

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA APLICADA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

FELIPE CECAGNO

**Análise e desenvolvimento de uma
solução para distribuição de vídeo
adaptativo em redes heterogêneas**

Trabalho de Graduação

Prof. Dr. Valter Roesler
Orientador

Porto Alegre, dezembro de 2010

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Pró-Reitor de Coordenação Acadêmica: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Prof^a. Valquíria Link Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do Curso de Ciência da Computação: Prof. João César Netto

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Primeiramente gostaria de deixar claro que esse agradecimento se tornou mais importante a partir do momento em que o discurso dos formandos durante a cerimônia de formatura foi suspenso. Por isso, aqui escrevo o que potencialmente seria o meu discurso.

Gostaria de agradecer aos meus pais, por me ensinarem desde pequeno a valorizar a educação e a cultura. Também por me apoiarem financeiramente nos momentos em que eu precisei durante a graduação.

À minha namorada Débora, por estar sempre do meu lado desde o início da faculdade, e não brigar comigo toda vez que eu precisei estudar ou trabalhar até tarde.

Aos meus colegas durante a graduação Luiz, Bruno, Jonas, Gui e Rosália, pela parceria, pelas conversas, pelas idéias e principalmente pela inspiração.

Aos colegas do laboratório do PRAV, pelo divertimento e pela competência nesses últimos dois anos. Também ao prof. Valter, orientador neste trabalho e no laboratório, por depositar em mim enorme confiança e por sempre ouvir o que eu tenho pra dizer.

Por fim, agradeço à UFRGS e meus ótimos professores, que me proporcionaram uma formação de excelência e a oportunidade de realizar estudos no exterior, o que contribuiu muito na construção da visão de mundo que eu tenho hoje. Em especial, aos professores Luciana, Valter e Manuel, pelo grande voto de confiança que fizeram me recomendando para o intercâmbio.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	9
LISTA DE TABELAS	10
RESUMO	11
ABSTRACT	12
1 INTRODUÇÃO	13
2 ARQUITETURAS PARA TRANSMISSÃO MULTIMÍDIA	15
2.1 Tecnologias de transporte	15
2.1.1 UDP	15
2.1.2 TCP	16
2.2 Estratégias para distribuição de conteúdo	17
2.2.1 <i>Unicast</i>	17
2.2.2 <i>Multicast</i>	19
2.2.3 <i>Unicast</i> com servidores intermediários	20
3 PROTOCOLOS PARA TRANSMISSÃO MULTIMÍDIA EM TEMPO REAL	22
3.1 RTP e RTCP	22
3.2 RTMP	24
3.3 RTSP	25
3.4 RDT	32
4 PARAMETRIZAÇÃO DA CODIFICAÇÃO DE VÍDEO	33
4.1 Biblioteca de codificação	33
4.2 Parâmetros adaptativos	34
4.2.1 <i>Bitrate</i>	34
4.2.2 Resolução	35

4.2.3	FPS	36
4.3	Mecanismos de adaptação	36
4.3.1	Múltiplas taxas fixas	36
4.3.2	Adaptação dinâmica	36
5	PROPOSTA DE SOLUÇÃO	37
5.1	Arquitetura mista	37
5.1.1	Teste de alcance do <i>multicast</i>	37
5.1.2	Teste e seleção do servidor mais adequado	38
5.1.3	Solicitação de início e fim de transmissão	38
5.1.4	Adaptação	38
5.2	Hierarquização	38
5.3	Métricas de avaliação e testes	39
5.3.1	Atraso	39
5.3.2	Desempenho	40
5.3.3	Saturação	40
5.3.4	Alcançabilidade	40
6	ARQUITETURA DE VALIDAÇÃO	41
6.1	Estudo de caso - IVA	41
6.1.1	Apresentador	41
6.1.2	Moderador	42
6.1.3	Suíte	43
6.1.4	Visualizador	44
6.1.5	Chat	45
6.2	Detalhes técnicos	45
6.2.1	Arquitetura de distribuição	45
6.2.2	Protocolo para distribuição de áudio e vídeo	46
6.2.3	Protocolo de controle de sessão	47
6.2.4	Perfis de fluxos de dados	47
6.3	Detalhamento da implementação	48
6.3.1	Contextualização	48
6.3.2	Integração	48
6.3.3	Interação	49
6.3.4	Implementação do SSU	50
7	RESULTADOS	52
7.1	Teste preliminar	52
7.1.1	Ambiente de testes	53
7.1.2	Configuração das máquinas	54
7.1.3	Avaliação de atraso	54
7.2	Teste conclusivo	56
7.2.1	Ambiente de testes	56
7.2.2	Configuração das máquinas	57
7.2.3	Avaliação de atraso	57
7.2.4	Avaliação de funcionamento	59
7.2.5	Avaliação de desempenho	60

7.2.6	Avaliação de saturação	60
7.2.7	Avaliação de alcançabilidade	61
8	CONCLUSÕES	62
	REFERÊNCIAS	63
	ANEXO A RESOLUÇÕES DE VÍDEO	65
	APÊNDICE A EXEMPLO DE MENSAGEM SDP	66
	APÊNDICE B TESTE PRELIMINAR - CONFIGURAÇÃO DAS MÁQUINAS	69
	APÊNDICE C TESTE CONCLUSIVO - CONFIGURAÇÃO DAS MÁQUINAS	70

LISTA DE ABREVIATURAS E SIGLAS

ASCII	<i>American Standard Code for Information Interchange</i>
AVP	<i>Audio Video Profile</i>
DNS	<i>Domain Name System</i>
EAD	Ensino a Distância
FPS	<i>Frames Per Second</i>
FTP	<i>File Transfer Protocol</i>
HD	<i>High Definition</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HVGA	<i>Half VGA</i>
IP	<i>Internet Protocol</i>
IVA	Sistema Interativo de Vídeo e Áudio
OSI	<i>Open Systems Interconnection</i>
P2P	<i>Peer-to-peer</i>
POP	Ponto de Presença
PRAV	Projetos em Áudio e Vídeo
RDT	<i>Real Data Transport</i>
RNP	Rede Nacional de Ensino e Pesquisa
RTCP	<i>RTP Control Protocol</i>
RTMP	<i>Real-Time Messaging Protocol</i>
RTP	<i>Real-Time Protocol</i>
RTSP	<i>Real-Time Streaming Protocol</i>
RTT	<i>Round Trip Time</i>
SD	<i>Standard Definition</i>
SDP	<i>Session Description Protocol</i>
SIP	<i>Session Initiation Protocol</i>
SSU	Servidor de <i>Stream Unicast</i>

TCP *Transmission Control Protocol*
UDP *User Datagram Protocol*
URL *Uniform Resource Locator*
UVGA *Ultra VGA*
VGA *Video Graphics Array*
VoIP *Voice over IP*
WVGA *Wide VGA*

LISTA DE FIGURAS

Figura 2.1: Exemplo de sessão ponto a ponto	17
Figura 2.2: Exemplo de sessão multiponto	18
Figura 2.3: Transmissão de dados utilizando <i>unicast</i>	18
Figura 2.4: Transmissão de dados utilizando <i>multicast</i>	20
Figura 2.5: Transmissão de dados utilizando <i>unicast</i> com servidores interme- diários	20
Figura 3.1: Cabeçalho do RTP	22
Figura 3.2: Máquina de estados do RTSP	26
Figura 4.1: Comparação entre formatos de vídeo	34
Figura 6.1: Interface do Apresentador IVA na tela de seleção de canal	41
Figura 6.2: Interface do Apresentador IVA na tela de sessão	42
Figura 6.3: Interface do Moderador IVA	43
Figura 6.4: Interface da Suíte IVA	44
Figura 6.5: Professor e o conceito de telepresença	44
Figura 6.6: Modelo de integração entre as entidades do IVA	45
Figura 6.7: Cabeçalho utilizado no IVA - Padrão em todos os pacotes de áudio e vídeo	46
Figura 6.8: Cabeçalho utilizado no IVA - Obrigatório nos pacotes de áudio	46
Figura 6.9: Cabeçalho utilizado no IVA - Obrigatório nos pacotes de vídeo	47
Figura 6.10: Modelo de integração real, em que entidades podem não estar na rede <i>multicast</i>	48
Figura 6.11: Utilização de servidores de <i>stream</i> para realizar entrega em <i>unicast</i>	49
Figura 6.12: Interação entre as entidades envolvidas na solução	50
Figura 7.1: Ambiente real de testes	52
Figura 7.2: Ambiente de testes - teste preliminar	53
Figura 7.3: Ambiente de testes - teste conclusivo	56
Figura 7.4: Apresentador 1 recebendo o fluxo de dados em <i>multicast</i>	59
Figura 7.5: Apresentador 3 recebendo o fluxo de dados em <i>unicast</i>	59
Figura 7.6: Taxa de recepção de dados em <i>unicast</i> do Apresentador 5	59
Figura 7.7: Demanda de processamento e memória do SSU durante a avaliação	60
Figura 7.8: Taxa de recepção e envio de dados do SSU	61
Figura 7.9: Ganho em alcançabilidade	61

LISTA DE TABELAS

Tabela 7.1: Teste preliminar: medição de atraso em cada ponto	55
Tabela 7.2: Teste conclusivo: medição de atraso em cada ponto	58

RESUMO

A escolha da estratégia de distribuição em um sistema computacional interativo através de áudio e vídeo é um fator importantíssimo e impacta diretamente na escalabilidade do sistema e alcançabilidade dos fluxos de dados. Em muitos projetos o *multicast* como estratégia de distribuição é evitado porque o mesmo não funciona bem na internet, embora seja mais eficiente do que o *unicast*. Nesses casos, alternativas diversas podem ser adotadas, dependendo dos requisitos mínimos de número de participantes e heterogeneidade da rede em que o sistema funcionará.

Este trabalho propõe uma solução para distribuição de vídeo que objetiva uma maior escalabilidade se comparado a distribuição *unicast* pura, e uma maior alcançabilidade se comparada a distribuição *multicast* pura.

Palavras-chave: Vídeo adaptativo, streaming, redes heterogêneas, IVA.

**Analysis and development of a solution for adaptative video stream
distribution on heterogeneous networks**

ABSTRACT

The choice of the distribution strategy in an interactive computational system through audio and video is a very important factor and it directly impacts the system's scalability and the data flow reachability. In many projects the multicast is avoided as a distribution strategy because it doesn't work well on the internet although being more efficient than the unicast. In these cases, different alternatives can be adopted, depending on the minimum requirements that are the number of participants and the heterogeneity of the network in which the system will work.

This work proposes a solution for video distribution that aims a bigger scalability if compared to the pure unicast distribution and a larger reachability if compared to the pure multicast distribution.

Keywords: adaptative video, streaming, heterogeneous networks, IVA.

1 INTRODUÇÃO

A utilização de recursos multimídia como áudio e vídeo passou a ter um papel importante para as pessoas nos últimos anos com a popularização das redes de banda larga. Essa tecnologia revolucionou a maneira como as pessoas enxergam a internet, viabilizando o entretenimento através do vídeo sob demanda e rompendo distâncias com aplicações de tempo real, tais como ferramentas de VoIP, videoconferência e telemedicina.

Com redes mais rápidas e mais baratas a cada dia, muitas empresas veem uma grande oportunidade de reduzir custos com telefone e deslocamento de pessoal, pois reuniões e treinamentos podem ser realizados através de videoconferências na internet. Diferente de uma videochamada, em que a transmissão de áudio e vídeo ocorre entre dois pontos, a videoconferência pode conectar diversas pessoas distribuídas geograficamente. Dependendo do número de participantes e da tecnologia de transmissão adotada, podem surgir problemas como por exemplo o excesso de tráfego na rede e o gerenciamento de participantes.

Dependendo das características apresentadas pela rede, diferentes estratégias para distribuição de conteúdo podem ser adotadas. *Multicast* é a estratégia mais eficiente, mas somente uma pequena parte dos roteadores na internet possuem suporte nativo. Já o *unicast* é uma estratégia simples e funcional, mas é uma solução que não escala para um grande número de participantes. Em uma rede híbrida, formada por ilhas *multicast* ligadas via *unicast*, o ideal é uma solução que aproveite o melhor dentre as duas estratégias.

Além disso, a variedade de dispositivos com diferentes resoluções e capacidades de processamento e as diferentes taxas de transmissão encontradas na internet sugerem uma abordagem de qualidade de vídeo adaptativa (HALSALL, 2001). Em uma rede híbrida, tal abordagem pode ser utilizada na transmissão para otimizar os fluxos destinados aos participantes *unicast*. Nesse caso, a transmissão *multicast* seria utilizada pelos participantes que estão ao alcance do *multicast*, e os demais participantes poderiam usufruir de uma transmissão *unicast* personalizada, de acordo com suas características.

Com base no contexto apresentado, o presente trabalho tem o objetivo de fundamentar o desenvolvimento de uma solução prática para distribuição de vídeo adaptativo em redes heterogêneas. No capítulo 2 serão abordados tópicos relacionados com o transporte de conteúdo multimídia, como protocolos de transporte e estratégias para distribuição. O capítulo 3 apresentará quatro protocolos de nível de aplicação desenvolvidos para conteúdo de tempo real. No capítulo 4 será apresentada a parametrização da codificação de vídeo bem como mecanismos de adaptação. O capítulo 5 apresentará a solução proposta para o problema bem como métricas de

avaliação. Um estudo de caso é apresentado no capítulo 6 juntamente com o detalhamento da implementação, e os resultados são apresentados no capítulo 7. Por fim, conclusões e trabalhos futuros são detalhados no capítulo 8.

2 ARQUITETURAS PARA TRANSMISSÃO MULTIMÍDIA

Segundo (KUROSE; ROSS, 2007), a função da camada de rede no modelo de referência OSI é prover um mecanismo de comunicação lógica entre *hosts* (dispositivos). Essa comunicação acontece na internet através do protocolo IP (*Internet Protocol*), que oferece os serviços de endereçamento e fragmentação (POSTEL, 1981a).

Na camada superior, a camada de transporte, o canal de comunicação é entre processos, e não mais entre *hosts*. No TCP e no UDP, protocolos da camada de transporte mais utilizados na internet, tal função é realizada através do conceito de portas de comunicação, o que torna possível comunicações simultâneas em um mesmo dispositivo.

2.1 Tecnologias de transporte

2.1.1 UDP

O UDP é um protocolo de transporte sem conexão que funciona sobre o protocolo IP. O principal objetivo do protocolo é prover um mecanismo de troca de mensagens entre dois dispositivos com o mínimo esforço, ou seja, da forma mais simples possível. As mensagens UDP são chamadas de datagramas (POSTEL, 1980).

Segundo (TANENBAUM, 2003), a principal vantagem do uso de UDP em relação ao IP diretamente é a adição, no seu cabeçalho, das portas de origem e destino. Sem elas, a camada de transporte não saberia para qual aplicação entregar um datagrama recebido - esse recurso é conhecido como multiplexação. Em vista disso, o UDP assume utilização do IP na camada de rede.

Por visar um mecanismo simples, o UDP possui as seguintes características:

- Não garante entrega de datagramas;
- Não possui proteção contra datagramas duplicados;
- Não garante ordem entre datagramas;
- Não realiza controle de fluxo;
- Não garante a integridade dos dados transmitidos.

Devido às características citadas acima, o protocolo é tido como não confiável. Muito embora à primeira vista possa ser difícil enxergar um uso para um protocolo com tantas faltas de funcionalidades, o UDP é utilizado em uma série de aplicações,

chamadas de tolerantes a perdas (KUROSE; ROSS, 2007). Um exemplo é o DNS (*Domain Name System*), que utiliza o UDP porque gera pouca sobrecarga na rede, e perdas não são críticas - uma solicitação não respondida gerará uma nova solicitação após um *timeout* (TANENBAUM, 2003).

Além do DNS, o UDP é amplamente utilizado para transporte de dados com propriedades de tempo real. O fato do protocolo não prever nenhum mecanismo de retransmissão de dados é uma característica interessante para tais aplicações, em que um dado retransmitido pode não possuir valor. É o caso de aplicações interativas de áudio e vídeo, em que o atraso máximo aceitável é 400 milissegundos (KUROSE; ROSS, 2007).

O UDP também possui uma característica fundamental para transmissão em larga escala que é proporcionar a base para o funcionamento do *multicast* (também conhecido como UDP *multicast*). Tal assunto será abordado na seção 2.2.2.

2.1.2 TCP

O TCP surgiu para suprir a necessidade de um protocolo confiável entre um par de processos (POSTEL, 1981b). Diferentemente do UDP, o TCP é orientado a conexão, e não possui o conceito de envoltório (como é o datagrama para o UDP), mas sim fluxo contínuo de octetos (*bytes*). O TCP assume a utilização do protocolo de rede IP e, assim como o UDP, possui portas de origem e destino no seu cabeçalho para permitir multiplexação.

Por ser orientado a conexão, o TCP exige uma troca de mensagens inicial para estabelecer a conexão antes de qualquer conversa. Essa troca de mensagens é feita em três passos, por isso é conhecida como *three-way handshake*.

Para alcançar a posição de protocolo confiável, o TCP garante:

- Recepção dos dados na mesma ordem do envio, garantida com a utilização de número de sequência;
- Retransmissão de dados perdidos fazendo uso de confirmações e *timeouts*;
- Descarte de pacotes duplicados;
- Descarte e retransmissão de dados danificados - danos são detectados através de um *checksum* para cada pacote;
- Controle de fluxo através de uma janela deslizante. A janela diminui a cada envio de dados, e aumenta a cada confirmação;
- Controle de congestionamento.

O TCP se encaixa perfeitamente bem em uma arquitetura cliente-servidor, em que um elemento central aguarda uma conexão (o servidor), enquanto um elemento periférico se conecta a ele para trocar informações, tipicamente utilizando um modelo requisição-resposta. Protocolos como o HTTP e o FTP são amplamente utilizados na internet hoje, e são exemplos de protocolos que fazem uso do serviço confiável que o TCP oferece.

