geradora. Esta estampilha deverá ser calculada de uma forma monótona e linear de modo a que seja possível ao receptor efectuar a sincronização do *medium* em questão e o cálculo do *jitter* que está a ocorrer.
- *SSRC* : Identificador da fonte de sincronização. Identifica univocamente a fonte geradora de tramas RTP. Este identificador deverá ser único sendo pois necessária a utilização de um algoritmo aleatório de geração de *SSRC*. O identificador de sincronização deverá possibilitar ao receptor de uma sessão RTP agrupar as tramas de um dado emissor para posterior processamento, sabendo que nesse domínio as estampilhas temporais e os números de sequência estão relacionados entre si.
- *CSRC* : Identificador da fonte contribuinte.

Multiplexagem de sessões RTP

O protocolo RTP utiliza como estratégia de multiplexagem o endereço de transporte, ou seja, um endereço de rede mais uma porta de destino definem uma única sessão RTP. Nessa sessão irão transitar pacotes RTP de um único tipo (definido pelo *payload type*), e gerados por uma ou mais entidades (identificadas pelo *SSRC*). Caso as entidades troquem entre si outros tipos de dados, uma nova sessão RTP, independente da primeira, deverá ser utilizada. Vejamos algumas justificações para a adopção desta estratégia, e não optar, por exemplo, por multiplexar através do uso de diferentes *payload types* para um mesmo *SSRC*:

- Se um dado método de codificação²⁴ fosse alterado numa dada sessão RTP ficaríamos sem saber qual deles é que realmente se alterou.
- A um identificador *SSRC* estará associado um dado domínio tanto ao nível de números de sequência como ao nível de estampilhas temporais. Multiplexando, numa mesma sessão RTP, vários media para um mesmo *SSRC* originaria a necessidade de termos domínios de estampilhas bem como diferentes números de

sequenciação que permitissem calcular o número de pacotes perdidos para cada um media transportados na mesma sessão RTP.

- Os pacotes RTCP transportam informação de controlo referente a cada uma das fontes de informação (designadas pelo SSRC) e não fazem qualquer referência aos respectivos conteúdos transportados nos pacotes RTP.
- Multiplexando vários media numa mesma sessão RTP compromete o uso de diferentes estratégias de encaminhamento para cada um dos media, consoante as suas necessidades. Outra estratégia que esta solução inviabilizaria, seria a recepção de um dado subconjunto dos media transmitidos. Este tipo de solução pode ser usada caso a entidade receptora não seja capaz de suportar²⁵ a totalidade dos media transmitidos por uma aplicação operando sobre o RTP.

RTCP - Real Time Control Protocol

Uma das características do RTP é a definição de um protocolo de controlo a ele associado. Algumas das principais características do RTCP são de seguida apresentadas. O RTCP é identificado pelo mesmo endereço de sessão RTP e pela porta de transporte da sessão RTP+1. No canal RTCP serão trocadas mensagens de controlo entre os diversos participantes na sessão RTP associada. Essas mensagens poderão ser de identificação das fontes (como o nome dos utilizadores participantes, endereços e-mail, etc.) ou então mensagens do estado da sessão. Estas últimas são geradas por todos os participantes da sessão e tem como objectivo principal o de relatar os valores dos diversos parâmetros de funcionamento (perdas de pacotes, variação nos atrasos, estado de sincronização, etc.) verificados por cada uma das entidades.

Existem dois tipos de relatórios que são gerados pelas entidades: *SR (Sender Reports)* : gerados pelas entidades que são simultaneamente receptores e emissores RTP; e *RR (Receiver Reports)* : gerados por entidades que são unicamente receptores RTP. Em ambos os casos, esses relatórios incluem informação de estado referente a todos os emissores RTP cujas mensagens estão a ser processadas pela entidade que gera

²⁴ o *payload type* identifica o tipo de *medium* transportado bem como a sua forma de codificação.

esses mesmos relatórios. Este é o mecanismo que permite ao emissor ter conhecimento do desenrolar da aplicação na perspectiva dos receptores de informação, permitindo a estes obterem informações úteis para efectuarem alguns ajustes ao seu funcionamento.

Formato de um pacote RTCP

Analisemos pois o caso concreto de um pacote do tipo SR. Na **Figura 53** são apresentados os seus componentes e é feita uma breve descrição do seu significado.

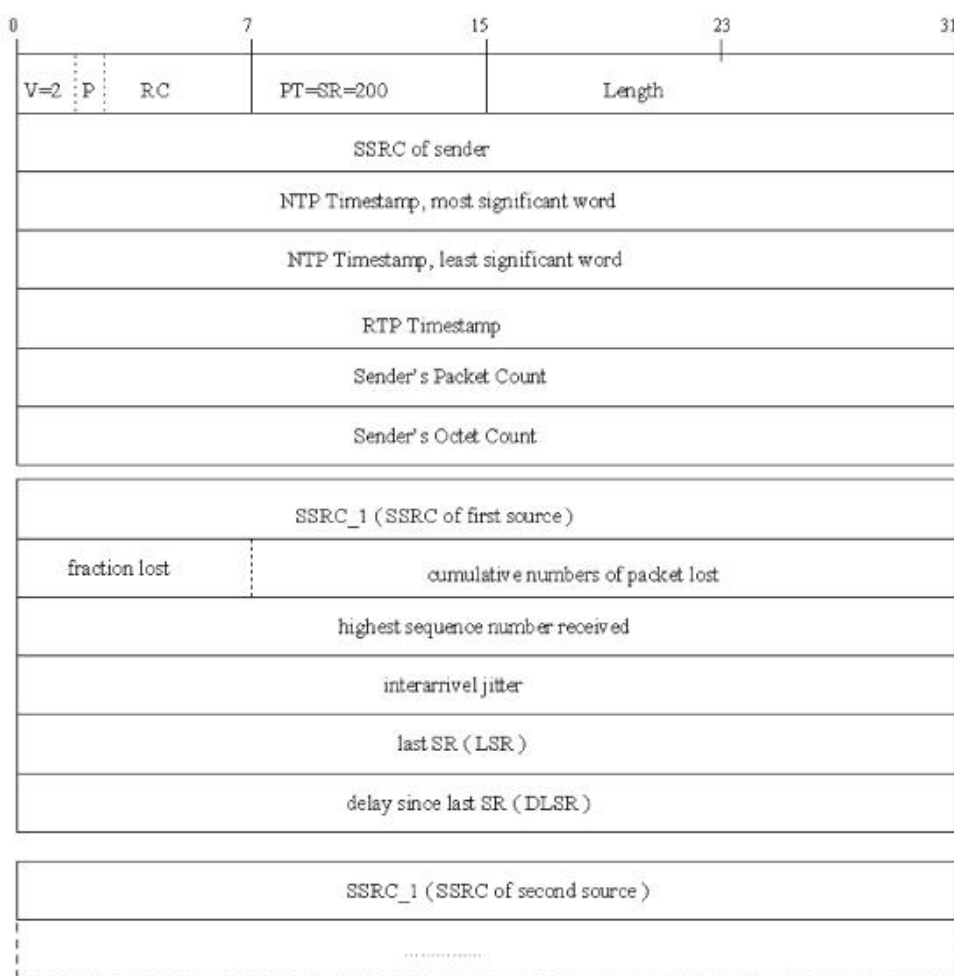


Figura 53 - Formato de um pacote RTCP [Sousa97].

²⁵ quer por falta de recursos do meio de comunicação, quer por falta de capacidade de processamento da entidade.

Cabeçalho

- *Version* : Versão do protocolo.
- *Padding* : Quando activado indica se este pacote contém octetos adicionais situados no fim do pacote RTCP.
- *RC (Reception Report Count)* : Número de blocos presentes neste pacote que são estatísticas de recepção para fontes de informação RTP.
- *Length* : Tamanho do pacote RTCP em número de palavras de 32 bits menos um.
- *SSRC* : identificador da entidade que gerou este relatório.

Informação da própria entidade emissora

Estas informações estão relacionadas com a própria entidade que gerou este relatório.

- *NTP Timestamp* : Estampilha temporal global do sistema. A sua utilização é útil para efectuar a sincronização de diferentes sessões RTP.
- *RTP Timestamp* : Estampilha temporal enviada no último pacote RTP processado.
- *Sender's Packet Count* : Número total de pacotes enviados pelo emissor até ao momento que gera este relatório.
- *Sender's Octet Count* : Número total de octetos enviados até ao momento em que gera este relatório.

Estatísticas das fontes

Os campos que a seguir se descrevem referem-se a uma determinada fonte emissora de tráfego RTP.

- *SSRC n* : Identificador da fonte à qual dizem respeito as estatísticas presente neste bloco.
- *fraction lost* : A percentagem de pacotes, gerados pela fonte em questão, que foram perdidos desde da geração do último pacote SR ou RR.
- *cumulative number of packet losts* : Número total de perdas de pacotes verificadas desde o início da sessão.
- *higest number received* : Último número de sequência recebido em pacotes RTP gerados pela fonte em questão.
- *jitter* : Uma estatística da variação de chegada de pacotes RTP gerados pela fonte (nas mesmas unidades das estampilhas RTP).
- *Last SR* : Estampilha temporal do último relatório gerado pela fonte.
- *DLSR*: Diferença entre o instante em que se recebeu o último SR (*Sender Report*) por parte da fonte, e o instante em que se está a gerar este pacote. Estes dois últimos parâmetros permitem a cada uma das entidades estimar o tempo de ida e volta de um pacote²⁶ até a uma dada entidade.

Como foi também referido existem mensagens associadas ao canal RTCP e geradas por cada entidade interveniente, que disponibilizam ao grupo informações de alto nível sobre os utilizadores que participam numa dada sessão RTP. Por exemplo, todas as entidades no início de uma dada sessão terão que enviar para o grupo mensagens do tipo: NAME - que identifica o utilizador presente numa dada sessão RTP;

²⁶ referido como *round-trip time*

EMAIL - endereço de correio electrónico desse mesmos utilizador; LOC - A localização geográfica do utilizador, etc. Da mesma forma quando um dado utilizador abandona uma dada sessão RTP terá que informar o grupo onde participava desse facto. Para tal será enviada uma mensagem do tipo BYE.

Um outro identificador associado a cada utilizador é o CNAME (*Canonical NAME*). Este deverá ser único para cada utilizador e poderá ser utilizado para associar dados que, embora transportados em diferentes canais RTP, pertencem à mesma fonte geradora. Um caso típico da sua utilização é a sincronização de diferentes media emitidos por um mesmo utilizador.

Para finalizar importa referir que, apesar dos mecanismos aqui descritos, o RTCP não se assume por si só como um protocolo de gestão de grupos, mas antes como um canal onde são trocadas mensagens de suporte a esse tipo de funcionalidades.