Porém, segundo (FOROUZAN, 2008), o protocolo não é adequado para tráfego em tempo real, pois solicita retransmissão de pacotes. Além disso, o TCP é um

protocolo ponto a ponto que estabelece uma conexão entre dois pontos, o que não é interessante para distribuição em *multicast* (STALLINGS, 2007).

Em aplicações multimídia, a característica confiável do TCP o torna atrativo para implementação de protocolos de controle. Entretanto, em sistemas distribuídos de larga escala, a manutenção da conexão que o TCP estabelece pode representar um desafio, ainda mais quando não existe um elemento central (servidor).

2.2 Estratégias para distribuição de conteúdo

Existem duas tecnologias de transmissão em uso disseminado nos dias de hoje (TANENBAUM, 2003):

- Links ponto a ponto;
- Links de difusão.

O *unicast*, tecnologia de links ponto a ponto, será detalhada na seção 2.2.1, enquanto a tecnologia de links de difusão *multicast* será detalhada na seção 2.2.2.

No contexto de videoconferência, dois termos são amplamente empregados: comunicação ponto a ponto e comunicação multiponto. Utiliza-se o termo ponto a ponto (do inglês, *Peer-to-peer*, ou P2P) para caracterizar uma comunicação entre um par de terminais. Uma videochamada entre duas pessoas ou dois locais é um exemplo de comunicação P2P, como mostra a figura 2.1.

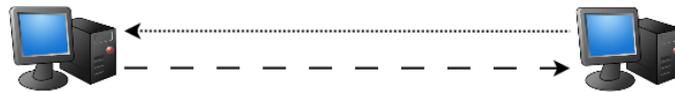


Figura 2.1: Exemplo de sessão ponto a ponto

Quando o número de participantes é maior do que dois, a comunicação é chamada de multiponto (SIMPSON, 2006). Tipicamente tal comunicação acontece com um servidor intermediário que recebe todos os fluxos de dados e distribui o conteúdo dentro da sessão.

Pode-se utilizar também uma comunicação multiponto direta, sem a presença de um servidor central. Entretanto, essa solução perde em escalabilidade pois cada participante deve enviar $n - 1$ fluxos de dados, sendo n o número de participantes; e aumenta a complexidade do sistema, uma vez que cada participante deve saber *a priori* o endereço IP dos demais.

A figura 2.2 ilustra as duas configurações de multiponto apresentadas. As flechas apresentam o fluxo de dados de apenas um participante. Em ambos os cenários, cada participante recebe três fluxos de dados. Porém, no cenário da esquerda, cada participante envia apenas um fluxo de dados, enquanto no cenário da direita, cada participante envia três.

2.2.1 Unicast

Unicast é o termo utilizado para caracterizar a comunicação entre uma origem e um destino (um para um) (FOROUZAN, 2008). Nesse tipo de comunicação, tanto a origem quanto o destino são identificados através de um endereço IP *unicast*. Tal

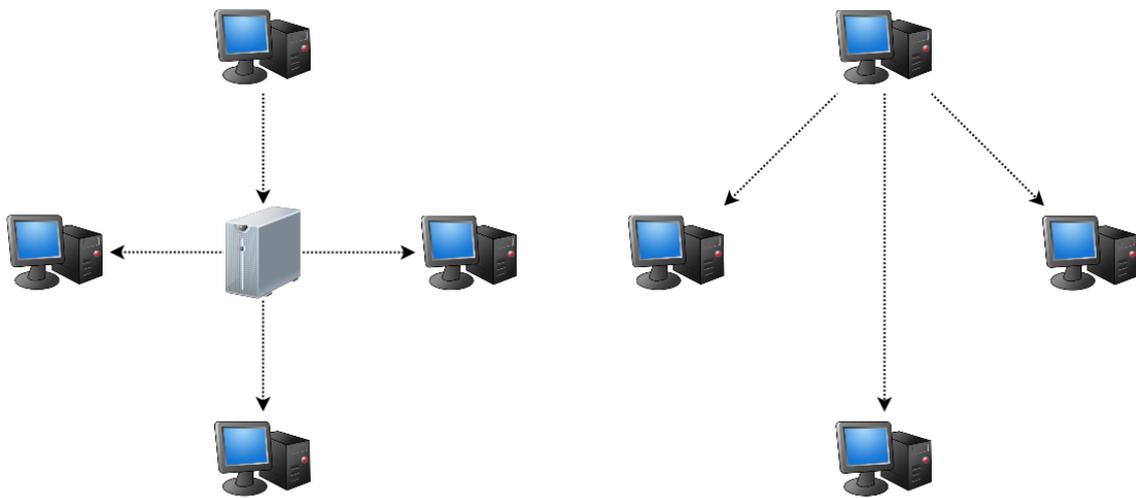


Figura 2.2: Exemplo de sessão multiponto: na esquerda, a presença de um servidor intermediário, e na direita, comunicação multiponto direta

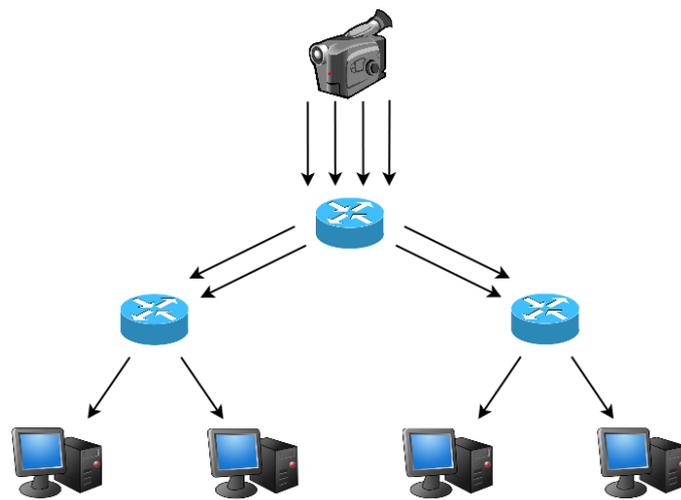


Figura 2.3: Transmissão de dados utilizando *unicast*

comunicação entre pares pode ser orientada a conexão (TCP) ou não orientada a conexão (UDP).

A comunicação em *unicast* é comumente utilizada na internet. Acesso a uma página web via HTTP, download de um arquivo usando FTP ou VoIP entre dois pontos são exemplos de aplicações na internet que utilizam *unicast*. Tipicamente as operações entre um par de processos são feitas utilizando tal tecnologia.

Porém, em uma comunicação de um para muitos, como é o caso de uma videoconferência, a utilização do *unicast* é pouco eficiente e não escalável. A figura 2.3 ilustra uma comunicação em *unicast* em que estão presentes um gerador de conteúdo e quatro consumidores. Do gerador de conteúdo partem quatro setas indicando fluxos de dados, uma para cada consumidor. Ao utilizar o *unicast*, o processamento na geração de réplicas e a utilização da largura de banda de saída no gerador de conteúdo crescem linearmente com o número de consumidores.

Além disso, no mesmo contexto, a aplicação geradora de conteúdo deve possuir um mecanismo de gerenciamento de usuários. Esse mecanismo deve prever operações como adição e remoção de usuários. Nesse sentido, pode-se utilizar um mecanismo estático, ou seja, os endereços IP e portas são configurados *a priori* no gerador de conteúdo, ou então dinâmico, em que consumidores podem entrar e sair no momento que desejarem.

2.2.2 *Multicast*

Multicast é o termo empregado para descrever uma comunicação de uma origem para um grupo de destinos (um para muitos). A origem é identificada através de um endereço *unicast*, e o grupo de destino recebe um endereço de grupo, ou um endereço *multicast* (FOROUZAN, 2008).

Também conhecido como transmissão por difusão seletiva, surgiu para reduzir os problemas existentes na comunicação em grupo. Com o *multicast*, os dados são enviados apenas uma vez para um endereço de grupo, e os roteadores intermediários se encarregam de fazer a entrega a todos os destinatários inscritos no grupo.

No *unicast*, como explicado na seção 2.2.1, a fonte deve enviar os dados para cada participante da sessão individualmente. Essa tecnologia ponto a ponto funciona perfeitamente bem na internet, tanto para acesso a páginas web no modelo cliente-servidor quanto para VoIP entre dois pontos, mas claramente essa solução não é escalável para um grande número de participantes. O *multicast* é a estratégia mais eficiente de entrega se comparada ao *unicast* e ao *broadcast*, pois é a que gera menos tráfego na rede (COMER, 2007).

A característica de escalabilidade pode ser observada na figura 2.4. Apenas um fluxo de dados é transmitido pela origem, e os nós intermediários gerenciam a transmissão.

Além de gerar um menor tráfego na rede, o *multicast* torna facilitado o gerenciamento de grupo. Os usuários que desejam receber o tráfego do grupo fazem uma requisição de entrada em um determinado endereço de grupo (diretiva JOIN), e mais tarde, para finalizar a participação, fazem uma nova requisição, dessa vez para sair do grupo (diretiva LEAVE). Com isso, o número de usuários é uma propriedade transparente para todos que fazem parte do grupo.

Um exemplo de implantação bem sucedida de rede de alta velocidade com suporte a *multicast* é a Rede Ipê. Operada pela Rede Nacional de Ensino e Pesquisa (RNP), a Rede Ipê conecta cerca de 600 instituições públicas e privadas em todo o Brasil e

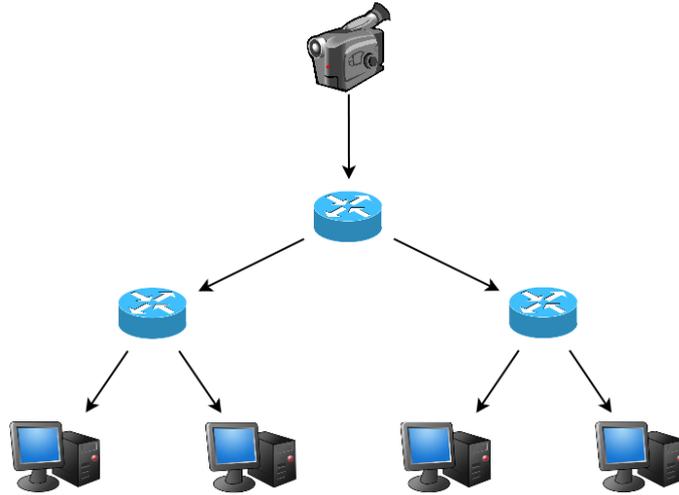


Figura 2.4: Transmissão de dados utilizando *multicast*

possui suporte a *multicast* nativo desde 2004 (Rede Nacional de Ensino e Pesquisa, 2009).

Na internet, o potencial do *multicast* é pouco explorado devido à heterogeneidade dos roteadores que a formam. Somente uma pequena fração de roteadores na internet são configurados para suportar difusão seletiva (FOROUZAN, 2008), e um roteador sem suporte é suficiente para impossibilitar um fluxo de dados em *multicast*.

2.2.3 *Unicast* com servidores intermediários

Como alternativa para o problema de escalabilidade do *unicast* com apenas uma fonte, pode-se utilizar uma rede de servidores intermediários, conforme a figura 2.5. Nessa configuração, um servidor recebe um fluxo de dados de outro servidor e o replica para n destinos, que podem ser tanto servidores quanto clientes finais.

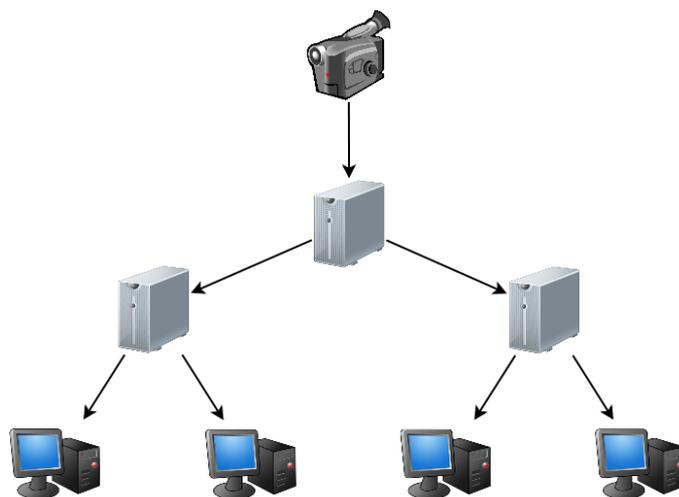


Figura 2.5: Transmissão de dados utilizando *unicast* com servidores intermediários

Apesar de não ser tão eficiente quanto o *multicast*, pois o fluxo de saída é n vezes o fluxo de entrada, a utilização de servidores *unicast* intermediários agrega maior escalabilidade ao modelo *unicast* puro. Sendo s o número de servidores, b o *bitrate*

do fluxo de dados e c a capacidade da rede, o número máximo de clientes n é definido por $\frac{s \times c}{b} - s + 1$.

Além disso, segundo a definição de árvore k -ária (STORER, 2001) e sendo $k = \frac{s \times c}{b}$, a quantidade de níveis da árvore de servidores intermediários será no mínimo

$$\lceil \log_k(k - 1) + \log_k(n + s) \rceil.$$

A quantidade de níveis da árvore de distribuição é uma propriedade importante, pois tem relação direta com o atraso fim a fim do fluxo de dados.

Porém, mesmo com maior escalabilidade a estratégia tem problemas, como a seleção da localização geográfica dos servidores intermediários e o método de controle. Em uma comunicação de tempo real visa-se sempre reduzir o atraso fim a fim do fluxo de dados, e dependendo da localização dos servidores e da maneira como eles são conectados, esse atraso pode aumentar a ponto de deteriorar a experiência do usuário em uma dada sessão.

Através da figura 3.1 é possível destacar três das funções mais significativas do RTP:

- *Sequence number*

O número de sequência em cada pacote é importante para que o receptor consiga reconstruir a ordem de envio dos pacotes - fundamental se a camada inferior não o fizer. Além disso, o número de sequência permite saber a quantidade de pacotes perdidos na rede, se for utilizado um meio não confiável, para que se possa monitorar a qualidade da recepção.

- *Timestamp*

A marca de tempo (ou *timestamp*) é usada para sincronização da transmissão. Isso é necessário pois o atraso da rede é variável de pacote para pacote, sendo tarefa do receptor ditar o ritmo de utilização destes, com base no ritmo de marcação dos pacotes (*timestamp*) na origem. Pode ser necessário também sincronizar mais de um fluxo de dados, por exemplo, áudio e vídeo, e tal sincronização é feita usando esse campo.

- *Synchronization Source (SSRC) Identifier*

Cada *stream* da sessão possui um identificador único randômico. Através desse identificador, o receptor pode agrupar as mensagens provenientes da mesma *stream*. No caso do receptor receber um fluxo de áudio e um de vídeo, é possível distingui-los e realizar o tratamento correto de cada um independentemente.

Além dessas funções, a figura 3.1 possui ainda um campo de utilização bastante específico, o *Contributing Source (CSRC) Identifiers*. O RTP fornece, através desse campo, um mecanismo de mixagem de conteúdo. Usemos como exemplo uma transmissão de vídeo e áudio em uma rede *multicast*. É possível introduzir à sessão uma entidade especial chamada mixer, que receberá os fluxos de dados dos demais participantes, mixará vídeo (em um estilo vários vídeos em uma tela) e áudio, e enviará, digamos, em *unicast*, para participantes que não tem uma rede de capacidade suficiente para receber tal quantidade de informação em *multicast*. O mixer, em cada pacote RTP recebido e mixado, colocará no cabeçalho a lista de identificadores dos participantes que contribuíram na construção do seu conteúdo.

O campo CC (CSRC Count) dirá, de 0 a 15, quantos participantes serão descritos na lista de contribuintes.

O mixer, entretanto, é só uma das duas entidades especiais que o RTP prevê. A outra é chamada de tradutor, e pode ser explicado facilmente através de um exemplo. Existem duas redes distintas com participantes de uma sessão em *multicast*. Porém, entre essas duas redes existe um firewall que não permite passagem de fluxo *multicast*. O RTP prevê, em situações como essas, que se coloque um tradutor em cada lado do firewall, e entre os tradutores se inicie uma sessão segura, em que o firewall não terá acesso ao conteúdo dos pacotes. Sendo assim, a comunicação entre as duas redes é feita através dos tradutores, e a sessão poderá ocorrer normalmente, mesmo com um firewall no meio do caminho.

Juntamente com o RTP atua um protocolo de controle chamado RTCP (RTP Control Protocol), que possui quatro funções básicas em uma sessão RTP:

- Informar aos demais participantes um relatório de qualidade de serviço. Isso é possível através do número de sequência dos pacotes RTP, onde é feita uma estimativa sobre a quantidade de pacotes que foram enviados mas não chegaram no destino. Com um relatório de qualidade de serviço enviado por cada participante, um administrador pode acompanhar o andamento da sessão, assim como identificar se um eventual problema é local ou global. Além disso, esse *feedback* pode ser útil também para mecanismos de adaptação.
- Através de um campo do RTCP chamado *canonical name* pode-se manter um registro para cada participante, uma vez que o SSRC pode mudar caso ocorra uma duplicação ou se o participante reiniciar.
- Cada participante pode estimar de forma independente o tamanho da sessão, pois recebe os relatórios RTCP de todos os outros. É importante saber o número de participantes para calcular o intervalo entre cada envio de relatório.
- Uma função opcional do RTCP é a possibilidade de integrar um controle mínimo da sessão, por exemplo, para identificação do participante a fim de exibí-la em uma interface gráfica. Entretanto, isso é útil para aplicações que necessitam de pouco controle sobre os participantes, do contrário será necessário um protocolo de controle de sessão. De qualquer forma, o RTCP é um ótimo meio de enviar dados de controle aos demais participantes.

O intervalo de envio dos relatórios RTCP é um aspecto bastante importante. O RTP foi desenvolvido para ser escalável a um grande número de participantes, e muito embora o número de participantes de uma sessão multimídia possa ser grande, a quantidade de geradores de conteúdo dentre esses participantes é pequeno, normalmente o mesmo do que em uma pequena sessão. Entretanto, os pacotes RTCP são enviados por todos os participantes, fazendo com que a banda ocupada pelo protocolo cresça linearmente com o número de participantes.