Por forma a que a utilização do protocolo seja escalável é aconselhado que a ocupação, em termos de largura de banda, das mensagens do protocolo RTCP não ultrapasse os 5% do total ocupado pelas sessões RTP mais canais RTCP. Cada entidade, baseando-se no número total de participantes nas sessões RTP e no tipo de media trocados pelas entidades, deverá ponderar a sua taxa de emissão de relatórios para a rede por forma a não se verificar, com o aumento progressivo de participantes, uma saturação da rede provocada pelos canais RTCP.

Anexo II

Código-fonte de alguns módulos do *middleware*

Código do *applet* SimplePlayerApplet.class

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.lang.String;
import java.net.URL;
import java.net.MalformedURLException;
import java.io.IOException;
import java.util.Properties;
import javax.media.*;

public class SimplePlayerApplet extends Applet implements ControllerListener {

    // media Player
    Player player = null;
    // component in which video is playing
    Component visualComponent = null;
    // controls gain, position, start, stop
    Component controlComponent = null;
    // displays progress during download
    Component progressBar = null;
    boolean firstTime = true;
    long CachingSize = 0L;
    Panel panel = null;
    int controlPanelHeight = 0;
    int videoWidth = 0;
    int videoHeight = 0;

    /**
     Read the applet file parameter and create the media
     player.
     */
    public void init() {
        //$ System.out.println("Applet.init() is called");
    }
}
```

```

setLayout(null);
setBackground(Color.white);
panel = new Panel();
panel.setLayout( null );
add(panel);
panel.setBounds(0, 0, 320, 240);

// input file name from html param
String mediaFile = null;
// URL for our media file
MediaLocator mrl = null;
URL url = null;

// Get the media filename info.
// The applet tag should contain the path to the
// source media file, relative to the html page.

If ((mediaFile = getParameter("FILE")) == null)
    Fatal("Invalid media file parameter");

try {
    url = new URL(getDocumentBase(), mediaFile);
    mediaFile = url.toExternalForm();
} catch (MalformedURLException meu) {
}

try {
    // Create a media locator from the file name
    if ((mrl = new MediaLocator(mediaFile)) == null)
        Fatal("Can't build URL for " + mediaFile);

    /*
    try {
        JMFSecurity.enablePrivilege.invoke(JMFSecurity.privilegeManager,
            JMFSecurity.writePropArgs);
        JMFSecurity.enablePrivilege.invoke(JMFSecurity.privilegeManager,
            JMFSecurity.readPropArgs);
        JMFSecurity.enablePrivilege.invoke(JMFSecurity.privilegeManager,
            JMFSecurity.connectArgs);
    } catch (Exception e) {}
    */

    // Create an instance of a player for this media
    try {
        player = Manager.createPlayer(mrl);
    } catch (NoPlayerException e) {
        System.out.println(e);
        Fatal("Could not create player for " + mrl);
    }

    // Add ourselves as a listener for a player's events
    player.addControllerListener(this);

} catch (MalformedURLException e) {
    Fatal("Invalid media file URL!");
} catch (IOException e) {
    Fatal("IO exception creating player for " + mrl);
}

// This applet assumes that its start() calls
// player.start(). This causes the player to become
// realized. Once realized, the applet will get
// the visual and control panel components and add
// them to the Applet. These components are not added
// during init() because they are long operations that
// would make us appear unresponsive to the user.
}

/**
//Start media file playback. This function is called the
//first time that the Applet runs and every
//time the user re-enters the page.
*/

public void start() {
    //$ System.out.println("Applet.start() is called");
    // Call start() to prefetch and start the player.

```

```

        If (player != null)
            player.start();
    }

    /**
    //Stop media file playback and release resource before
    //Leaving the page.
    */
    public void stop() {
        //$ System.out.println("Applet.stop() is called");
        if (player != null) {
            player.stop();
            player.deallocate();
        }
    }

    public void destroy() {
        //$ System.out.println("Applet.destroy() is called");
        player.close();
    }

    /**
    This controllerUpdate function must be defined in order to
    implement a ControllerListener interface. This
    function will be called whenever there is a media event
    */
    public synchronized void controllerUpdate(ControllerEvent event) {
        // If we're getting messages from a dead player,
        // just leave
        if (player == null)
            return;

        // When the player is Realized, get the visual
        // and control components and add them to the Applet
        if (event instanceof RealizeCompleteEvent) {
            if (progressBar != null) {
                panel.remove(progressBar);
                progressBar = null;
            }

            int width = 320;
            int height = 0;
            if (controlComponent == null)
                if ((controlComponent =
                    player.getControlPanelComponent()) != null) {

                    controlPanelHeight = controlComponent.getPreferredSize().height;
                    panel.add(controlComponent);
                    height += controlPanelHeight;
                }
            if (visualComponent == null)
                if ((visualComponent =
                    player.getVisualComponent()) != null) {
                    panel.add(visualComponent);
                    Dimension videoSize = visualComponent.getPreferredSize();
                    videoWidth = videoSize.width;
                    videoHeight = videoSize.height;
                    width = videoWidth;
                    height += videoHeight;
                    visualComponent.setBounds(0, 0, videoWidth, videoHeight);
                }

            panel.setBounds(0, 0, width, height);
            if (controlComponent != null) {
                controlComponent.setBounds(0, videoHeight,
                    width, controlPanelHeight);
                controlComponent.invalidate();
            }
        }
        else if (event instanceof CachingControlEvent) {
            if (player.getState() > Controller.Realizing)
                return;
            // Put a progress bar up when downloading starts,
            // take it down when downloading ends.
            CachingControlEvent e = (CachingControlEvent) event;
            CachingControl cc = e.getCachingControl();

```