A norma define que o RTCP não pode utilizar mais do que 5% do total de tráfego da sessão. Dentre os 5%, $\frac{1}{4}$ da banda consumida deve ser utilizada pelos geradores de conteúdo. Se a proporção de geradores de conteúdo sobre o total de participantes da sessão ultrapassar $\frac{1}{4}$, então os geradores de conteúdo devem utilizar o total de banda reservada para o protocolo.

Uma abordagem diferente também pode ser utilizada, sem separar geradores de conteúdo e consumidores de conteúdo. Dessa forma, todos os participantes calculam e utilizam o mesmo intervalo de envio do RTCP.

O RTP e o RTCP devem utilizar portas diferentes, pois o protocolo depende dos números de porta para identificar o que é dado RTP e o que é controle RTCP. O RTP deve usar uma porta par, e o RTCP deve utilizar a porta ímpar subsequente.

3.2 RTMP

O RTMP (*Real-Time Messaging Protocol*) é um protocolo proprietário desenvolvido pela Adobe Systems para *streaming* multimídia (áudio, vídeo e dados) entre produtos da plataforma Adobe Flash (ADOBE, 2009). Em 2009, a especificação do protocolo foi divulgada pela Adobe para promover a interação de aplicações externas com os seus produtos.

O protocolo foi desenvolvido para atuar sobre qualquer protocolo de transporte que envie fluxos de mensagens. Os campos presentes no cabeçalho das mensagens são: *Timestamp*, *Length* (tamanho do campo de dados), *Type Id* (tipo da mensagem, sendo que um conjunto define mensagens de controle) e *Message Stream Id* (índice da *stream* contido na mensagem, utilizado para demultiplexação de *streams*).

As mensagens, conforme definido acima, são utilizadas somente no nível de aplicação. Depois de pronta, a mensagem passa por um processo de fragmentação, criando os chamados *chunks* (em português, “nacos”), que mais tarde são enviados pela rede. As informações contidas no cabeçalho das mensagens são necessárias na criação de fragmentos. O tamanho dos *chunks* varia entre 128 bytes até 65536 bytes, e cada *chunk* possui um pequeno cabeçalho de tamanho variável.

Com a utilização dos *chunks*, a estratégia é reduzir a carga extra gerada por cabeçalhos de tamanho fixo. O tamanho dos *chunks* é variável e pode ser configurado via mensagem de controle, de forma independente para cada direção.

A sessão é iniciada através de uma conversa inicial entre cliente e servidor, em três passos. Essas mensagens iniciais, ao contrário das demais mensagens do protocolo, possuem tamanho fixo. Informações como versão do protocolo e *timestamp* inicial são trocadas nessa etapa.

3.3 RTSP

O RTSP (*Real-Time Streaming Protocol*) é um protocolo de controle de *stream* de dados com propriedades de tempo real (SCHULZRINNE; RAO; LANPHIER, 1998). O protocolo não é restrito a mídias ao vivo, e é comumente utilizado para comunicação entre cliente e servidor de *stream* na internet. É o caso do serviço YouTube¹ oferecido pelo Google em sua versão para dispositivos móveis, que utiliza RTSP como protocolo de controle.

A motivação para criação do protocolo é que tipicamente o usuário deseja ter um mínimo controle sobre o vídeo que está assistindo (FOROUZAN, 2008). O protocolo serve como uma espécie de controle remoto de um servidor multimídia. Assim como diversos protocolos da camada de aplicação, o RTSP funciona no modelo requisição-resposta, e as mensagens são enviadas pela rede em ASCII.

Tanto o cliente quanto o servidor RTSP podem ser modelados através da máquina de estados, ilustrada na figura 3.2. Ela define os estados que as entidades podem assumir e as mensagens que desencadeiam transição entre os estados.

Através de um exemplo típico de sessão, os principais métodos do protocolo serão demonstrados. O exemplo foi extraído através da ferramenta de análise de tráfego *Wireshark*² durante o acesso a um vídeo do *Youtube Mobile* no *player* de vídeo *VLC Player*³.

OPTIONS

A primeira mensagem parte do cliente, e possui a função de solicitar ao servidor os métodos suportados por ele. Essa mensagem carrega a URL do vídeo solicitado, a versão do protocolo utilizada, um número de sequência e o nome do agente que está fazendo o acesso, nesse caso, o player de vídeo.

¹<http://m.youtube.com> acessado em junho de 2010.

²<http://www.wireshark.org> acessado em junho de 2010.

³<http://www.videolan.org/vlc> acessado em junho de 2010.

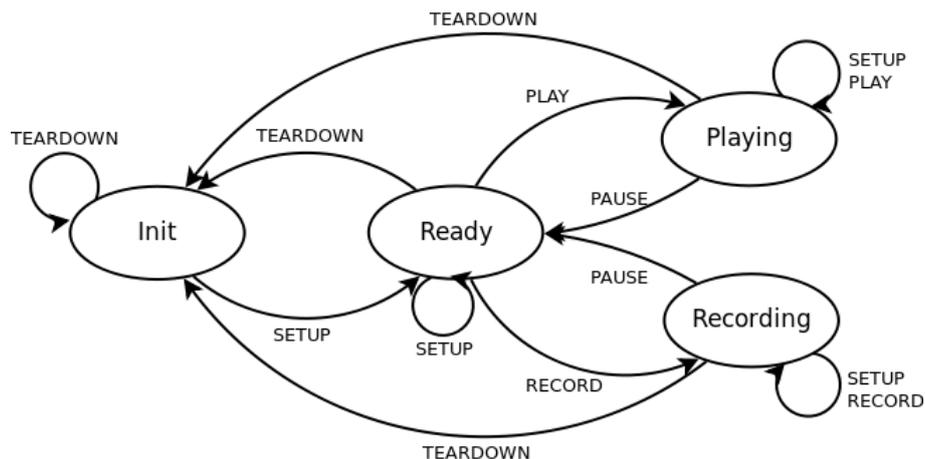


Figura 3.2: Máquina de estados do RTSP

```

1 Request: OPTIONS rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4
2 SHWL84RYRBMYESARFELJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNr
3 VGh1bWJuYWlsYUx7S8dXISww=/0/0/0/video.3gp RTSP/1.0\r\n
4 Method: OPTIONS
5 URL: rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHWL84RY
6 RBMYESARFELJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJu
7 YWlsYUx7S8dXISww=/0/0/0/video.3gp
8 CSeq: 1\r\n
9 User-Agent: VLC media player (LIVE555 Streaming Media v2010.01.07)
10 \r\n
11 \r\n
  
```

RTSP/1.0 200 OK

A resposta do servidor é positiva frente a solicitação - o status 200 é tipicamente utilizado para indicar sucesso. A mensagem carrega a série de métodos suportada pelo servidor, um número de sequência e o nome do servidor, nesse caso, *Google RTSP 1.0*.

```

1 Response: RTSP/1.0 200 OK\r\n
2 Status: 200
3 Public: DESCRIBE, GET_PARAMETER, OPTIONS, PAUSE, PLAY, SETUP,
4 SET_PARAMETER, TEARDOWN\r\n
5 CSeq: 1\r\n
6 Server: Google RTSP 1.0\r\n
7 \r\n
  
```

DESCRIBE

Depois de receber os métodos suportados pelo servidor, o cliente solicita uma descrição da sessão em um protocolo chamado *Session Description Protocol* (SDP) (HANDLEY; JACOBSON, 1998). A descrição é detalhada na mensagem seguinte, e contém informações técnicas necessárias para a sessão ocorrer.

```

1 Request: DESCRIBE rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk
2 4SHWL84RYRBMYESARFELJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNr
3 rVGh1bWJuYWlsYUx7S8dXISww=/0/0/0/video.3gp RTSP/1.0\r\n
  
```

```

4   Method: DESCRIBE
5   URL: rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHWL84RY
6       RBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJl
7       YWlsYUuSxr7S8dXISww=/0/0/0/video.3gp
8   CSeq: 2\r\n
9   Accept: application/sdp\r\n
10  User-Agent: VLC media player (LIVE555 Streaming Media v2010.01.07)
11      \r\n
12  \r\n

```

RTSP/1.0 200 OK

Encapsulada na mensagem positiva do servidor é enviada uma mensagem SDP que descreve as *streams* da sessão multimídia - informações fundamentais para dar início à sessão. Dentre as informações contidas na mensagem, destacam-se o endereço IP do servidor de *stream* (74.125.211.182), a duração da *stream* (361.083000 seg), o protocolo de aplicação que fará a entrega (RTP/AVP⁴97 para vídeo e RTP/AVP 96 para áudio), a dimensão do vídeo (144 x 176), codificadores de áudio e vídeo utilizados (MP4V-ES para vídeo e MP4A-LATM para áudio), entre outros.

Cada *stream* multimídia é descrita individualmente - no exemplo, o vídeo contém uma *stream* de áudio e uma de vídeo -, sendo necessário por isso fazer a requisição das *streams* uma a uma indicando as portas utilizadas para cada sessão RTP, que é independente para cada *stream*. A requisição é feita indicando a URL do vídeo e o índice da *stream* desejada, campo chamado de *trackID*.

A mensagem SDP encapsulada na resposta do servidor pode ser visualizada no apêndice A.

```

1   Response: RTSP/1.0 200 OK\r\n
2       Status: 200
3   Content-type: application/sdp
4   Cache-Control: must-revalidate\r\n
5   Date: Mon, 07 Jun 2010 12:22:32 GMT\r\n
6   Expires: Mon, 07 Jun 2010 12:22:32 GMT\r\n
7   Last-Modified: Mon, 07 Jun 2010 12:22:32 GMT\r\n
8   Content-Base: rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHW
9       L84RYRBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJl
10      uYWlsYUuSxr7S8dXISww=/0/0/0/video.3gp/\r\n
11  CSeq: 2\r\n
12  Server: Google RTSP 1.0\r\n
13  Content-length: 541
14  \r\n
15  Session Description Protocol
16  (...)

```

SETUP

⁴AVP é o acrônimo para *Audio Video Profile*, e é comumente utilizado numa sessão RTP para indicar o perfil a qual pertence uma determinada *stream*, para permitir a identificação e futura exibição no ponto remoto. O perfil representa qual é o padrão de codificação utilizado. No exemplo, RTP/AVP 96 e 97 são perfis definidos como dinâmicos, e por isso podem ser definidos através de outro mecanismo, nesse caso utilizando o SDP (SCHULZRINNE; CASNER, 2003).

Na etapa seguinte, dando início às requisições de envio de *stream*, o cliente solicita a *stream* de *trackID* = 2, no exemplo indicando a *stream* de vídeo. Junto com a requisição é indicado que a transmissão será *unicast*, e é sugerido um par de portas para a sessão RTP.

```

1 Request: SETUP rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SH
2           WL84RYRBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVG
3           h1bWJuYWlsYUuSxr7S8dXISww=/0/0/0/video.3gp/trackID=2
4           RTSP/1.0\r\n
5     Method: SETUP
6     URL: rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHWL84RY
7           RBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJu
8           YWlsYUuSxr7S8dXISww=/0/0/0/video.3gp/trackID=2
9 CSeq: 3\r\n
10 Transport: RTP/AVP; unicast; client_port=1234-1235
11 User-Agent: VLC media player (LIVE555 Streaming Media v2010.01.07)
12           \r\n
13 \r\n

```

RTSP/1.0 200 OK

O servidor atribui àquela sessão um identificador único (2456c233), confirma as portas indicadas pelo cliente (1234-1235) para a sessão solicitada e indica as portas utilizadas por ele (10580-10581). O estado do servidor frente ao *stream* solicitado é configurado para *play*.

```

1 Response: RTSP/1.0 200 OK\r\n
2     Status: 200
3     Session: 2456c233; timeout=90
4     Transport: RTP/AVP; unicast; mode=play; client_port=1234-1235;
5               server_port=10580-10581; source=74.125.211.182; ssrc=2677249C
6     Cache-Control: must-revalidate\r\n
7     Date: Mon, 07 Jun 2010 12:22:32 GMT\r\n
8     Expires: Mon, 07 Jun 2010 12:22:32 GMT\r\n
9     Last-Modified: Mon, 07 Jun 2010 12:22:32 GMT\r\n
10    CSeq: 3\r\n
11    Server: Google RTSP 1.0\r\n
12    \r\n

```

A interação entre cliente e servidor se repete, dessa vez para a *stream* de áudio.

PLAY

Depois da conversa inicial de configuração, o cliente utiliza o método PLAY enviando o identificador que ele recebeu para a sessão e o intervalo de tempo da *stream* que ele deseja receber (instante 0.000 até indeterminado).

```

1 Request: PLAY rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHW
2           L84RYRBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh
3           1bWJuYWlsYUuSxr7S8dXISww=/0/0/0/video.3gp/ RTSP/1.0\r\n
4     Method: PLAY
5     URL: rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHWL84RY
6           RBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJu
7           YWlsYUuSxr7S8dXISww=/0/0/0/video.3gp/
8 CSeq: 5\r\n
9 Session: 2456c233

```

```

10 | Range: npt=0.000-\r\n
11 | User-Agent: VLC media player (LIVE555 Streaming Media v2010.01.07)
12 |   \r\n
13 | \r\n

```

RTSP/1.0 200 OK

O servidor confirma o início do envio da *stream*, indicando o intervalo de tempo que será enviado (de 0.000 até 361.083, tamanho total do arquivo) e uma série de informações referentes ao RTP (número de sequência utilizado, *timestamp*, entre outros). Nesse momento, o agente do cliente inicia a exibição de áudio e vídeo.

```

1 | Response: RTSP/1.0 200 OK\r\n
2 |   Status: 200
3 |   Session: 2456c233;timeout=90
4 |   Range: npt=0.000-361.083\r\n
5 |   [truncated] RTP-Info: url=rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHWL84RYRBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJuYWlsYUx7S8dXISww=/0/0/0/video.3gp/trackID=2;seq=11707;rtptime=1017946954,url=rtsp://v7.cache6.c.you
6 |   saWNrVGh1bWJuYWlsYUx7S8dXISww=/0/0/0/video.3gp/trackID=2;
7 |   seq=11707;rtptime=1017946954,url=rtsp://v7.cache6.c.you
8 |   seq=11707;rtptime=1017946954,url=rtsp://v7.cache6.c.you
9 | CSeq: 5\r\n
10 | Server: Google RTSP 1.0\r\n
11 | \r\n

```

GET_PARAMETER

O cliente, logo após receber o OK descrito acima, envia uma mensagem GET_PARAMETER sem nenhum texto no corpo da mensagem. Esse método tipicamente faz requisição de parâmetros indicados no texto da mensagem (pacotes recebidos, *jitter*). Quando o corpo da mensagem não indica parâmetros, o método serve para testar a comunicação com o servidor (uma espécie de *ping*).

```

1 | Request: GET_PARAMETER rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHWL84RYRBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJuYWlsYUx7S8dXISww=/0/0/0/video.3gp/
2 |   IaPwk4SHWL84RYRBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJuYWlsYUx7S8dXISww=/0/0/0/video.3gp/
3 |   NsaWNrVGh1bWJuYWlsYUx7S8dXISww=/0/0/0/video.3gp/
4 |   RTSP/1.0\r\n
5 |   Method: GET_PARAMETER
6 |   URL: rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHWL84RYRBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJuYWlsYUx7S8dXISww=/0/0/0/video.3gp/
7 |   RBMYESARFEIJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJuYWlsYUx7S8dXISww=/0/0/0/video.3gp/
8 |   YWlsYUx7S8dXISww=/0/0/0/video.3gp/
9 | CSeq: 6\r\n
10 | Session: 2456c233
11 | User-Agent: VLC media player (LIVE555 Streaming Media v2010.01.07)
12 |   \r\n
13 | \r\n

```

RTSP/1.0 200 OK

A requisição do cliente é prontamente atendida com uma resposta positiva do servidor (*I'm alive*).

```

1 | Response: RTSP/1.0 200 OK\r\n
2 |   Status: 200
3 | CSeq: 6\r\n

```

```

4 | Session: 2456c233
5 | Server: Google RTSP 1.0\r\n
6 | Content-length: 0
7 | \r\n

```

PAUSE

Ao utilizar o comando *pause* do agente do cliente, uma mensagem de mesmo nome é enviada ao servidor solicitando uma parada momentânea no envio da *stream*. Junto com essa mensagem, naturalmente, é enviado o identificador da sessão.

```

1 | Request: PAUSE rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SH
2 |         WL84RYRBYESARFELJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVG
3 |         h1bWJuYWlsYUlxSxr7S8dXISww=/0/0/0/video.3gp/ RTSP/1.0\r\n
4 |         Method: PAUSE
5 |         URL: rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHWL84RY
6 |         RBMYESARFELJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJu
7 |         YWlsYUlxSxr7S8dXISww=/0/0/0/video.3gp/
8 | CSeq: 7\r\n
9 | Session: 2456c233
10 | User-Agent: VLC media player (LIVE555 Streaming Media v2010.01.07)
11 | \r\n
12 | \r\n

```

RTSP/1.0 200 OK

O servidor confirma a parada no envio de dados.

```

1 | Response: RTSP/1.0 200 OK\r\n
2 |         Status: 200
3 | CSeq: 7\r\n
4 | Session: 2456c233
5 | Server: Google RTSP 1.0\r\n
6 | \r\n

```

A descontinuidade no número de sequência é devido ao uso duplicado dos comandos PLAY/PAUSE, que foram utilizados para comprovar o correto funcionamento do cliente/servidor.

TEARDOWN

Esse método é utilizado pelo cliente para encerrar a sessão RTSP e aconteceu, no exemplo, ao utilizar o comando *stop* do player de vídeo.

```

1 | Request: TEARDOWN rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk
2 |         4SHWL84RYRBYESARFELJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWN
3 |         rVGh1bWJuYWlsYUlxSxr7S8dXISww=/0/0/0/video.3gp/ RTSP/1.0\r\n
4 |         Method: TEARDOWN
5 |         URL: rtsp://v7.cache6.c.youtube.com/CkgLEny73wIaPwk4SHWL84RY
6 |         RBMYESARFELJbXYtZ29vZ2xlSARSB3Jlc3VsdHNaDkNsaWNrVGh1bWJu
7 |         YWlsYUlxSxr7S8dXISww=/0/0/0/video.3gp/
8 | CSeq: 11\r\n
9 | Session: 2456c233
10 | User-Agent: VLC media player (LIVE555 Streaming Media v2010.01.07)
11 | \r\n
12 | \r\n

```

RTSP/1.0 200 OK

A solicitação do cliente é atendida e a sessão, encerrada.

```

1 Response: RTSP/1.0 200 OK\r\n
2   Status: 200
3 CSeq: 11\r\n
4 Session: 2456c233
5 Server: Google RTSP 1.0\r\n
6 \r\n

```

Além dos métodos descritos no exemplo, o protocolo prevê outros:

- ANNOUNCE

Se enviada pelo cliente, tem a função de descrever uma apresentação identificada por uma URL. Se enviada pelo servidor, tem a função de atualizar a descrição da sessão em tempo real.
- RECORD

Inicia a gravação de uma determinada sessão.
- REDIRECT

Método utilizado pelo servidor para indicar que o cliente deve se conectar a um outro servidor.
- SET_PARAMETER

Tem a função de atualizar um parâmetro da sessão no sentido cliente -> servidor. Parâmetros de transporte da *stream* devem ser configurados somente no comando SETUP.

Conforme visto no exemplo acima, o RTSP é exclusivamente um protocolo de controle. O transporte de dados em si deve ser feito através de um outro protocolo, tipicamente o RTP ou o RDT.

Os diversos métodos previstos pelo protocolo podem ser classificados quanto à obrigatoriedade de implementação em Obrigatório, Recomendado e Opcional. Além disso, os métodos possuem classificação diferente dependendo da entidade (cliente ou servidor) (SCHULZRINNE; RAO; LANPHIER, 1998).

- Implementação do servidor
 - Obrigatório: OPTIONS, PLAY, SETUP, TEARDOWN
 - Recomendado: DESCRIBE, PAUSE
 - Opcional: ANNOUNCE, GET_PARAMETER, RECORD, SET_PARAMETER
- Implementação do cliente
 - Obrigatório: -
 - Recomendado: -
 - Opcional: ANNOUNCE, GET_PARAMETER, OPTIONS, REDIRECT, SET_PARAMETER

3.4 RDT

O RDT é um protocolo proprietário de transporte desenvolvido pela *RealNetworks* nos anos 90 para transmissão de áudio e vídeo. É tipicamente utilizado juntamente com um protocolo de controle de *stream*, tal como o RTSP, e é uma alternativa ao RTP (HELIX, 2010).

O RDT hoje faz parte do projeto Helix Community.

4 PARAMETRIZAÇÃO DA CODIFICAÇÃO DE VÍDEO

Dados multimídia podem ter funções diferentes em um sistema computacional. Tanto para armazenamento quanto para transmissão via rede, normalmente áudio e vídeo são codificados com o intuito de reduzir a quantidade de dados. Em se tratando de transmissão via rede, a parametrização da codificação é fortemente dependente do perfil de uso.

Por exemplo, em uma sessão de videoconferência em *multicast*, os parâmetros de codificação devem ser ajustados de tal forma que todos os participantes tenham poder de processamento suficiente para decodificar os dados. Mais importante do que isso, vídeo e áudio devem ser suficientemente compactados para que todos os participantes obtenham uma qualidade de recepção satisfatória.

Em uma sessão *unicast*, em que cada par de participantes possui tipicamente um cenário diferente quanto à largura de banda, pode-se utilizar uma abordagem adaptativa (HALSALL, 2001). Nesse capítulo serão abordados os parâmetros de codificação que podem ser explorados para adaptação, bem como mecanismos para realizá-la.

4.1 Biblioteca de codificação

Em uma transmissão de áudio e vídeo normalmente são utilizados dados comprimidos. A compressão tem a função fundamental de reduzir a quantidade de dados na rede, e pode ser personalizada conforme a disponibilidade de processamento e largura de banda.

O processo de codificação tipicamente introduz erros, ou seja, há perda de qualidade. A deterioração do vídeo ou áudio depende do perfil de codificação utilizado. Existem ainda codificadores que não geram perdas de qualidade, porém são menos eficazes quanto à compressão.

Codificação e decodificação de vídeo podem ser realizadas em software ou hardware. Em geral, o recurso de hardware específico não está presente em um PC, mas está disponível para realizar esta tarefa em dispositivos móveis como telefones celulares, *smartphones* e *tablets*.

Em se tratando de software, uma biblioteca bastante utilizada é a “FFmpeg”, que possui código aberto e implementação para grande parte dos principais codificadores e decodificadores de áudio e vídeo existentes. Além disso, a biblioteca oferece recursos de pós-processamento e aplicação de filtros (FFmpeg, 2010).

4.2 Parâmetros adaptativos

Com o objetivo de gerar *streams* de vídeo e áudio otimizadas para cada grupo de clientes, uma série de parâmetros pode ser configurada na biblioteca de codificação. Dentre os parâmetros ajustáveis, um deles é o codificador. Porém, realizar uma comparação entre codificadores não é o objetivo deste trabalho, que se limita a ajustar parâmetros como *bitrate*, resolução e FPS. Tais parâmetros serão melhor explicados nas subseções abaixo.

Embora uma sessão multimídia seja composta de *streams* de vídeo e áudio, a adaptação da codificação de áudio não oferece uma grande oportunidade. Segundo a tabela contida na figura 4.1, o conteúdo de um DVD é de aproximadamente 8% de áudio, ou seja, apenas uma pequena parcela se comparada à quantidade de vídeo.

	VCD	SVCD	DVD	HDDVD HDTV (WMVHD)	AVI DivX XviD WMV	MOV Quick- Time	RM Real- Media	AVI DV
Resolution	352x240 NTSC/PAL	480x480 480x576	720x480 ² 720x576 ²	1920x1080 ² 1280x720 ²	640x480 ²	640x480 ²	320x240 ²	720x480 720x576
Video Compression	MPEG1	MPEG2	MPEG2, MPEG1	MPEG2 (WMV- MPEG4)	MPEG4	Sorenson, Cinepak, MPEG4 ...	RM	DV
Video bitrate	1150kbps	~2000kbps	~5000kbps	~20Mbps (~8Mbps)	~1000kbps	~1000kbps	~350kbps	25Mbps
Audio Compression	MP1	MP1	MP1, MP2, AC3, DTS, PCM	MP1, MP2, AC3, DTS, PCM	MP3, WMA, OGG, AAC, AC3	QDesign Music, MP3 ...	RM	DV
Audio bitrate	224kbps	~224kbps	~448kbps	~448kbps	~128kbps	~128kbps	~64kbps	~1500kbps
Size/min	10 MB/min	10-20 MB/min	30-70 MB/min	~150MB/min (~60MB/min)	4-10 MB/min	4-20 MB/min	2-5 MB/min	216MB/min
Min/74min CD	74min	35-60min	10-20min	~4min (~10min)	60-180min	30-180 min	120-300 min	3min
Hours/DVD	N/A	N/A	1-2hrs (2-5hrs [®])	~30min (~1hrs)	7-18hrs	3-18hrs	14-35hrs	20min
Hours/ DualLayerDVD	N/A	N/A	2-4hrs (5-9hrs [®])	~55min (~2hrs)	13-30hrs	6-30hrs	25-65hrs	37min
DVD Player Compatibility	Great	Good	Excellent	None	Few	None	None	None
Computer CPU Usage	Low	High	Very High	Super high	Very High	High	Low	High
Quality	Good	Great*	Excellent*	Superb*	Great*	Great*	Decent*	Excellent

Figura 4.1: Comparação entre formatos de vídeo (ViDEOHelp.com, 2010)

4.2.1 Bitrate

O *bitrate*, ou taxa de transferência de bits, é um parâmetro crítico com relação à ocupação do link de rede, e reflete diretamente na qualidade percebida pelo usuário. Quanto menor for o *bitrate*, menor é a qualidade obtida e menor é a ocupação do link. Além disso, quanto menor for o *bitrate* escolhido para codificação, maior é a demanda por processamento para realizar a operação.

A oportunidade de adaptação do *bitrate* se deve à grande diversidade de larguras de banda presentes na internet. Dentro da Rede Ipê, as conexões variam entre 6 Mbps e 10 Gbps¹. Em redes comerciais de banda larga, como a GVT, os serviços disponíveis variam entre 3 Mbps e 100 Mbps². Já em redes 3G, que atingem pessoas que acessam a internet via *smartphones* e *netbooks*, a velocidade máxima atingida

¹<http://www.rnp.br/backbone/index.php> acessado em junho de 2010.

²<http://www.gvt.com.br> acessado em junho de 2010

é 1 Mbps³⁴⁵.

4.2.2 Resolução

A resolução representa o par altura/largura de uma imagem. Ao expandir o conceito para vídeo, a resolução é o tamanho de cada imagem que o compõe. Em geral, a unidade de medida é o pixel, ou ponto. Algumas resoluções bastante conhecidas para vídeo são:

- SD (720x480)
- HD (1280x720)
- Full-HD (1920x1080)

Dentre os diversos dispositivos móveis disponíveis no mercado hoje, podemos citar as seguintes resoluções de tela:

- CGA (320x240 - Nokia N95)
- HVGA (480x320 - HTC Magic/Apple iPhone 3GS)
- WVGA (800x480 - HTC Desire)
- Double HVGA (960x640 - Apple iPhone 4)
- UVGA (1024x768 - Apple iPad)

O anexo A apresenta um diagrama com as resoluções mais conhecidas.

Considerando a diversidade de dispositivos e, principalmente, de tamanhos distintos de telas, é evidente que um vídeo não pode possuir resolução ótima para qualquer dispositivo. Peguemos o exemplo em que um vídeo deve ser exibido em uma TV Full-HD de 42" e, também, em um *smartphone* com resolução HVGA.

Se o vídeo possuir a resolução exata da TV, o *smartphone* gastará muito processamento para decodificar o vídeo exageradamente grande, terá que redimensioná-lo antes da exibição, além de, em se tratando de *stream* via rede, gastar uma grande quantidade de banda de forma desnecessária. Já se o vídeo possuir a resolução ideal para o *smartphone*, a TV exibirá o vídeo com grande facilidade, mas a qualidade percebida por um usuário ao assistir o vídeo será bastante prejudicada, pois cada pixel do vídeo se transformará em um grande bloco na tela.

Quanto maior é a resolução, maior é o tamanho da *stream* na rede. Por se tratar de duas dimensões de pixels, a relação é quadrática, ou seja, se um vídeo tiver sua altura e largura reduzidos à metade, a ocupação da rede cairá quatro vezes. Em vista disso, a resolução é um parâmetro de fundamental importância para adaptação.

³<http://www.vivo.com.br> acessado em junho de 2010.

⁴<http://www.oi.com.br> acessado em junho de 2010.

⁵<http://www.claro.com.br> acessado em junho de 2010.

4.2.3 FPS

FPS, ou quadros por segundo, é o número de imagens contidas em um segundo de vídeo. O efeito que o FPS tem sobre o vídeo é a suavização da transição de uma imagem para a outra. Quanto maior for o FPS, mais suave será o vídeo e mais natural ele parecerá.

No contexto de adaptação, o FPS possui relação linear com a ocupação da rede e também com a demanda de processamento. Se o FPS de um vídeo for reduzido pela metade, a ocupação da banda também será reduzida pela metade, e o tempo disponível para realizar a decodificação do vídeo será dobrado, ou seja, a carga de processamento também cairá pela metade.

4.3 Mecanismos de adaptação

4.3.1 Múltiplas taxas fixas

O mecanismo de adaptação com múltiplas taxas fixas é um recurso estático. Uma série de perfis são pré-definidos no servidor e incluem todos os parâmetros necessários no processo de codificação.

Para cada *stream* recebida são geradas novas streams conforme os perfis pré-definidos. Algum mecanismo é utilizado para selecionar qual é o perfil mais adequado a um determinado cliente.

4.3.2 Adaptação dinâmica

Utilizando adaptação dinâmica, não existem perfis pré-definidos. Os perfis de codificação serão gerados sob demanda, na medida exata para cada cliente. No caso de um perfil já existente ser requisitado, não existe a necessidade de gerar uma nova *stream* réplica da existente - pode-se utilizar a mesma.

5 PROPOSTA DE SOLUÇÃO

Conforme introduzido no início deste trabalho, o problema a ser resolvido é a distribuição de vídeo em redes híbridas. Sabe-se que, de acordo com a discussão da seção 2.2.2, o *multicast* é a melhor estratégia de entrega, pois é a que gera menor tráfego na rede. Porém, tal estratégia não pode ser utilizada quando nem todos os participantes da sessão possuem suporte a *multicast*.

5.1 Arquitetura mista

Sendo assim, e conforme a seção 2.2, este trabalho propõe uma estratégia mista. Aproveitando o melhor das estratégias apresentadas, a solução é a utilização do *multicast* pelos clientes que o alcançam e a utilização de servidores de *stream unicast* que atendam os demais clientes conforme o perfil de cada um.

O problema pode ser subdividido em problemas menores, como segue:

- Teste de alcance do *multicast*;
- Teste e seleção do servidor mais adequado;
- Solicitação de início e fim de transmissão;
- Adaptação.

5.1.1 Teste de alcance do *multicast*

A solução propõe dois perfis de clientes: os clientes que alcançam *multicast* e os que não alcançam. Sendo assim, a primeira tarefa é detectar se o cliente é atingível pelo fluxo em *multicast*.

Para realizar tal operação, o cliente pode abrir uma conexão de espera UDP *multicast* e aguardar um pacote chegar. Entretanto, testes de laboratório indicaram que o tempo para que a árvore de distribuição *multicast* seja montada é bastante variável, podendo atingir até um minuto. Por isso, a solução é esperar por um curto intervalo de tempo e, na ausência de um tráfego *multicast*, solicitar uma *stream unicast* - entretanto, caso o tráfego *multicast* inicie depois de um tempo de espera, a *stream unicast* é cancelada.

Porém, para que essa solução funcione é fundamental que o gerador de conteúdo envie um fluxo constante de dados em *multicast*. Dessa forma, todos os clientes têm condição de realizar o teste.

5.1.2 Teste e seleção do servidor mais adequado

Suponha que em uma sessão multimídia, por exemplo, uma videoconferência, existe um servidor que possui conhecimento global da sessão, ou seja, sabe quem são os clientes e quem são os servidores de *stream unicast*. Quando um cliente solicita uma *stream unicast*, o servidor deve estar apto a selecionar o servidor de *stream* mais adequado.

A fim de selecionar corretamente o servidor de *stream* mais adequado, o servidor pode enviar uma lista para o cliente que fez a requisição com os endereços de alguns servidores de *stream* conectados na sessão, e o cliente, através desses endereços, pode realizar testes sobre cada um. Um teste bastante simples e eficaz é o envio de um *Echo Request* (PING) para cada servidor a fim de contabilizar os valores de RTT e número de saltos. Ao devolver esses dados ao servidor principal, o mesmo pode escolher o servidor geograficamente mais próximo ou que possua o menor RTT, por exemplo.

A existência de um servidor principal da sessão é importante pelo conhecimento global do ambiente. Com ele é possível aplicar o método descrito acima, além de possuir maior controle sobre todos os participantes da sessão.

5.1.3 Solicitação de início e fim de transmissão

De posse do endereço de um servidor de *stream unicast*, é possível trocar mensagens solicitando o início ou o fim de uma transmissão de forma direta ou ainda indiretamente, através do servidor principal da sessão. Todas as mensagens devem possuir garantia de entrega, pois uma vez fora da sessão, um participante não gostaria de ser importunado com dados provenientes da mesma, ou seja, ao sair da sessão, o servidor de *stream unicast* que o atendia deve encerrar a transmissão.

5.1.4 Adaptação

Após o início de uma transmissão, o cliente e o servidor de *stream unicast* entram em modo de adaptação. O cliente repassa algumas estatísticas da transmissão, como por exemplo a taxa de perda de pacotes e o atraso médio. Com base nas estatísticas recebidas, o servidor estima um perfil de vídeo mais adequado e modifica o fluxo atual. O objetivo desse processo é otimizar a qualidade de vídeo percebida pelo usuário.

5.2 Hierarquização

Pode-se imaginar basicamente duas formas de utilização de servidores de *stream* conforme descrito acima. Uma restrição pode ser inserida para que os fluxos de entrada do servidor sejam somente fluxos *multicast*. Dessa forma, o sistema terá um único nível de servidores.

Entretanto, essa restrição pode ser relaxada, possibilitando que um servidor de *stream* receba fluxos de um outro servidor de *stream*, tornando a organização dos servidores hierárquica. Utilizando uma organização hierárquica ganha-se em alcançabilidade, mas perde-se com o aumento do atraso e da complexidade.

Imagine duas redes distintas separadas por uma conexão de 1 Mbps. Na rede 1 está o gerador de conteúdo, com uma taxa de áudio e vídeo de 1 Mbps, e a distribuição dentro da rede acontece em *multicast*. Porém, participantes da rede 2

desejam entrar na sessão. Se o sistema não for hierarquizado, apenas um participante da rede 2 poderá receber o fluxo de dados da sessão, estrangulando o link. Com hierarquização, pode-se imaginar dois servidores de *stream*, um na rede 1 e outro na rede 2. O servidor da rede 1 encaminha o fluxo de dados para o servidor da rede 2, que por sua vez distribui o conteúdo para todos os participantes.

No exemplo acima faz-se uso de hierarquização para aumentar a alcançabilidade do sistema. Entretanto, há um aumento de complexidade considerável na interação entre as entidades, uma vez que o sistema deve se organizar automaticamente para que o link seja ocupado pelo fluxo de dados entre os servidores de *stream*, e não entre um servidor e um participante. Nesse caso, pode-se inserir uma restrição para garantir que os servidores de *stream* que alcançam o fluxo em *multicast* só encaminhem esse fluxo para outros servidores, e os servidores sem acesso ao fluxo em *multicast* atendam os demais participantes.