```
// Add the bar if not already there ...
if (progressBar == null) {
    if ((progressBar = cc.getControlComponent()) != null) {
        panel.add(progressBar);
        panel.setSize(progressBar.getPreferredSize());
        validate();
    }
}
} else if (event instanceof EndOfMediaEvent) {
    // We've reached the end of the media; rewind and
    // start over
    player.setMediaTime(new Time(0));
    player.start();
} else if (event instanceof ControllerErrorEvent) {
    // Tell TypicalPlayerApplet.start() to call it a day
    player = null;
    Fatal(((ControllerErrorEvent)event).getMessage());
} else if (event instanceof ControllerClosedEvent) {
    panel.removeAll();
}
}

void Fatal (String s) {
    // Applications will make various choices about what
    // to do here. We print a message
    System.err.println("FATAL ERROR: " + s);
    throw new Error(s); // Invoke the uncaught exception
                       // handler System.exit() is another
                       // choice.
}
}
```

Código do *applet* **Auscultador.class**

```
/*
Applet que recolhe as propriedades do sistema cliente e da QoS de rede.
Computa as adaptações a transmitir às aplicações multicast com base numa fórmula.
Envia os resultados para formulários HTML via LiveConnect.
Funciona em Internet Explorer e Nestcape.
(c) Sérgio Deusdado - Julho de 2002.
*/

import java.lang.String;
import java.lang.Math;
import com.ms.security.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import java.applet.AppletContext;
import java.net.URL;
import java.net.MalformedURLException;
import java.io.*;
import java.net.*;
import java.util.*;
import netscape.javascript.*;
import netscape.security.PrivilegeManager;

public class Auscultador extends Applet implements ActionListener {
    FontMetrics fm;
    int xtab, ystep ;
    String nl ; // nova linha
    String[] graphString = new String[20] ;
    int strings = 0;
    //TextArea ta = new TextArea (70, 70);
    Button startbutton = new Button("Recolhe informação e Computa adaptação" ) ;
    private static String execcommand;
    String dir_destino;
    String tipoSO,tipoProcessador,memTotal,memDisponivel;
    String processos,estadoProcessador,memEstado,swapTamanho;
    JSObject win;

    public void init() {

        try {
            if (Class.forName("com.ms.security.PolicyEngine") != null) { // Para o IE
                PolicyEngine.assertPermission(PermissionID.SYSTEM);
            }
        }
        catch (Throwable cnfe) {
        }

        try{
            PrivilegeManager.enablePrivilege("SuperUser") ; // Para o NN

```

```

    }
    catch(Exception cnfe) {
        System.out.println("netscape.security.PrivilegeManager class not found");
    }

    dir_destino=System.getProperty("user.home");
    // caso sejam sistemas Unix as propriedades são inseridas num ficheiro usando o
comando top
        if(System.getProperty("os.name").indexOf("Linux") != -1 ||
System.getProperty("os.name").indexOf("SUNOS") != -1 ||
System.getProperty("os.name").indexOf("Solaris") != -1)
        {
            dir_destino += "/";
            executand = "top >"+dir_destino+"propriedades.tmp" ;
        }
    // caso sejam sistemas windows então o utilitario msinfo32.exe escreve num
ficheiro as propriedades
        if(System.getProperty("os.name").indexOf("Windows") != -1)
        {
            dir_destino += "\\";
            File fich = new File("C:\\Program Files\\Common Files\\Microsoft
Shared\\MSInfo\\msinfo32.exe");
            if (fich.exists())
                executand = "C:\\Program Files\\Common Files\\Microsoft
Shared\\MSInfo\\msinfo32 /report "+dir_destino+"propriedades.tmp /categories
+SystemSummary";
            else
            {
                fich = new File("C:\\Programas\\Ficheiros comuns\\Microsoft
Shared\\MSInfo\\msinfo32.exe");
                if (fich.exists())
                {
                    executand = "C:\\Programas\\Ficheiros comuns\\Microsoft
Shared\\MSInfo\\msinfo32 /report "+dir_destino+"propriedades.tmp /categories
+SystemSummary";
                }
                else abrirFicheiro();
            }
        }
        this.setBackground(Color.white) ;
        startbutton.addActionListener(this) ;
        add(startbutton) ;
        startbutton.setBackground(Color.lightGray) ;
    }
    public void abrirFicheiro(){
        Frame frame0 = new Frame();
        FileDialog fd = new FileDialog(frame0, "Abrir ficheiro MSINFO32.exe",
FileDialog.LOAD);
        fd.show();
        String directoria = fd.getDirectory();
        String nomeFicheiro = fd.getFile();

```

```

        executando = directoria + nomeFicheiro + " /report
"+dir_destino+"propriedades.tmp /categories +SystemSummary";
    }

    public void actionPerformed(ActionEvent e) {
        if( (e.getActionCommand()).equals("Recolhe informação e Computa adaptação"))
        {
            try{
                PrivilegeManager.enablePrivilege("UniversalExecAccess"); // required for NN
            }
            catch(Exception cnfe) {
                System.out.println("netscape.security.PrivilegeManager class not found");
            }
            try {
                Process proc = Runtime.getRuntime().exec(executando);
            }
            catch(IOException ieo) {
                System.out.println("Não foi possível recolher as propriedades do sistema." +
executando);
            }