O aumento do atraso fim a fim se deve ao aumento de entidades intermediárias. Quanto maior for a altura da árvore hierárquica, maior será o atraso do sistema. Vale lembrar que o limite máximo para atraso em uma aplicação interativa de áudio e vídeo é 400 milissegundos, e tal limite deve ser levado em conta no momento de organizar hierarquicamente as entidades.

5.3 Métricas de avaliação e testes

Antes de propor um ambiente para validação é necessário apresentar as métricas que serão utilizadas para avaliação da solução.

5.3.1 Atraso

O atraso é uma métrica crítica de avaliação, pois segundo (KUROSE; ROSS, 2007), em uma aplicação de áudio e vídeo interativa, o atraso fim a fim pode variar entre 150 e 400 ms. A medição do atraso é realizada a partir da geração do conteúdo até a exibição no ponto remoto. Atrasos maiores do que 400 ms podem gerar dificuldades de comunicação, e até tornar a interação ininteligível.

A medição do atraso para a avaliação do modelo terá como objetivo comparar o atraso percebido pelos clientes *multicast* em relação ao atraso dos clientes *unicast* com vídeo adaptado. Espera-se que o maior atraso registrado seja inferior ao limite citado acima.

Para realizar a medição do atraso, o gerador de conteúdo terá uma câmera filmadora apontada para um cronômetro, e a filmagem será lançada ao sistema como conteúdo da sessão. Ao seu lado, um cliente *multicast* e clientes *unicast* com vídeo adaptado estarão recebendo o conteúdo da sessão, e desse cenário serão tiradas fotografias, a fim de:

1. capturar a imagem do cronômetro original;
2. capturar as imagens do cronômetro sendo exibidas nas telas dos clientes.

Comparando o cronômetro original e o cronômetro exibido nos pontos remotos é possível estimar o atraso do sistema. Dessa forma será feita uma avaliação e comparação das medições realizadas.

5.3.2 Desempenho

O servidor de *stream unicast*, entidade proposta neste trabalho, adapta o fluxo de vídeo para clientes *unicast* através de um processo de transcodificação. Conforme explicado anteriormente, o processo de codificação demanda um processamento considerável. Por isso, essa nova entidade deve passar por uma avaliação de desempenho para que se possa estimar a quantidade de *streams* que um servidor consegue gerar a partir de uma *stream* base.

Para isso será avaliada a demanda de processamento necessária para gerar diferentes perfis de vídeo. Com base nesse dado será estimada a quantidade de perfis diferentes que um servidor consegue gerar.

5.3.3 Saturação

Além da capacidade de processamento, não se pode esperar que a largura de banda de saída do servidor de *stream* seja ilimitada. A saturação diz respeito à ocupação de link do servidor, e dessa métrica também depende o número de participantes que ele pode atender.

Entretanto, a saturação deve ser levada em conta no momento de aceitar um cliente *unicast*. Uma vez que o servidor já está saturado, novos clientes devem ser recusados. Um indicador possível de saturação é a taxa perda de pacotes, tanto na entrada como na saída.

5.3.4 Alcançabilidade

Essa é a métrica mais importante e a principal justificativa para o uso da arquitetura mista no lugar do *multicast* puro. O ganho com alcançabilidade pode ser medido ao determinar a quantidade de participantes capazes de acompanhar uma sessão em uma rede heterogênea.

6 ARQUITETURA DE VALIDAÇÃO

Sendo o objetivo deste trabalho desenvolver uma solução prática, um software foi selecionado para servir como base do desenvolvimento. Tal software e suas características serão exploradas nas seções seguintes.

6.1 Estudo de caso - IVA

O IVA (Sistema Interativo de Áudio e Vídeo) é um sistema desenvolvido pelo Laboratório do PRAV - UFRGS e consiste em uma ferramenta para Ensino a Distância com suporte a transmissão de áudio e vídeo em alta qualidade (ROESLER; HUSEMANN; COSTA, 2009) (MACHADO; ROESLER; HUSEMANN, 2008) (Projetos em Áudio e Vídeo, 2010). A escolha da ferramenta é justificada pela tecnologia de transmissão que ela utiliza nativamente: o *multicast* (detalhada na seção 2.2.2).

6.1.1 Apresentador

O Apresentador é a entidade cliente do IVA. Ele possui os recursos de produção e exibição de conteúdo multimídia, além de oferecer possibilidade de interação através de mensagens de texto e pedido de atenção (expressar vontade de interação por áudio e vídeo).

Tipicamente o professor, ao realizar uma apresentação/aula, utiliza o Apresentador para permitir que os alunos ou salas remotas, que por sua vez também utilizam o Apresentador, recebam áudio e vídeo para acompanhar a exposição. Os clientes remotos também enviam áudio e vídeo para permitir interação com o professor e com outros clientes remotos.



Figura 6.1: Interface do Apresentador IVA na tela de seleção de canal

Para entrar em uma sessão, o Apresentador deve ter conhecimento do endereço

IP do Moderador (detalhado na subseção 6.1.2). A figura 6.1 apresenta a interface do Apresentador para seleção de um canal - cada canal representa um Moderador diferente.

A figura 6.2 mostra a interface do Apresentador dentro de uma sessão. Na parte inferior estão o controle de volume do microfone, a pequena área de visualização da própria câmera, a área destinada para mensagens de chat, e na barra estão os botões de exibição da área de chat, de pedido de atenção, habilitação de microfone e câmera e controle de volume. A grande área azul destina-se à exibição do vídeo principal da sessão.

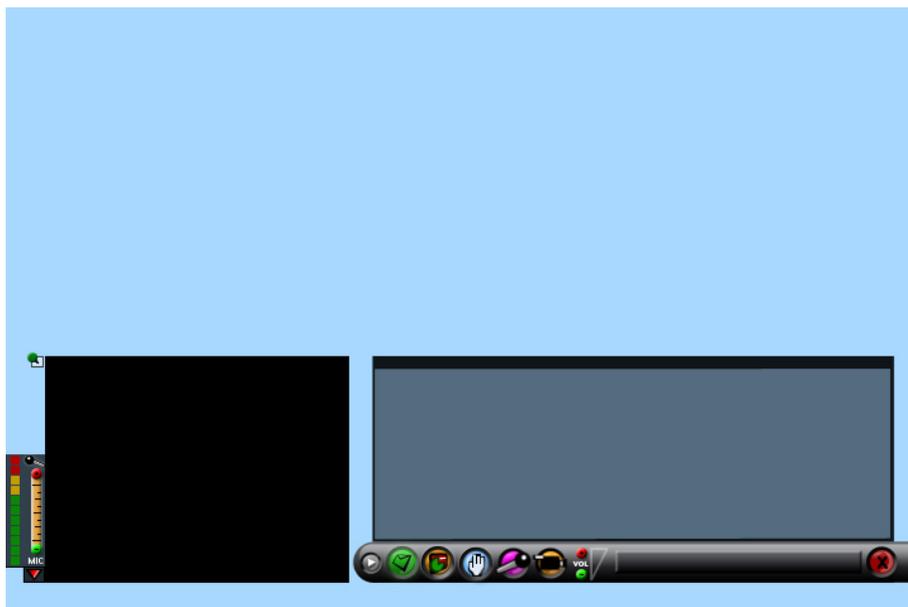


Figura 6.2: Interface do Apresentador IVA na tela de sessão

6.1.2 Moderador

O Moderador é a entidade central do IVA, ou servidor, e possui duas funções principais:

- Selecionar os Apresentadores geradores de conteúdo
Quando um Apresentador se conecta à sessão, o Moderador tem o poder de promover tal Apresentador para ser um gerador de conteúdo. O número de geradores de conteúdo de uma sessão é limitado a vinte - mais detalhes serão abordados na subseção 6.1.3. Além disso, o Moderador tem o poder de remover um cliente da sessão.
- Moderar mensagens de texto
As entidades Apresentador e Chat são entidades clientes com o recurso de interação através de mensagens de texto com o Moderador. Tais mensagens podem conter, por exemplo, dúvidas de alunos relacionadas ao conteúdo da aula apresentada na sessão.

No modelo de aula que o IVA propõe, o operador do Moderador aparece como um auxiliar pedagógico que tem conhecimento sobre o assunto da exposição. Além

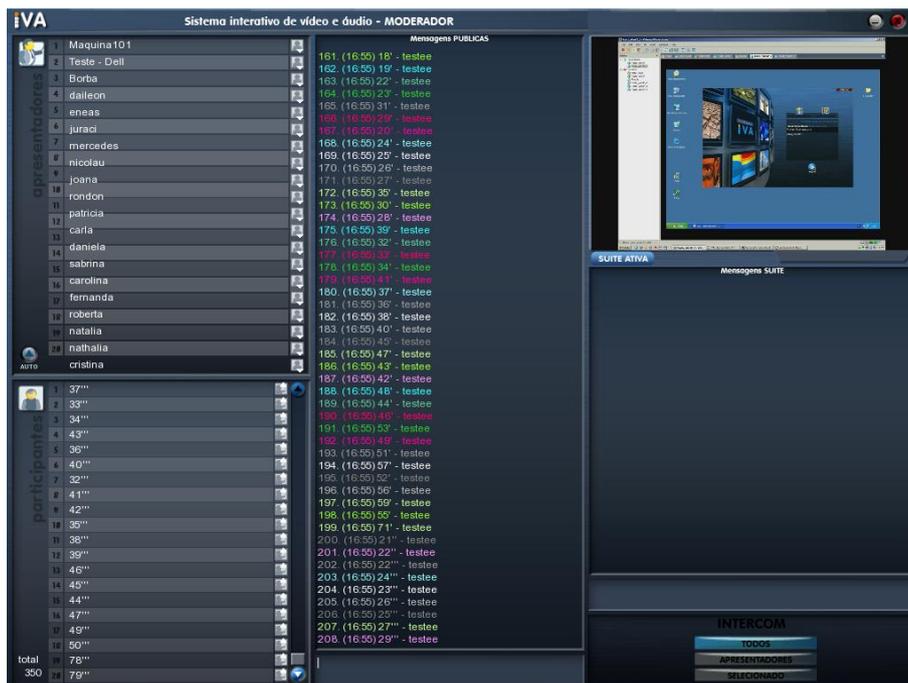


Figura 6.3: Interface do Moderador IVA

disso, sugere-se que o operador do Moderador esteja próximo do professor para intervir durante a exposição quando há dúvidas ou necessidade de intervenções com áudio e vídeo por parte de algum outro Apresentador.

A sugestão de proximidade entre operador do Moderador e professor não é restritiva. O Moderador possui uma área de visualização para acompanhar a sessão através de áudio e vídeo, e possui um canal de comunicação por mensagem de texto com todos os Apresentadores. Dessa forma, questões que estão além do seu conhecimento e necessidade de intervenções podem ser encaminhadas ao professor.

6.1.3 Suíte

A Suíte tem a função de escolher quais dos Apresentadores pré-selecionados pelo Moderador serão exibido para todos os participantes da sessão. Ela possui seis áreas de pré-visualização para Apresentadores e uma grande área para visualização do Apresentador que está “no ar”. Ou seja, o operador da Suíte pode escolher seis dos vinte Apresentadores para realizar pré-visualização, e um dos seis Apresentadores para ir ao ar.

O recurso de mixagem de áudio também é oferecido pela Suíte. O operador pode selecionar o áudio dos Apresentadores em pré-visualização que ele deseja botar no ar. A mixagem de áudio torna possível um debate entre professor e salas remotas, por exemplo. Tal recurso é opcional e pode ser habilitado ou desabilitado de forma simples.

Além de mixagem de áudio, existe o recurso de áudio prioritário. O operador da Suíte pode selecionar um Apresentador para ter prioridade e, independente da configuração de áudio adotada para a sessão, o Apresentador prioritário terá seu áudio no ar. Um uso típico desse recurso é a colocação do áudio do professor como prioritário, ou seja, o seu áudio sempre será ouvido pelos demais, independente da utilização ou não do recurso de mixagem.



Figura 6.4: Interface da Suíte IVA

A figura 6.4 apresenta a interface da Suíte. Ela é composta pela lista de Apresentadores à esquerda, seis áreas de pré-visualização, a área de vídeo principal ao centro e configurações de áudio à direita.

Um outro recurso que a Suíte introduz é a telepresença. Cada Apresentador em pré-visualização pode ser transmitido em fluxos *multicast* secundários, que poderão ser exibidos através dos Visualizadores (detalhados na subseção 6.1.4). O nome “telepresença” vem da utilização desse recurso: o professor, em seu ambiente de aula, pode utilizar aparelhos de TV para exibir os Apresentadores das salas remotas, simulando a presença dos alunos. Dessa forma, o professor pode visualizar os alunos das salas remotas levantando a mão em sinal de dúvida, por exemplo, e tornando a experiência de interação mais rica (figura 6.5).



Figura 6.5: Professor e o conceito de telepresença

6.1.4 Visualizador

Conforme citado na subseção 6.1.3, o Visualizador tem a função de exibir áudio e vídeo de um sinal de telepresença. Ao entrar em uma sessão, o Visualizador escolhe um sinal de telepresença dentre os sinais disponíveis.

Da mesma forma que o professor pode fazer uso de Visualizadores para exibir as salas remotas, elas também podem utilizar Visualizadores para realizar a exibição de outras salas.

Além disso, o professor pode ocasionalmente utilizar mais de um Apresentador

durante sua exposição: um para transmitir sua imagem e voz e outro para transmitir um conjunto de slides ou um arquivo de vídeo. Assim, as salas remotas podem utilizar um Visualizador em uma TV para exibir a imagem do professor, enquanto um projetor com o Apresentador da sala exibe o conjunto de slides que o professor preparou.

6.1.5 Chat

O Chat é uma entidade cliente que não possui recursos de áudio e vídeo. De forma simples, é utilizada apenas para realizar troca de mensagens de texto.

Em uma sala remota, por exemplo, com um computador para cada aluno, a aula é tipicamente acompanhada através de um projetor e um televisor auxiliar. Em caso de dúvida, o aluno poderia contatar o monitor da sala para enviar uma mensagem ou solicitar interação.

Tornando disponível um Chat para cada aluno, abre-se um canal de comunicação direto para troca de mensagens com o Moderador. Isso pode proporcionar uma maior interação durante a aula e resolver o problema da não interação por timidez do aluno.

6.2 Detalhes técnicos

6.2.1 Arquitetura de distribuição

A distribuição de áudio e vídeo do IVA é feita através da Suíte. Os Apresentadores que estiverem com pré-visualização ativada enviam os fluxos de áudio e vídeo em *unicast* para a Suíte, que por sua vez encaminha os fluxos - no ar e em telepresença - para os endereços *multicast* pré-selecionados.

A arquitetura de distribuição é ilustrada pela figura 6.6. Setas indicam o fluxo de dados dos Apresentadores até a Suíte em *unicast* e outras indicam os fluxos *multicast*, a partir da Suíte e para os Apresentadores. Note que todas as entidades clientes possuem uma conexão TCP com o Moderador - tal conexão é utilizada para troca de mensagens de controle.

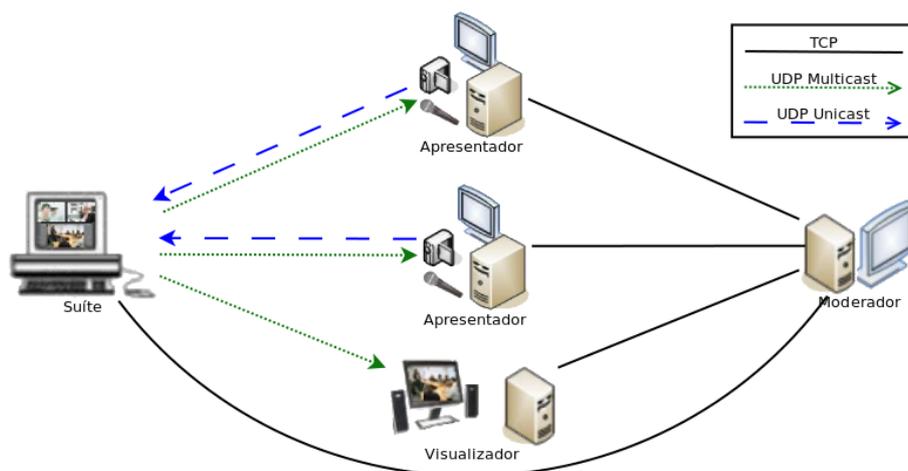


Figura 6.6: Modelo de integração entre as entidades do IVA

6.2.2 Protocolo para distribuição de áudio e vídeo

Para realizar a transmissão dos fluxos de áudio e vídeo, o IVA utiliza um protocolo proprietário fortemente baseado no RTP. A preferência de um protocolo proprietário frente a uma opção padrão aconteceu pela necessidade de recursos que o RTP não oferece. Tais recursos serão detalhados nessa subseção.

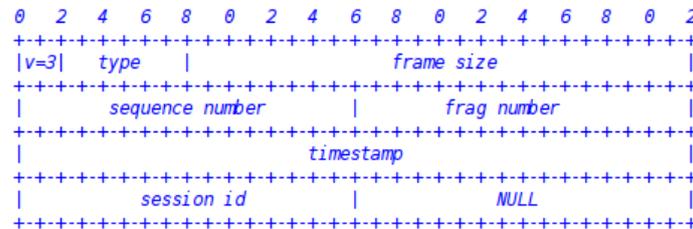


Figura 6.7: Cabeçalho utilizado no IVA - Padrão em todos os pacotes de áudio e vídeo

O cabeçalho base utilizado nos pacotes do IVA, ilustrado na figura 6.7 apresenta campos em comum com o RTP, tais como versão, número de sequência e *timestamp*. Os demais campos serão detalhados abaixo:

- Tipo: Define o tipo do pacote, que pode ser de vídeo, áudio ou dados;
- Tamanho do quadro: Define o número de fragmentos que compõem o quadro;
- Número do fragmento: Define o número do fragmento atual;
- Identificador de sessão: Define o identificador único da sessão.

O identificador de sessão tem um papel fundamental em uma transmissão, pois evita que pacotes de uma determinada sessão interfiram na transmissão de outra. Tal cenário pode acontecer se duas sessões utilizarem o mesmo endereço *multicast* para transmissão, por exemplo.

Os pacotes de áudio e vídeo possuem campos adicionais distintos e têm a função de informar o perfil de codificação daquele fluxo de dados. Pacotes de áudio, conforme apresentado na figura 6.8, informam *bitrate* e codec utilizados na codificação. As *flags* tem a função de identificar múltiplas fontes de áudio, que têm impacto posterior sobre o mecanismo de tratamento de eco acústico.



Figura 6.8: Cabeçalho utilizado no IVA - Obrigatório nos pacotes de áudio

Já pacotes de vídeo, conforme ilustrado na figura 6.9, trazem dados como *bitrate*, altura e largura dos quadros, codec utilizado, número de quadros por segundo e ainda um campo com o identificador do vídeo - esse último tem a função de identificar o fluxo de vídeo e evitar conflitos.

A utilização de dados tão específicos de áudio e vídeo se deve a um recurso importante do IVA chamado de multitaxas. A Suíte possui controle dinâmico sobre

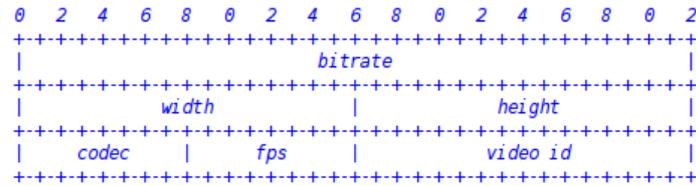


Figura 6.9: Cabeçalho utilizado no IVA - Obrigatório nos pacotes de vídeo

o perfil de codificação de cada Apresentador. Durante uma transmissão, a Suíte pode variar parâmetros de codificação dos Apresentadores e essa alteração é aplicada sem a necessidade de uma troca de mensagens de controle.

6.2.3 Protocolo de controle de sessão

O IVA utiliza um protocolo de controle de sessão baseado no SIP (*Session Initiation Protocol*) (ROSENBERG et al., 2002). O protocolo prevê as seguintes operações:

- Registro de entidades clientes (REGISTER);
- Promoção de clientes participantes para geradores de conteúdo (PRESENTER_IN);
- Despromoção de Apresentadores (PRESENTER_OUT);
- Início e parada de transmissão de áudio e vídeo (START_TX e STOP_TX);
- Configuração de canais de telepresença (SET_TP);
- Troca de mensagens de texto (CHAT_MESSAGE);
- Indicação de troca de status de câmera e microfone e pedido de atenção (WARNING).

A utilização de um protocolo proprietário ao invés do protocolo padrão existente se deve à flexibilidade que um protocolo proprietário dá ao projeto. Várias mensagens são comuns entre o protocolo IVA e o SIP, mas o modelo que o IVA utiliza com uma entidade centralizadora de controle e outra entidade centralizadora de conteúdo multimídia impossibilitou que apenas as mensagens padrão fossem utilizadas.

6.2.4 Perfis de fluxos de dados

O Apresentador, gerador de conteúdo do IVA, produz uma *stream* de vídeo e uma de áudio. Essas *streams* são codificadas e enviadas através do protocolo detalhado na subseção 6.2.2.

O vídeo do IVA utiliza a resolução *Standard Definition* (SD - 720x480) como padrão. Podem ser utilizados os codificadores MPEG2, MPEG4 e H264. Em testes realizados em laboratório, o perfil de codificação que mostrou a melhor relação “qualidade *vs.* largura de banda utilizada *vs.* custo computacional” foi a utilização de MPEG4, 30 frames por segundo e um *bitrate* de 1,4 Mbps.

Para a codificação de áudio, utiliza-se tipicamente um *bitrate* de 128 kbps com os codificadores MP2 ou AAC. Segundo a tabela contida na figura 4.1, tal *bitrate* é utilizado na codificação em formatos como o DivX.

6.3 Detalhamento da implementação

6.3.1 Contextualização

Em um cenário real de utilização do IVA, participantes sem acesso a *multicast* serão impedidos de visualizar o conteúdo da sessão. Tal situação é ilustrada na figura 6.10, em que os fluxos de dados em *multicast* para determinados participantes não são encaminhados por roteadores intermediários.

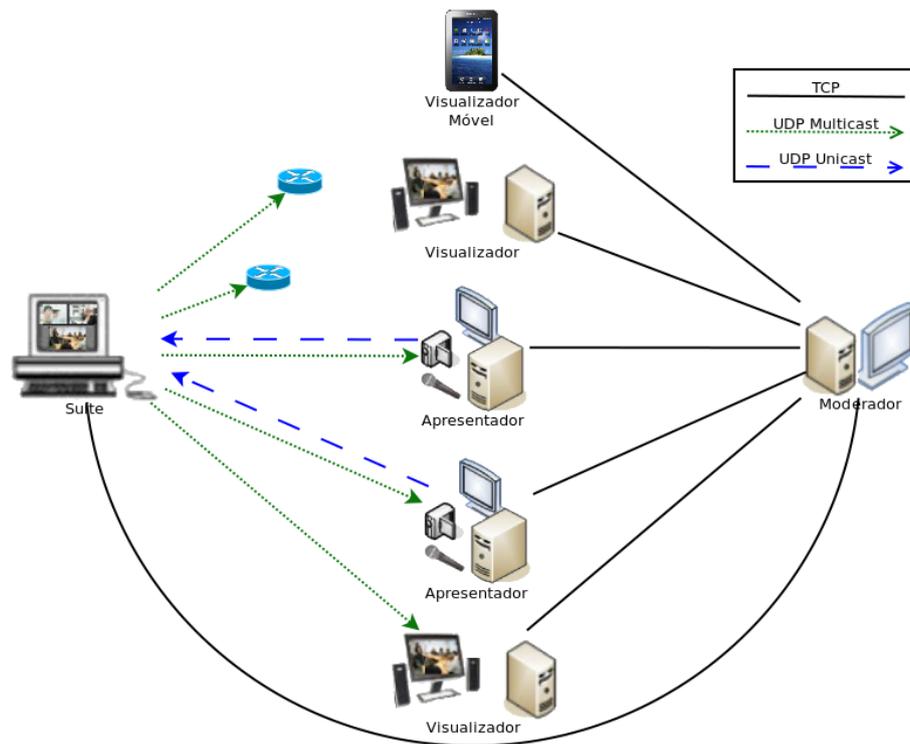


Figura 6.10: Modelo de integração real, em que entidades podem não estar na rede *multicast*

6.3.2 Integração

A solução proposta para o problema do *multicast* não atingir todos os clientes é baseada na criação de uma nova entidade, o Servidor de *Stream Unicast* (SSU). Essa entidade se comporta como um cliente no sistema, com algumas particularidades e restrições.

Pelo menos um SSU deverá ser instalado em uma rede que seja atingida pelo fluxo *multicast* da sessão. Esse SSU receberá o fluxo *multicast* e enviará toda a informação do *multicast* via um canal UDP *unicast* diretamente para o cliente que não possui suporte a *multicast*. Os novos fluxos de dados enviados pelo SSU serão adaptados às limitações do dispositivo e da rede, ou seja, poderão ter resolução, FPS e *bitrate* diferentes dos originais.

A figura 6.11 apresenta o modelo com um SSU instalado. A sua presença deve ser transparente para o sistema, ou seja, os usuários não precisam saber de onde vem o fluxo de dados que eles estão visualizando. O Moderador, que possui conhecimento de todos os participantes da sessão, também possui um canal de comunicação TCP com cada SSU e utiliza esse canal para troca de mensagens de controle.

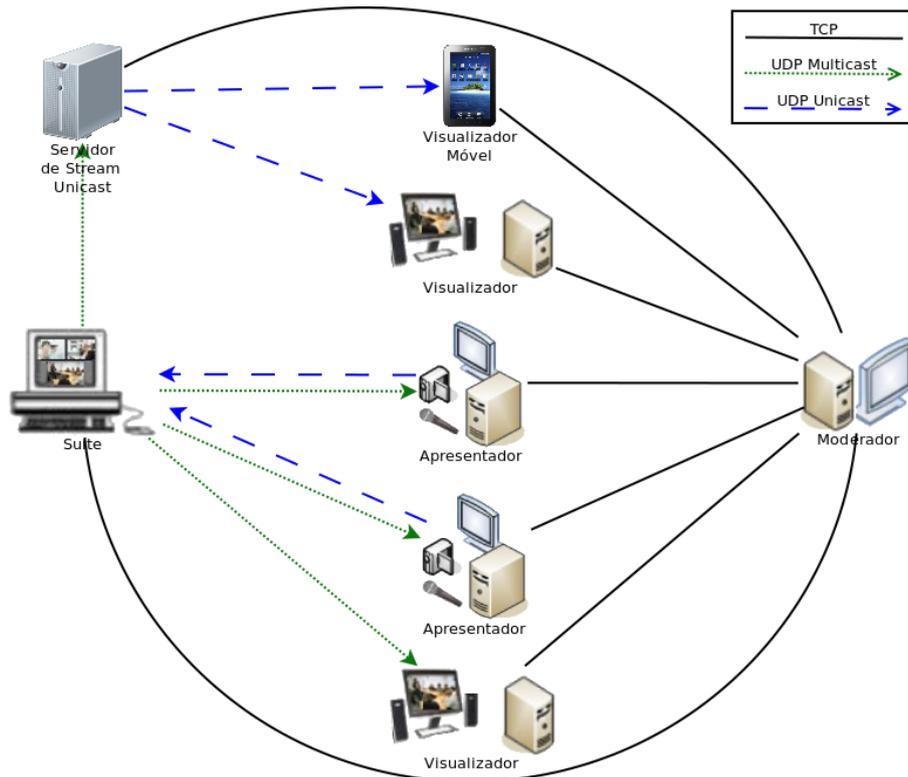


Figura 6.11: Utilização de servidores de *stream* para realizar entrega em *unicast*

O cliente, ao detectar a ausência de um fluxo em *multicast*, solicita ao Moderador um fluxo de dados em *unicast*. O Moderador seleciona entre todos os servidores de *stream* conectados na sessão os que são potencialmente mais adequados - servidores que já refutaram uma requisição não são selecionados, por exemplo. O endereço dos selecionados é enviado ao cliente que fez a requisição inicial.

O cliente dispara para cada endereço IP recebido do Moderador uma mensagem *Echo Request*, a fim de testar RTT e o número de saltos a partir dele para cada servidor. Todos os dados são enviados novamente ao Moderador, que nesse ponto já tem a possibilidade de ordená-los por adequabilidade.

Com as métricas recebidas do cliente, o Moderador ordena os servidores de *stream* por número de saltos - quanto mais próximo do cliente, mais adequado -, e o critério de desempate é o número de clientes que um servidor está atendendo. Dessa forma, com uma lista ordenada, o Moderador envia a requisição para os servidores em ordem crescente até que algum retorne uma resposta positiva.

Ao receber uma mensagem positiva do servidor de *stream*, o Moderador encaminha ao cliente o endereço e o índice de sessão do servidor que o está atendendo para identificação da origem do fluxo de dados, e assim a transmissão está estabelecida. Para a transmissão ser encerrada, basta o cliente sair da sessão ou então solicitar ao Moderador a interrupção do fluxo de dados, indicando o índice do servidor que o atende.

6.3.3 Interação

Conforme detalhado na seção 5.1, o início de uma transmissão *unicast* depende de uma interação inicial entre as entidades. A figura 6.12 ilustra os passos já descritos anteriormente.

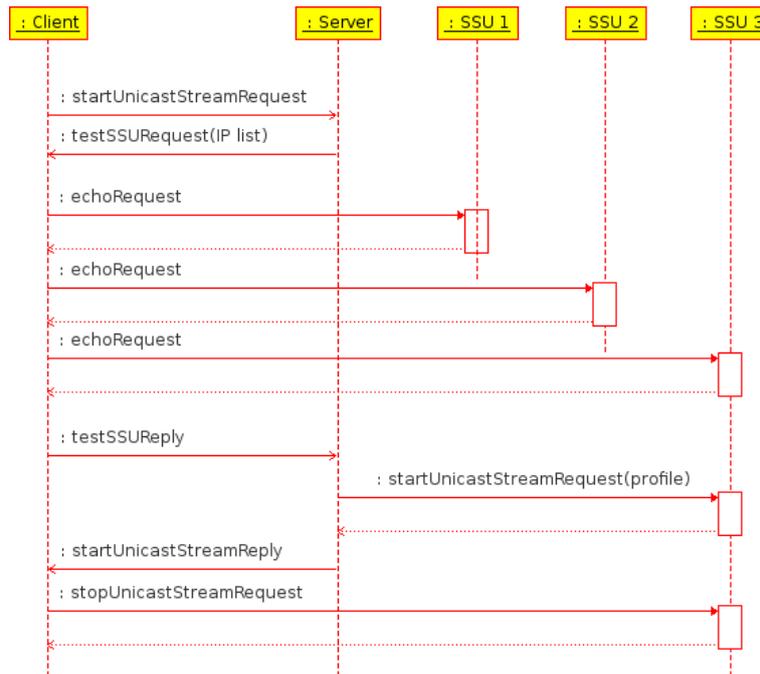


Figura 6.12: Interação entre as entidades envolvidas na solução

6.3.4 Implementação do SSU

O SSU foi implementado na linguagem C++, assim como todo o IVA. Não foi criada uma interface gráfica pois o SSU não necessita interação com um operador.

Para o Moderador, o SSU é um cliente conforme os demais, com algumas particularidades. Ainda assim, o SSU deve realizar um registro prévio como a entidade que é, e depois do registro inicial ele está apto a iniciar uma transmissão *unicast* a um participante da sessão.

Os processos de recepção de um fluxo *multicast* e envio de um fluxo *unicast* são feitos através de uma biblioteca própria do IVA. Já os processos de decodificação, redimensionamento e codificação são realizados através da biblioteca FFmpeg. Além disso, é utilizada uma biblioteca de alta performance para filas com múltiplos consumidores, própria do IVA.

Abaixo é apresentado o pseudoalgoritmo que descreve início e fim de uma transmissão:

```
1 AdicionaCliente($fonte, $destino, $perfil)
2     se nao esta recebendo de $fonte
3         inicia recebimento de $fonte
4     seleciona fluxos decodificados de $fonte
5     busca fluxo com resolucao desejada
6     se nao existe fluxo com resolucao desejada
7         gera novo fluxo com resolucao desejada
8     seleciona fluxos codificados de fluxo pre-selecionado
9     seleciona fluxo codificado mais parecido com $perfil
10    se nao existe fluxo codificado parecido com $perfil
11        gera novo fluxo codificado com $perfil
12    inicia envio de fluxo codificado para $destino
13
14 RemoveCliente($destino)
15     seleciona fluxos enviados para $destinos
16     finaliza envio de fluxos para $destino
17     se nao existe envio do mesmo fluxo para outro destino
18         remove fluxo
```

7 RESULTADOS

Na seção 5.3, uma série de métricas foi proposta para avaliar a solução, e o capítulo 6 apresentou a arquitetura que será utilizada para validação. Para isso, um ambiente de testes foi proposto, e será descrito na seção 7.1. Entretanto, como será visto posteriormente, um novo ambiente teve de ser proposto para obter resultados conclusivos.

A figura 7.1 ilustra o cenário real de testes. À esquerda, uma câmera é utilizada apontada para um cronômetro, conforme descrito na subseção 5.3.1. À direita, os computadores que fizeram parte dos ambientes de testes.



Figura 7.1: Ambiente real de testes

7.1 Teste preliminar

Na seção 5.3, quatro métricas foram propostas para avaliar a solução. O primeiro teste, chamado aqui de teste preliminar, limitou-se a avaliar a primeira métrica: o atraso. Entretanto, os resultados indicaram problemas no ambiente e nas máquinas utilizadas, conforme será detalhado posteriormente, e um novo teste foi feito, apresentado na seção seguinte.

7.1.1 Ambiente de testes

O ambiente de testes contou com uma configuração típica de sessão do IVA. Um Moderador, uma Suíte e um Apresentador gerador de conteúdo formam o conjunto mínimo necessário para colocar o sistema em funcionamento. A figura 7.2 ilustra o ambiente de testes completo.

Todos os testes rodaram em um ambiente controlado de rede. Foram utilizados dois *switches* de rede configuráveis com recurso de desabilitação de fluxos *multicast*. Um roteador *wireless* também foi utilizado, com o objetivo de testar o sistema em diferentes configurações de rede.

O *multicast* foi habilitado no *switch* 1 e desabilitado no outro. Dessa forma, todos os clientes conectados ao *switch* 1 receberiam o fluxo de dados da sessão em *multicast* e os demais seriam atendidos pelo SSU.

Ao *switch* 1 foram conectados Moderador, Suíte, um Apresentador gerador de conteúdo e um SSU. Os demais clientes foram configurados de tal forma a representar perfis diversos dentro da sessão. Os clientes conectados ao *switch* 2 possuem uma largura de banda limitada, o que significa uma limitação no *bitrate* da sessão, e os clientes conectados via *wireless* possuem, além da largura de banda limitada, uma baixa resolução. Tais clientes poderiam ser, por exemplo, dispositivos móveis com tela pequena, como *smartphones*. Entretanto, todas as máquinas utilizadas nos testes possuem arquitetura PC.