            // caso sejam sistemas Unix
            if(System.getProperty("os.name").indexOf("Linux") != -1 ||
System.getProperty("os.name").indexOf("SUNOS") != -1 ||
System.getProperty("os.name").indexOf("Solaris") != -1)
            {
                lerFicheiroUnix(dir_destino+"propriedades.tmp");
                // enviar para o FORM da página HTML via LiveConnect
                try {
                    JScript win = (JScript)JScript.getWindow(this);
                    win.eval("EscreveParametroMT('"+processos+"');");
                    win.eval("EscreveParametroMD('"+memEstado+"');");
                    win.eval("EscreveParametroCPU('"+estadoProcessador+"');");
                    win.eval("EscreveParametroSO('"+swapTamanho+"');");
                }
                catch (Throwable cnfe){ }
            }
            // caso sejam sistemas Windows
            if(System.getProperty("os.name").indexOf("Windows") != -1)
            {
                lerFicheiroWindows(dir_destino+"propriedades.tmp");

                // enviar para o FORM da página HTML via LiveConnect
                try {
                    JScript win = (JScript)JScript.getWindow(this);
                    win.eval("EscreveParametroMT('"+memTotal+"');");
                    win.eval("EscreveParametroMD('"+memDisponivel+"');");
                    win.eval("EscreveParametroCPU('"+tipoProcessador+"');");
                    win.eval("EscreveParametroSO('"+tipoSO+"');");
                    win.eval("EscreveParametroPOP('"+estadoProcessador+"');");
                    win.eval("calcula_adaptacao();");
                }
            }
        }
    }
}

```

```

        }
        catch (Throwable cnfe){ }
    }
}
}

//-----
public void lerFicheiroWindows (String fileName){

    StringBuffer buf = new StringBuffer();

    File teste = new File (fileName);
    showStatus("Aguarde por favor, examinado o sistema.\n");
    do {}
    while (!teste.exists());

    do {}
    while (teste.length()<1000);
    showStatus("A ler as propriedades.\n");
    try {
        FileInputStream fis = new FileInputStream(fileName);
        InputStreamReader insr = new InputStreamReader(fis,"UnicodeLittle");
        Reader in = new BufferedReader(insr);

        int ch;
        while ((ch = in.read()) > -1) {
            buf.append((char)ch);
        }
        in.close();
        teste = new File (fileName);
        teste.delete();

    } catch (IOException e) {
        e.printStackTrace();
        System.out.println("Não foi possível ler o ficheiro.\n");
    }
    recolhePropriedadesWindows(buf);
}
//-----
public void lerFicheiroUnix (String fileName){

    StringBuffer buf = new StringBuffer();

    File teste = new File (fileName);
    showStatus("Aguarde por favor, examinado o sistema.\n");
    do {}
    while (!teste.exists());

    do {}
    while (teste.length()<1000);
    showStatus("A ler as propriedades.\n");

```

```

try {
    FileInputStream fis = new FileInputStream(fileName);
    InputStreamReader insr = new InputStreamReader(fis,"UnicodeLittle");
    Reader in = new BufferedReader(insr);

    int ch;
    while ((ch = in.read()) > -1) {
        buf.append((char)ch);
    }
    in.close();
    teste = new File (fileName);
    teste.delete();

} catch (IOException e) {
    e.printStackTrace();
    System.out.println("Não foi possível ler o ficheiro.\n");
}
recolhePropriedadesUnix(buf);
}
//-----
public void recolhePropriedadesWindows (StringBuffer buf ){

String substring, str;
int i, n, j, max;

    // Procurar no ficheiro o tipo de SO
    substring = "OS Name";
    max = buf.length() - substring.length();

testel:
    for (i = 0; i <= max; i++) {
        n = substring.length();
        j = i;
        int k = 0;
        while (n-- != 0) {
            if (buf.charAt(j++) != substring.charAt(k++)) {
                continue testel;
            }
        }
        n = j;
        // Encontra a substring que dita o tipo de SO
        while (buf.charAt(n)!='\n') n++;
        str = buf.toString();
        tipoSO = str.substring(j,n);
        tipoSO = tipoSO.trim();

        break testel;
    }
// Para a versão portuguesa do Windows
    if (tipoSO==null)

```

```

{
substring = "Nome do SO";
max = buf.length() - substring.length();
teste2:
for (i = 0; i <= max; i++) {
n = substring.length();
j = i;
int k = 0;
while (n-- != 0) {
if (buf.charAt(j++) != substring.charAt(k++)) {
continue teste2;
}
}
n = j;
// Encontra a substring que dita o tipo de SO
while (buf.charAt(n)!='\n') n++;
str = buf.toString();
tipoSO = str.substring(j,n);
tipoSO = tipoSO.trim();

break teste2;
}
}

// Procurar no ficheiro o tipo de Processador
substring = "Processor";
max = buf.length() - substring.length();

teste3:
for (i = 0; i <= max; i++) {
n = substring.length();
j = i;
int k = 0;
while (n-- != 0) {
if (buf.charAt(j++) != substring.charAt(k++)) {
continue teste3;
}
}
}
n = j;
// Encontra a substring que dita o tipo de processador
while (buf.charAt(n)!='\n') n++;
str = buf.toString();
tipoProcessador = str.substring(j,n);
tipoProcessador = tipoProcessador.trim();

break teste3;
}

// Para a versão portuguesa do Windows
if (tipoProcessador==null)
{
substring = "Processador";

```