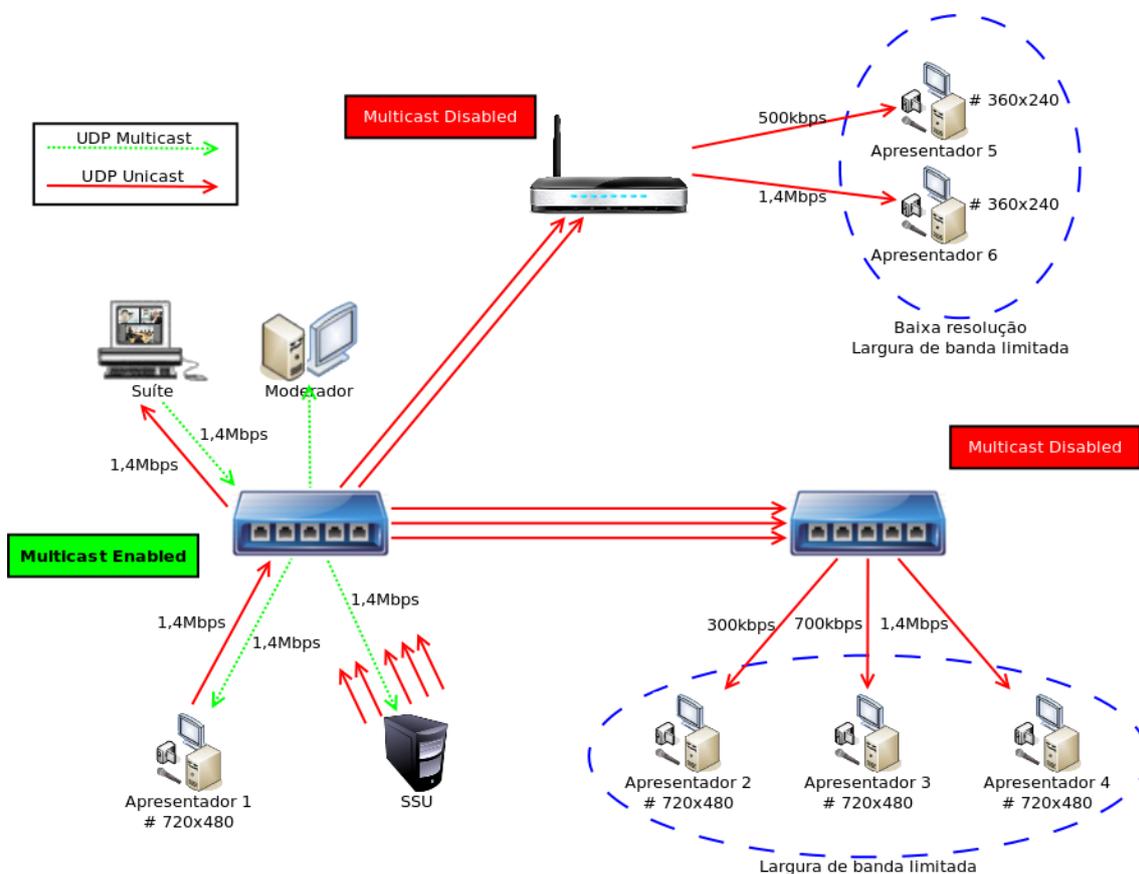


Figura 7.2: Ambiente de testes - teste preliminar

7.1.2 Configuração das máquinas

Conforme citado anteriormente, todas as máquinas utilizadas possuem arquitetura PC, e a descrição detalhada de cada uma encontra-se no apêndice B. A configuração das máquinas utilizadas como Moderador e Suíte não foram detalhadas porque suas configurações são irrelevantes nesta avaliação.

7.1.3 Avaliação de atraso

Conforme pôde ser visto no início deste capítulo, a medição de atraso foi feita utilizando uma câmera filmadora apontada para um cronômetro, e a imagem da câmera era transportada através do sistema até todos os pontos. Dessa configuração geral foram feitas diversas fotografias, e o atraso foi medido comparando o tempo do cronômetro original com o tempo dos pontos remotos.

A tabela 7.1 apresenta todos os dados gerados a partir do teste. Ao examinar a tabela, dois dados são contraditórios. O primeiro e mais importante é: por que o Apresentador 1, que é o único Apresentador a receber os dados direto do *multicast*, possui atraso médio superior a quase todos os demais Apresentadores? Em segundo lugar, por que o atraso médio do Apresentador 6 é tão superior aos demais?

O segundo questionamento possui uma explicação mais simples. A máquina utilizada como Apresentador 6 não só apresentou um atraso médio muito superior aos demais como também apresentou mau funcionamento durante os testes. Segundo a equipe responsável pelo equipamento, o mau funcionamento era frequente, o que desqualificou os resultados da sua avaliação.

Com relação ao Apresentador 1, existem duas hipóteses. A primeira é a mesma do problema anterior, um defeito específico da máquina. A segunda é referente a única configuração diferente dos demais Apresentadores. O Apresentador 1 é o único da sessão a capturar, codificar e enviar vídeo para o sistema. Essa característica pode justificar o atraso médio maior do que os demais.

Cronômetro	Apresentador 1	Apresentador 2	Apresentador 3	Apresentador 4	Apresentador 5	Apresentador 6	MIN	MAX	AVG	STDEV
00:43.17	00:42.85 (0.32s)	00:42.93 (0.24s)	00:42.93 (0.24s)	00:42.89 (0.28s)	00:42.85 (0.32s)	00:42.85 (0.32s)	0.24s	0.32s	0.287s	0.036s
00:58.25	00:57.91 (0.34s)	00:58.04 (0.21s)	00:58.03 (0.22s)	00:58.04 (0.21s)	00:58.04 (0.21s)	00:57.84 (0.41s)	0.21s	0.41s	0.267s	0.079s
01:11.81	01:11.53 (0.28s)	01:11.64 (0.17s)	01:11.64 (0.17s)	01:11.62 (0.19s)	01:11.53 (0.28s)	01:11.55 (0.26s)	0.17s	0.28s	0.225s	0.049s
00:42.43	00:42.18 (0.25s)	00:42.25 (0.18s)	00:42.25 (0.18s)	00:42.21 (0.22s)	00:42.27 (0.16s)	00:42.12 (0.31s)	0.16s	0.31s	0.217s	0.051s
01:26.45	01:26.15 (0.30s)	01:26.28 (0.17s)	01:26.28 (0.17s)	01:26.25 (0.20s)	01:26.18 (0.27s)	01:25.98 (0.47s)	0.17s	0.47s	0.263s	0.105s
01:29.67	01:29.42 (0.25s)	01:29.50 (0.17s)	01:29.49 (0.18s)	01:29.46 (0.21s)	01:29.48 (0.19s)	01:28.92 (0.75s)	0.17s	0.75s	0.292s	0.207s
01:34.15	01:33.93 (0.22s)	01:33.96 (0.19s)	01:33.96 (0.19s)	01:33.93 (0.22s)	01:33.89 (0.26s)	01:33.89 (0.26s)	0.19s	0.26s	0.223s	0.029s
01:37.42	01:37.18 (0.24s)	01:37.21 (0.21s)	01:37.21 (0.21s)	01:37.21 (0.21s)	01:37.18 (0.24s)	01:36.81 (0.61s)	0.21s	0.61s	0.287s	0.145s
01:43.31	01:43.14 (0.17s)	01:43.14 (0.17s)	01:43.14 (0.17s)	01:43.14 (0.17s)	01:43.08 (0.23s)	01:42.87 (0.44s)	0.17s	0.44s	0.225s	0.099s
01:46.92	01:46.71 (0.21s)	01:46.75 (0.17s)	01:46.75 (0.17s)	01:46.76 (0.16s)	01:46.71 (0.21s)	01:46.68 (0.24s)	0.16s	0.24s	0.193s	0.029s
01:51.21	01:51.00 (0.21s)	01:51.06 (0.15s)	01:51.06 (0.15s)	01:51.03 (0.18s)	01:51.00 (0.21s)	01:50.82 (0.39s)	0.15s	0.39s	0.215s	0.082s
01:56.52	01:56.37 (0.15s)	01:56.43 (0.09s)	01:56.43 (0.09s)	01:56.43 (0.09s)	01:56.37 (0.15s)	01:55.89 (0.63s)	0.09s	0.63s	0.200s	0.194s
02:05.51	02:05.25 (0.26s)	02:05.31 (0.20s)	02:05.31 (0.20s)	02:05.34 (0.17s)	02:05.28 (0.23s)	02:04.78 (0.73s)	0.17s	0.73s	0.298s	0.195s
13:49.14	13:48.92 (0.22s)	13:48.88 (0.26s)	13:48.95 (0.19s)	13:48.95 (0.19s)	13:48.83 (0.31s)	13:48.75 (0.39s)	0.19s	0.39s	0.260s	0.072s
13:53.50	13:53.25 (0.25s)	13:53.37 (0.13s)	13:53.39 (0.11s)	13:53.39 (0.11s)	13:53.28 (0.22s)	13:52.75 (0.75s)	0.11s	0.75s	0.262s	0.225s
13:57.39	13:57.12 (0.27s)	13:57.18 (0.21s)	13:57.19 (0.20s)	13:57.18 (0.21s)	13:57.15 (0.24s)	13:56.68 (0.71s)	0.20s	0.71s	0.307s	0.182s
14:03.34	14:03.10 (0.24s)	14:03.17 (0.17s)	14:03.17 (0.17s)	14:03.17 (0.17s)	14:03.08 (0.26s)	14:02.63 (0.71s)	0.17s	0.71s	0.287s	0.193s
14:09.19	14:08.87 (0.32s)	14:08.93 (0.26s)	14:08.92 (0.27s)	14:08.97 (0.22s)	14:08.89 (0.30s)	14:08.64 (0.55s)	0.22s	0.55s	0.320s	0.108s
	(0.250s)	(0.186s)	(0.182s)	(0.189s)	(0.238s)	(0.496s)	0.18s	0.50s	0.257s	0.110s

Tabela 7.1: Teste preliminar: medição de atraso em cada ponto

7.2 Teste conclusivo

Devido aos resultados contraditórios obtidos no primeiro teste, algumas mudanças foram propostas no ambiente a fim de obter resultados mais consistentes e conclusivos.

7.2.1 Ambiente de testes

A primeira mudança no ambiente de testes foi a remoção da máquina utilizada como Apresentador 6 e a sua substituição por uma nova com configuração similar. Além disso, as duas hipóteses levantadas para o resultado inconsistente de atraso médio do Apresentador 1 foram tratadas.

Como medida para a primeira hipótese levantada anteriormente, a máquina utilizada como Apresentador 1 também foi substituída, o que exclui a possibilidade de ser uma característica específica daquele PC. Em segundo lugar, um novo Apresentador foi inserido na sessão, o Apresentador 0, e sua função foi apenas gerar o conteúdo consumido pelos demais.

Dessa forma, todos os Apresentadores, de 1 a 6, possuem as mesmas características na sessão, e, desconsiderando diferenças de hardware entre os equipamentos, todos podem ser avaliados da mesma maneira. O novo ambiente de testes é ilustrado na figura 7.3.

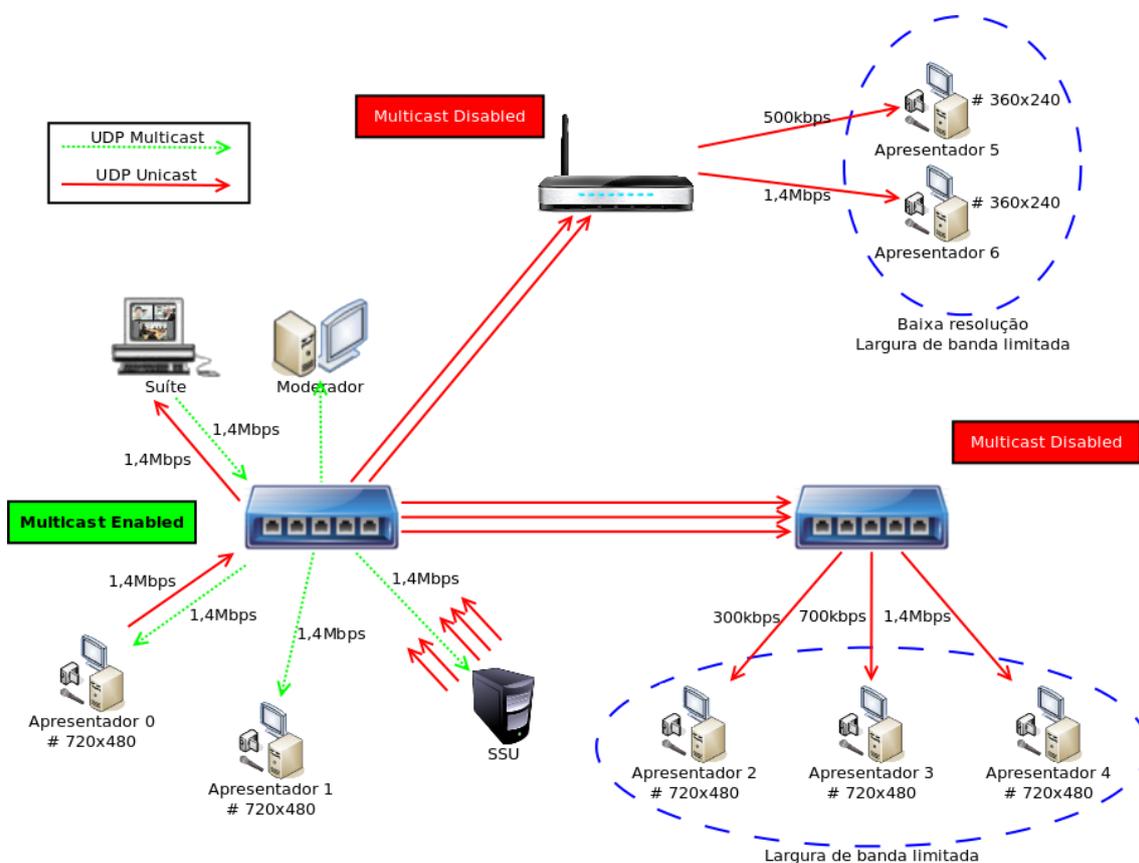


Figura 7.3: Ambiente de testes - teste conclusivo

7.2.2 Configuração das máquinas

O apêndice C apresenta a nova configuração das máquinas utilizadas no teste. Conforme citado anteriormente, os Apresentadores 1 e 6 tiveram mudança de configuração.

7.2.3 Avaliação de atraso

Os dados da nova avaliação de atraso são apresentados na tabela 7.2. Na nova avaliação, composta por 22 tomadas de tempo, o Apresentador 1 continuou com um atraso médio superior a alguns, mas a diferença caiu para 20 milissegundos. Sabendo que um sistema de vídeo que utiliza trinta quadros por segundo exibe em média um quadro a cada 33 milissegundos, uma diferença de 20 milissegundos (entre os Apresentadores 1 e 3) representa menos que um quadro.

Além disso, existem flutuações de atraso normais em um sistema computacional que realiza exibição de vídeo. Cada ponto possui um relógio diferente, o que significa que a exibição de novos quadros não é sincronizada entre todos os pontos. O cronômetro utilizado tem precisão centesimal, o que também pode representar variações nas medições.

Cronômetro	Apresentador 1	Apresentador 2	Apresentador 3	Apresentador 4	Apresentador 5	Apresentador 6	MIN	MAX	AVG	STDEV
00:27:04.51	00:27:04.34 (0.17s)	00:27:04.34 (0.17s)	00:27:04.34 (0.17s)	00:27:04.32 (0.19s)	00:27:04.29 (0.22s)	00:27:04.31 (0.20s)	0.17s	0.22s	0.187s	0.019s
00:28:04.76	00:28:04.59 (0.17s)	00:28:04.59 (0.17s)	00:28:04.61 (0.15s)	00:28:04.59 (0.17s)	00:28:04.59 (0.17s)	00:28:04.59 (0.17s)	0.15s	0.17s	0.167s	0.007s
00:29:05.31	00:29:05.09 (0.22s)	00:29:05.14 (0.17s)	00:29:05.14 (0.17s)	00:29:05.11 (0.20s)	00:29:05.09 (0.22s)	00:29:05.11 (0.20s)	0.17s	0.22s	0.197s	0.021s
00:29:49.93	00:29:49.75 (0.18s)	00:29:49.75 (0.18s)	00:29:49.76 (0.17s)	00:29:49.75 (0.18s)	00:29:49.71 (0.22s)	00:29:49.71 (0.22s)	0.17s	0.22s	0.192s	0.020s
00:34:28.96	00:34:28.75 (0.21s)	00:34:28.75 (0.21s)	00:34:28.73 (0.23s)	00:34:28.75 (0.21s)	00:34:28.75 (0.21s)	00:34:28.75 (0.21s)	0.21s	0.23s	0.213s	0.007s
00:37:59.45	00:37:59.23 (0.22s)	00:37:59.25 (0.20s)	00:37:59.26 (0.19s)	00:37:59.25 (0.20s)	00:37:59.23 (0.22s)	00:37:59.23 (0.22s)	0.19s	0.22s	0.208s	0.012s
00:41:12.00	00:41:11.79 (0.21s)	00:41:11.82 (0.18s)	00:41:11.82 (0.18s)	00:41:11.79 (0.21s)	00:41:11.75 (0.25s)	00:41:11.73 (0.27s)	0.18s	0.27s	0.217s	0.033s
00:58:42.76	00:58:42.56 (0.20s)	00:58:42.56 (0.20s)	00:58:42.55 (0.21s)	00:58:42.53 (0.23s)	00:58:42.53 (0.23s)	00:58:42.53 (0.23s)	0.20s	0.23s	0.217s	0.014s
00:59:42.90	00:59:42.69 (0.21s)	00:59:42.71 (0.19s)	00:59:42.71 (0.19s)	00:59:42.71 (0.19s)	00:59:42.68 (0.22s)	00:59:42.69 (0.21s)	0.19s	0.22s	0.202s	0.012s
01:00:42.71	01:00:42.51 (0.20s)	01:00:42.51 (0.20s)	01:00:42.51 (0.20s)	01:00:42.51 (0.20s)	01:00:42.49 (0.22s)	01:00:42.51 (0.20s)	0.20s	0.22s	0.203s	0.007s
01:01:42.90	01:01:42.73 (0.17s)	01:01:42.75 (0.15s)	01:01:42.73 (0.17s)	01:01:42.73 (0.17s)	01:01:42.70 (0.20s)	01:01:42.73 (0.17s)	0.15s	0.20s	0.172s	0.015s
01:16:09.25	01:16:09.09 (0.16s)	01:16:09.04 (0.21s)	01:16:09.12 (0.13s)	01:16:09.09 (0.16s)	01:16:09.09 (0.16s)	01:16:09.09 (0.16s)	0.13s	0.21s	0.163s	0.024s
01:17:09.71	01:17:09.51 (0.20s)	01:17:09.51 (0.20s)	01:17:09.51 (0.20s)	01:17:09.51 (0.20s)	01:17:09.50 (0.21s)	01:17:09.49 (0.22s)	0.20s	0.22s	0.205s	0.008s
01:18:09.31	01:18:09.07 (0.24s)	01:18:09.11 (0.20s)	01:18:09.11 (0.20s)	01:18:09.11 (0.20s)	01:18:09.11 (0.20s)	01:18:09.07 (0.24s)	0.20s	0.24s	0.213s	0.019s
01:35:32.25	01:35:32.07 (0.18s)	01:35:32.03 (0.22s)	01:35:32.05 (0.20s)	01:35:32.05 (0.20s)	01:35:32.03 (0.22s)	01:35:32.03 (0.22s)	0.18s	0.22s	0.207s	0.015s
01:36:32.95	01:36:32.73 (0.22s)	01:36:32.78 (0.17s)	01:36:32.76 (0.19s)	01:36:32.75 (0.20s)	01:36:32.73 (0.22s)	01:36:32.75 (0.20s)	0.17s	0.22s	0.200s	0.017s
01:37:32.21	01:37:31.98 (0.23s)	01:37:32.01 (0.20s)	01:37:32.00 (0.21s)	01:37:31.97 (0.24s)	01:37:31.98 (0.23s)	01:37:32.00 (0.21s)	0.20s	0.24s	0.220s	0.014s
01:39:32.20	01:39:31.96 (0.24s)	01:39:32.03 (0.17s)	01:39:32.03 (0.17s)	01:39:32.01 (0.19s)	01:39:31.96 (0.24s)	01:39:31.96 (0.24s)	0.17s	0.24s	0.208s	0.032s
01:40:32.43	01:40:32.20 (0.23s)	01:40:32.26 (0.17s)	01:40:32.20 (0.23s)	01:40:32.23 (0.20s)	01:40:32.20 (0.23s)	01:40:32.20 (0.23s)	0.17s	0.23s	0.215s	0.023s
01:41:32.31	01:41:32.17 (0.14s)	01:41:32.17 (0.14s)	01:41:32.17 (0.14s)	01:41:32.13 (0.18s)	01:41:32.12 (0.19s)	01:41:32.12 (0.19s)	0.14s	0.19s	0.163s	0.024s
01:42:32.51	01:42:32.25 (0.26s)	01:42:32.28 (0.23s)	01:42:32.31 (0.20s)	01:42:32.28 (0.23s)	01:42:32.31 (0.20s)	01:42:32.28 (0.23s)	0.20s	0.26s	0.225s	0.021s
01:45:03.79	01:45:03.53 (0.26s)	01:45:03.61 (0.18s)	01:45:03.61 (0.18s)	01:45:03.59 (0.20s)	01:45:03.53 (0.26s)	01:45:03.57 (0.22s)	0.18s	0.26s	0.217s	0.033s
	(0.205s)	(0.187s)	(0.185s)	(0.198s)	(0.215s)	(0.212s)	0.19s	0.22s	0.200s	0.012s