```

max = buf.length() - substring.length();
teste4:
  for (i = 0; i <= max; i++) {
    n = substring.length();
    j = i;
    int k = 0;
    while (n-- != 0) {
      if (buf.charAt(j++) != substring.charAt(k++)) {
        continue teste4;
      }
    }
    n = j;
    while (buf.charAt(n)!='\n') n++;
    str = buf.toString();
    tipoProcessador = str.substring(j,n);
    tipoProcessador = tipoProcessador.trim();

    break teste4;
  }
}
// Procurar no ficheiro a quantidade de memoria total
substring = "Total Physical Memory";
max = buf.length() - substring.length();

teste5:
  for (i = 0; i <= max; i++) {
    n = substring.length();
    j = i;
    int k = 0;
    while (n-- != 0) {
      if (buf.charAt(j++) != substring.charAt(k++)) {
        continue teste5;
      }
    }
    n = j;
    // Encontra a substring que dita a memoria total
    while (buf.charAt(n)!='K') n++;
    str = buf.toString();
    memTotal = str.substring(j,n);
    memTotal = memTotal.trim();

    break teste5;
  }
}
// Para a versão portuguesa do Windows
if (memTotal==null)
{
  substring = "Memória física total";
  max = buf.length() - substring.length();
  teste6:
  for (i = 0; i <= max; i++) {
    n = substring.length();

```

```

j = i;
int k = 0;
while (n-- != 0) {
    if (buf.charAt(j++) != substring.charAt(k++)) {
        continue teste6;
    }
}
n = j;

while (buf.charAt(n)!='K') n++;
str = buf.toString();
memTotal = str.substring(j,n);
memTotal = memTotal.trim();
break teste6;
}
}

// Procurar no ficheiro a quantidade de memoria disponivel
substring = "Available Physical Memory";
max = buf.length() - substring.length();

teste7:
for (i = 0; i <= max; i++) {
    n = substring.length();
    j = i;
    int k = 0;
    while (n-- != 0) {
        if (buf.charAt(j++) != substring.charAt(k++)) {
            continue teste7;
        }
    }
    n = j;
    // Encontra a substring que dita a memoria disponivel
    while (buf.charAt(n)!='K') n++;
    str = buf.toString();
    memDisponivel = str.substring(j,n);
    memDisponivel = memDisponivel.trim();

    break teste7;
}

// Para a versão portuguesa do Windows
if (memDisponivel==null)
{
    substring = "Memória física disponível";
    max = buf.length() - substring.length();
    teste8:
    for (i = 0; i <= max; i++) {
        n = substring.length();
        j = i;
        int k = 0;
        while (n-- != 0) {

```

```

        if (buf.charAt(j++) != substring.charAt(k++)) {
            continue teste8;
        }
    }
    n = j;

    while (buf.charAt(n)!='K') n++;
    str = buf.toString();
    memDisponivel = str.substring(j,n);
    memDisponivel = memDisponivel.trim();
    break teste8;
    }
}
// calcular a percentagem de ocupação do processador com base na memória
usada

    try {
        float md = Float.valueOf(memDisponivel.trim()).floatValue();
        float mt = Float.valueOf(memTotal.trim()).floatValue();

        float mo = mt-md;
        int ep = (int)((mo/mt/3)*100);
        estadoProcessador = new Integer(ep).toString();
        estadoProcessador = estadoProcessador + '%';
    } catch (NumberFormatException nfe) {
        System.out.println("NumberFormatException: " + nfe.getMessage());
    }
}

//-----
public void calculaMododeAdaptacao (String lb, String rt){
    String ad_lb="";
    String ad_fps="";
    String ad_cod_v="";
    String ad_cor="";
    String par_lb="";
    String par_fps="";
    String par_cod_v="";
    String par_cor="";
    String ad_cod_a="";
    String par_cod_a="";

    // calcula o modo de adaptação para as aplicações de videoconferência multicast

    double K = 0.02;

    float B = Float.valueOf(lb.trim()).floatValue();
    float RTT = Float.valueOf(rt.trim()).floatValue();

    float mt = Float.valueOf(memTotal.trim()).floatValue();
    float M = Float.valueOf(memDisponivel.trim()).floatValue();
    int POP = (int)((mt-M)/mt/3)*100;

```

```

int Modo = (int) ((B/(RTT/2)+ M/POP)*K);

if (Modo < 1) {Modo = 1;};
if (Modo > 5) {Modo = 5;};

//Tabela de modos de adaptação, estrutura: Kbps - fps - codec vídeo - cor/P&B -
codec áudio
String modos [] [] =
{
    { "64", "10", "h263", "gray", "lpc" },
    { "128", "15", "h263", "", "gsm" },
    { "256", "20", "h261", "", "dvi" },
    { "512", "25", "h261", "", "pcm" },
    { "1024", "30", "h261", "", "116" }
};

String extenso [] [] =
{
    { "64", "10", "H.263", "Escala de Cinzentos", "LPC" },
    { "128", "15", "H.263", "Imagem colorida", "GSM" },
    { "256", "20", "H.261", "Imagem colorida", "DVI" },
    { "512", "25", "H.261", "Imagem colorida", "PCM" },
    { "1024", "30", "H.261", "Imagem colorida", "Linear-16 bit" }
};

float BW_Modo = Float.valueOf(modos [Modo-1] [0].trim()).floatValue();
//Evita que se tente transmitir numa taxa de transmissão superior à largura de
banda medida
while (( BW_Modo > B ) && ( Modo > 1 )) { Modo -=1;};

// Atribuições dos parâmetros
ad_lb = extenso [Modo-1] [0];
par_lb = modos [Modo-1] [0];
ad_fps = extenso [Modo-1] [1];
par_fps = modos [Modo-1] [1];
ad_cod_v = extenso [Modo-1] [2];
par_cod_v = modos [Modo-1] [2];
ad_cor = extenso [Modo-1] [3];
par_cor = modos [Modo-1] [3];
ad_cod_a = extenso [Modo-1] [4];
par_cod_a = modos [Modo-1] [4];

try
{
    JavaScript win = (JavaScript)JavaScript.getWindow(this);
    //Resultados da adaptação - Parâmetros adaptados
    win.eval("EscreveParametroLB_ad('"+ad_lb+"');");
    win.eval("EscreveParametroFPS_ad('"+ad_fps+"');");
    win.eval("EscreveParametroCDV_ad('"+ad_cod_v+"');");
    win.eval("EscreveParametroCOR_ad('"+ad_cor+"');");
    win.eval("EscreveParametroLB_par('"+par_lb+"');");
    win.eval("EscreveParametroFPS_par('"+par_fps+"');");
    win.eval("EscreveParametroCDV_par('"+par_cod_v+"');");
}

```

```

win.eval("EscreveParametroCOR_par('"+par_cor+"'");");
win.eval("EscreveParametroCDA_ad('"+ad_cod_a+"'");");
win.eval("EscreveParametroCDA_par('"+par_cod_a+"'");");
}
catch (Throwable cnfe){ }

try{
getAppletContext().showDocument(new URL
("http://www.esa.ipb.pt/multicast/adapt/conexao.php?"

+ "%lbm="+par_lb+"%fps="+par_fps+"%cdv="+par_cod_v+"%cor="+par_cor+"%cda="+par_cod_a), "_b
lank");
}
catch (MalformedURLException e) {
System.out.println("URL malformado");
}

}
//-----
public void recolhePropriedadesUnix (StringBuffer buf ){

String substring, str;
int i, n, j, max;

// Procurar no ficheiro o tipo de SO
substring = "processes";
max = buf.length() - substring.length();

testel:
for (i = 0; i <= max; i++) {
n = substring.length();
j = i;
int k = 0;
while (n-- != 0) {
if (buf.charAt(j++) != substring.charAt(k++)) {
continue testel;
}
}
n = j;
// Encontra a substring que dita os processos correntes
while (buf.charAt(n) != '\n') n++;
str = buf.toString();
processos = str.substring(j, n);
processos = processos.trim();

break testel;
}

// Procurar no ficheiro a ocupaço do Processador
substring = "CPU states";

```

```
        max = buf.length() - substring.length();
teste2:
    for (i = 0; i <= max; i++) {
        n = substring.length();
        j = i;
        int k = 0;
        while (n-- != 0) {
            if (buf.charAt(j++) != substring.charAt(k++)) {
                continue teste2;
            }
        }
        n = j;
        // Encontra a substring que dita a ocupação do Processador
        while (buf.charAt(n)!='\n') n++;
        str = buf.toString();
        estadoProcessador = str.substring(j,n);
        estadoProcessador = estadoProcessador.trim();

        break teste2;
    }

// Procurar no ficheiro o estado da memória
substring = "Mem:";
max = buf.length() - substring.length();

teste3:
    for (i = 0; i <= max; i++) {
        n = substring.length();
        j = i;
        int k = 0;
        while (n-- != 0) {
            if (buf.charAt(j++) != substring.charAt(k++)) {
                continue teste3;
            }
        }
        n = j;
        // Encontra a substring que dita o estado da memória
        while (buf.charAt(n)!='\n') n++;
        str = buf.toString();
        memEstado = str.substring(j,n);
        memEstado = memEstado.trim();

        break teste3;
    }

// Procurar no ficheiro o espaço de swap
substring = "Swap:";
max = buf.length() - substring.length();

teste4:
    for (i = 0; i <= max; i++) {
```

```
n = substring.length();
j = i;
int k = 0;
while (n-- != 0) {
    if (buf.charAt(j++) != substring.charAt(k++)) {
        continue teste4;
    }
}
n = j;
// Encontra a substring que dita o espaço de swap
while (buf.charAt(n)!='\n') n++;
str = buf.toString();
swapTamanho = str.substring(j,n);
swapTamanho = swapTamanho.trim();
break teste4;
}
}
}
```

Código do *applet* `leParametros.class`

```

/*
Recebe do applet Auscultador.class os resultados calculados para a adaptação.
Os dados são transmitidos por query apenas ao URL.
Os dados são inseridos em formulários HTML via LiveConnect.
Funciona em Internet Explorer e Nestcape.
(c) Sérgio Deusdado - Julho de 2002.
*/

import java.applet.*;
import java.awt.*;
import java.lang.*;
import java.net.*;
import netscape.javascript.JSObject;
import netscape.javascript.*;

public class leParametros extends Applet {
    JSObject win;

    public void init() {
        // obtém o URL
        String s = getDocumentBase().toString();
        // extrai a parte dos dados
        String dados = s.substring(s.indexOf('?') + 1);
        // retira o 1º título
        dados = dados.substring(5);
        // decompoe a mensagem dos dados
        String lbn = dados.substring(0,dados.indexOf('%'));
        dados = dados.substring(dados.indexOf('%')+5);
        String fps = dados.substring(0,dados.indexOf('%'));
        dados = dados.substring(dados.indexOf('%')+5);
        String cdv = dados.substring(0,dados.indexOf('%'));
        dados = dados.substring(dados.indexOf('%')+5);
        String cor = dados.substring(0,dados.indexOf('%'));
        dados = dados.substring(dados.indexOf('%')+5);
        String cda = dados.trim();
        try
        {
            JSObject win = (JSObject)JSObject.getWindow(this);
            //Resultados da adaptação - Parâmetros adaptados para a janela da conexão
            win.eval("EscreveParametroLB_par('"+lbn+"'");
            win.eval("EscreveParametroFPS_par('"+fps+"'");
            win.eval("EscreveParametroCDV_par('"+cdv+"'");
            win.eval("EscreveParametroCOR_par('"+cor+"'");
            win.eval("EscreveParametroCDA_par('"+cda+"'");
        }
        catch (Throwable cnfe){ }
    }
}

```

Código do *applet* **IniciaApAdaptada.class**

```

/*
Aplicação para despoletar, via browser, as ferramentas multicast
necessárias aos diferentes tipos de conferência.
Pré-parametriza as aplicações de áudio(rat) e vídeo(vic).
Funciona em Internet Explorer e Nestcape.
(c) Sérgio Deusdado - Julho de 2002.
*/

import com.ms.security.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import java.io.* ;
import java.util.*;
import netscape.security.PrivilegeManager;

public class IniciaApAdaptada extends Applet implements ActionListener {
    String val_enderecov, val_enderecoa;
    String val_ttl;
    String val_portav, val_portaa;
    String val_lbm, val_cda, val_cdv, val_cor;
    String val_fps;
    String val_video;
    String val_audio;
    String param_cor;

    String nl ;      // new line character
    String[] graphString = new String[20] ;
    int strings = 0;
    TextArea ta = new TextArea (25, 80);
    Button startbutton = new Button("Iniciar Conferência") ;
    private static String execommand1, execommand2;

    public void init() {
        try {
            val_enderecov = this.getParameter("enderecov");
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        try {
            val_enderecoa = this.getParameter("enderecoa");
        }
        catch(Exception e) {
            e.printStackTrace();
        }

        try {
            val_ttl = this.getParameter("ttl");

```

```
}
catch(Exception e) {
    e.printStackTrace();
}
try {
    val_portav = this.getParameter("portav");
}
catch(Exception e) {
    e.printStackTrace();
}
try {
    val_portaa = this.getParameter("portaa");
}
catch(Exception e) {
    e.printStackTrace();
}
try {
    val_lbm = this.getParameter("lb_max");
}
catch(Exception e) {
    e.printStackTrace();
}
try {
    val_fps = this.getParameter("fps");
}
catch(Exception e) {
    e.printStackTrace();
}
try {
    val_cdv = this.getParameter("cod_video");
}
catch(Exception e) {
    e.printStackTrace();
}
try {
    val_cda = this.getParameter("cod_audio");
}
catch(Exception e) {
    e.printStackTrace();
}
try {
    val_cor = this.getParameter("cor");
}
catch(Exception e) {
    e.printStackTrace();
}
try {
    val_video = this.getParameter("video");
}
catch(Exception e) {
    e.printStackTrace();
}
```

```

    }
    try {
        val_audio = this.getParameter("audio");
    }
    catch(Exception e) {
        e.printStackTrace();
    }

    if (val_cor!="") {param_cor="";} else {param_cor=" -c "+val_cor;};

    executando1 = "vic -t "+val_ttl+" -B "+val_lbm+param_cor+" -f "+val_cdv+" -F
"+val_fps+" -X transmitOnStartup="+val_video+" "+val_enderecov+"/"+val_portav;
    executando2 = "rat -t "+val_ttl+" -f "+val_cda+"/dvi/gsm
"+val_enderecoa+"/"+val_portaa;

    try {
        if (Class.forName("com.ms.security.PolicyEngine") != null) { // para o IE
            PolicyEngine.assertPermission(PermissionID.SYSTEM);
        }
    }
    catch (Throwable cnfe) {
    }

    this.setBackground(Color.white) ;
    startbutton.addActionListener(this) ;
    add(startbutton) ;
    startbutton.setBackground(Color.blue) ;
    startbutton.setForeground(Color.white) ;

    try{
        PrivilegeManager.enablePrivilege("UniversalExecAccess") ; // necessário no NN
    }
    catch(Exception cnfe) {
        System.out.println("netscape.security.PrivilegeManager class not found") ;
    }

    if(System.getProperty("os.name").equals("Windows NT") ||
System.getProperty("os.name").equals("Windows 2000")) // NT ou Win2000, ajustar path
    {
        if (val_cor!="") {param_cor="";} else {param_cor=" -c "+val_cor;};
        executando1 = "vic -t "+val_ttl+" -B "+val_lbm+param_cor+" -f "+val_cdv+" -F
"+val_fps+" -X transmitOnStartup="+val_video+" "+val_enderecov+"/"+val_portav;
        executando2 = "rat -t "+val_ttl+" -f "+val_cda+"/dvi/gsm
"+val_enderecoa+"/"+val_portaa;
    }
}

public void actionPerformed(ActionEvent e) {
    if( (e.getActionCommand()).equals("Iniciar Conferência")) {
    try{
        PrivilegeManager.enablePrivilege("UniversalExecAccess") ; // necessário no NN
    }
}

```

```
catch(Exception cnfe) {
    System.out.println("netscape.security.PrivilegeManager class not found") ;
}
try {
    Process proc1 = Runtime.getRuntime().exec(execommand1) ;
    Process proc2 = Runtime.getRuntime().exec(execommand2) ;

}
catch(IOException ieo) {
    System.out.println("Impossível abrir" + execommand1 + " ou " + execommand2) ;
}
}
}
```