Tabela 7.2: Teste conclusivo: medição de atraso em cada ponto

7.2.4 Avaliação de funcionamento

A fim de validar o funcionamento do sistema foram utilizados dois aplicativos. O primeiro é o Wireshark, software de análise de tráfego da rede. O segundo é o NetMeter, um aplicativo que mede a quantidade de tráfego em um determinado intervalo de tempo e exibe as informações de forma gráfica.

Assim, com o Wireshark nas máquinas hospedeiras dos Apresentadores 1 e 3, esperava-se visualizar:

1. Um tráfego constante em *multicast* chegando na máquina do Apresentador 1;
2. Um tráfego constante em *unicast* chegando na máquina do Apresentador 3.

A figura 7.4 ilustra um tráfego partindo do endereço 143.54.132.209 e chegando no endereço 233.10.156.100. O primeiro é o endereço da Suíte, responsável pela geração do conteúdo da sessão em *multicast*, e o segundo é o endereço *multicast* da sessão. Já a figura 7.5 ilustra um tráfego *unicast* partindo de 143.54.132.69, endereço do SSU, e chegando em 143.54.132.115, endereço do Apresentador 3.

21259	108.030391	143.54.132.209	233.10.156.100	LLC	I P, N(R)=96, N(S)=0; DSAP 0x94
21260	108.030466	143.54.132.209	233.10.156.100	LLC	I P, N(R)=96, N(S)=0; DSAP 0x94
21261	108.063497	143.54.132.209	233.10.156.100	LLC	I P, N(R)=96, N(S)=0; DSAP EIA R
21262	108.063657	143.54.132.209	233.10.156.100	LLC	I P, N(R)=96, N(S)=0; DSAP EIA R
21263	108.063714	143.54.132.209	233.10.156.100	LLC	I P, N(R)=96, N(S)=0; DSAP EIA R

Figura 7.4: Apresentador 1 recebendo o fluxo de dados em *multicast*

20416	182.823471	143.54.132.69	143.54.132.115	UDP	Source port: incp Destination port: 12051
20417	182.823473	143.54.132.69	143.54.132.115	UDP	Source port: incp Destination port: 12051
20418	182.823476	143.54.132.69	143.54.132.115	UDP	Source port: incp Destination port: 12051
20419	182.823478	143.54.132.69	143.54.132.115	UDP	Source port: incp Destination port: 12051
20420	182.823479	143.54.132.69	143.54.132.115	UDP	Source port: incp Destination port: 12051

Figura 7.5: Apresentador 3 recebendo o fluxo de dados em *unicast*

Além da recepção através do método correto, *unicast* ou *multicast*, a quantidade de dados recebida em cada ponto deve respeitar o valor especificado no ambiente de testes. A figura 7.6 ilustra o *bitrate* recebido pelo Apresentador 5, cujo valor esperado era 500 kbps. Conforme a figura, o *bitrate* no segundo de avaliação foi de 486,7 kbps, com pico máximo de 694,8 kbps, valores que estão dentro do esperado considerando flutuações normais na taxa de envio do vídeo.

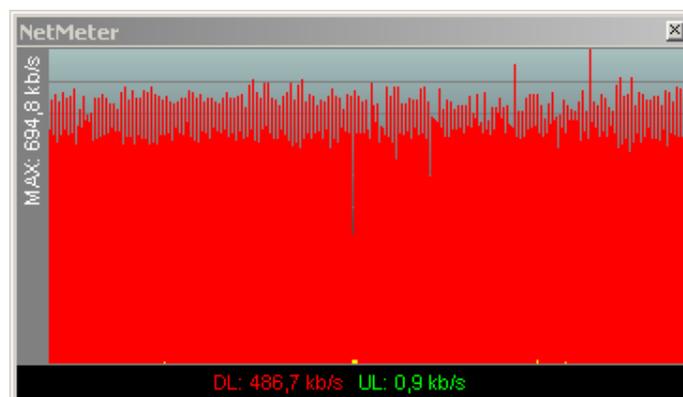


Figura 7.6: Taxa de recepção de dados em *unicast* do Apresentador 5

7.2.5 Avaliação de desempenho

A fim de avaliar o desempenho do SSU, foi utilizado o Gerenciador de Tarefas do Windows para medir demanda de processamento e consumo de memória, ilustrados na figura 7.7. Durante os testes, o consumo de memória permaneceu constante, enquanto a demanda de processamento flutuava entre 12% e 18%.



Processo	Tempo	CPU	Memória
smncc.exe	SYSTEM	00	5.048 K
spoolsv.exe	SYSTEM	00	4.476 K
stacsv.exe	SYSTEM	00	4.476 K
streamserver.exe	Felipe	15	26.480 K
stray.exe	Felipe	00	12.964 K
svchost.exe	LOCAL SERVICE	00	3.784 K
svchost.exe	SYSTEM	00	5.048 K

Figura 7.7: Demanda de processamento e memória do SSU durante a avaliação

As tarefas executadas pelo SSU durante os testes são detalhadas como segue:

1. Recepção da *stream multicast* da sessão codificada (1,4 Mbps, 720x480);
2. Decodificação da *stream* recebida;
3. Geração e envio de uma nova *stream* codificada para o Apresentador 2 (300 kbps, 720x480);
4. Geração e envio de uma nova *stream* codificada para o Apresentador 3 (700 kbps, 720x480);
5. Envio da *stream* codificada original para o Apresentador 4 (1,4 Mbps, 720x480);
6. Geração de uma nova *stream* decodificada com resolução reduzida (360x240);
7. Geração e envio de uma nova *stream* codificada para o Apresentador 5 (500 kbps, 360x240);
8. Geração e envio de uma nova *stream* codificada para o Apresentador 6 (1,4 Mbps, 360x240).

Foram quatro perfis diferentes gerados no teste. Com uma demanda de processamento de até 18%, estima-se a capacidade de geração de até 22 perfis diferentes.

7.2.6 Avaliação de saturação

A figura 7.8 ilustra a quantidade de tráfego de rede no SSU, tanto de entrada como de saída. Observa-se entrada de 1,41 Mbps e saída de 4,51 Mbps, com pico máximo de 4,87 Mbps. De fato, a quantidade de dados de entrada confere com a taxa de transmissão do gerador de conteúdo da sessão. Já a saída pode ser calculada como segue:

1. Apresentador 2: 300 kbps;
2. Apresentador 3: 700 kbps;
3. Apresentador 4: 1,4 Mbps;
4. Apresentador 5: 500 kbps;

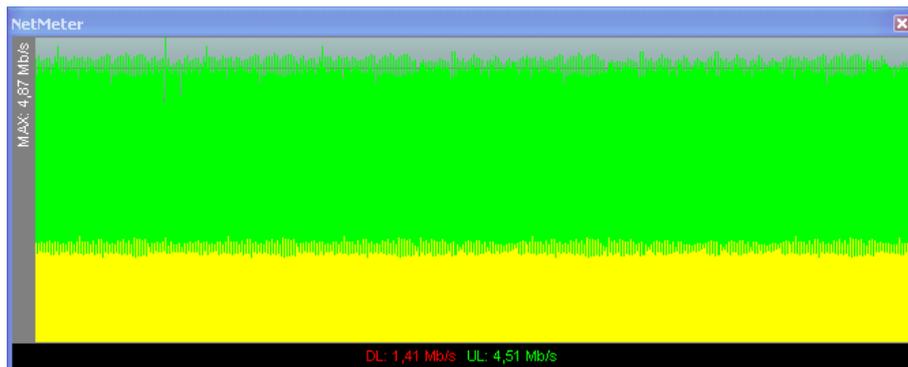


Figura 7.8: Taxa de recepção e envio de dados do SSU

5. Apresentador 6: 1,4 Mbps.

Somando as taxas pretendidas para cada Apresentador cliente do SSU obtém-se o valor de 4,3 Mbps, o que é aproximadamente o valor observado no NetMeter. A diferença explica-se pela flutuação do *bitrate* durante uma transmissão.

Considerando que o teste foi executado numa rede inteiramente Gigabit, que para cinco clientes o SSU apresenta uma saída de 4,51 Mbps e que o *bitrate* de vídeo é de 1,4 Mbps, mais de 200 clientes poderiam ser atendidos, em teoria, por um único servidor.

7.2.7 Avaliação de alcançabilidade

O ganho em alcançabilidade no ambiente de testes demonstrado é bastante óbvio. Sem uma solução como a proposta neste trabalho, os Apresentadores 2 a 6 não teriam condição de acompanhar a sessão, ou seja, apenas 16% dos participantes acompanhariam a sessão de forma funcional.

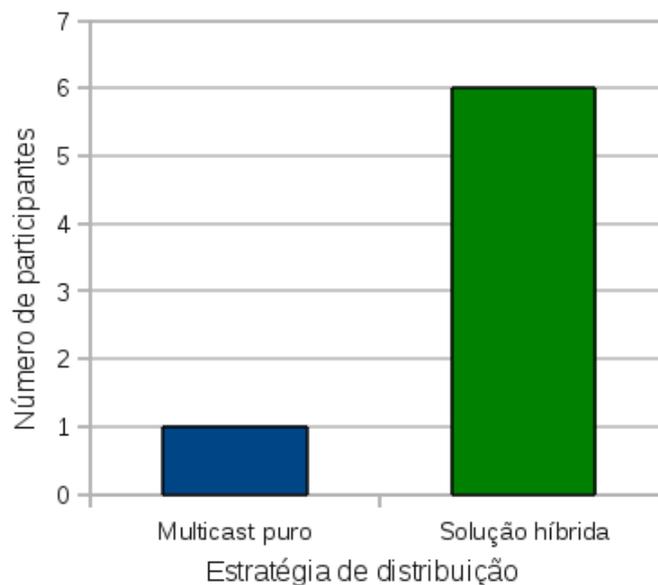


Figura 7.9: Ganho em alcançabilidade

8 CONCLUSÕES

A principal conclusão da avaliação realizada neste trabalho diz respeito à medição de atraso. O processo de recodificação de vídeo em resolução SD não agrega um atraso considerável a um sistema computacional de áudio e vídeo em tempo real.

Além disso, a solução proposta mostrou-se viável aplicada ao IVA. Pode-se utilizar um único SSU para atender um grande número de participantes e, o mais importante, uma teleaula ou reunião remota pode alcançar agora pontos que antes não era possível devido à estratégia de distribuição utilizada.

Dois aspectos apresentados neste trabalho podem ser melhor desenvolvidos no futuro. O primeiro é a questão de adaptação em si, que não pôde ser implementada de forma eficiente e por isso foi utilizado um mecanismo fixo de adaptação de vídeo. Entretanto, conforme descrito nos capítulos que seguiram, a adaptação representa uma grande oportunidade ao utilizar o *unicast* como estratégia de distribuição.

Outro ponto que poderia ser melhorado no futuro é a questão de hierarquização para alcançar uma maior escalabilidade do sistema de distribuição. Embora as avaliações tenham mostrado que um único SSU pode ser capaz de atender um grande número de participantes, a hierarquização dos servidores pode ainda ser importante nos casos em que existe um gargalo de rede.

Ainda, uma avaliação de qualidade poderia ser desenvolvida no futuro a fim de melhorar o algoritmo de adaptação. Conforme descrição neste trabalho, o servidor tem condições de adaptar os parâmetros de codificação dos participantes, mas sem ter uma referência de qualidade. Uma avaliação de qualidade pode permitir uma melhor estimativa do par “resolução *bitrate*” a fim de, por exemplo, adaptar a resolução em função do tamanho da tela do participante, e o *bitrate* em função da qualidade desejada.

REFERÊNCIAS

- ADOBE. **Real Time Messaging Chunk Stream Protocol**. [S.l.]: Adobe Systems Incorporated, 2009.
- COMER, D. E. **Redes de computadores e internet**. 4th.ed. Porto Alegre, RS: Bookman, 2007.
- FFmpeg. Disponível em: <<http://www.ffmpeg.org/>>. Acesso em: junho 2010.
- FOROUZAN, B. A. **Comunicação de dados e redes de computadores**. 4th.ed. [S.l.]: McGraw-Hill, 2008.
- HALSALL, F. **Multimedia communications: applications, networks, protocols and standards**. 1st.ed. [S.l.]: Pearson Education Inc., 2001.
- HANDLEY, M.; JACOBSON, V. **SDP: session description protocol**. United States: [s.n.], 1998.
- HELIX. **Helix DNA Network Protocols**. Disponível em: <<https://protocol.helixcommunity.org/>>. Acesso em: fevereiro 2010.
- KUROSE, J. F.; ROSS, K. W. **Computer Networking: a top-down approach (4th edition)**. 4.ed. [S.l.]: Addison Wesley, 2007.
- MACHADO, C.; ROESLER, V.; HUSEMANN, R. **IVA: sistema interativo de áudio e vídeo - manual de instalação e uso do sistema**. [S.l.: s.n.], 2008.
- POSTEL, J. **User Datagram Protocol: RFC 768**. [S.l.: s.n.], 1980. 3p. n.768.
- POSTEL, J. **RFC 791: internet protocol**. [S.l.: s.n.], 1981.
- POSTEL, J. **Transmission Control Protocol: RFC 793**. [S.l.: s.n.], 1981. 85p. n.793.
- Projetos em Áudio e Vídeo. Disponível em: <<http://www.inf.ufrgs.br/prav>>. Acesso em: junho 2010.
- Rede Nacional de Ensino e Pesquisa. Disponível em: <<http://www.rnp.br>>. Acesso em: setembro 2009.
- ROESLER, V.; HUSEMANN, R.; COSTA, C. A new multimedia synchronous distance learning system: the iva study case. In: ACM SYMPOSIUM ON APPLIED COMPUTING, 2009. **Proceedings...** [S.l.: s.n.], 2009. p.1765–1770.

ROSENBERG, J. et al. **RFC3261: sip: session initiation protocol**. [S.l.: s.n.], 2002.

SCHULZRINNE, H.; CASNER, S. **RTP Profile for Audio and Video Conferences with Minimal Control**. [S.l.]: Internet Engineering Task Force, 2003. n.3551. (Request for Comments).

SCHULZRINNE, H. et al. **RTP: a transport protocol for real-time applications**. United States: [s.n.], 2003.

SCHULZRINNE, H.; RAO, A.; LANPHIER, R. **Real Time Streaming Protocol (RTSP): RFC 2326**. [S.l.]: Network Working Group, 1998. Status: PROPOSED STANDARD.

SIMPSON, W. **Video over IP: a practical guide to technology and applications**. [S.l.]: Focal Pr, 2006.

STALLINGS, W. **Data and computer communications**. 8th.ed. [S.l.]: Pearson Prentice Hall, 2007.

STORER, J. A. **An Introduction to Data Structures and Algorithms**. [S.l.]: Birkhauser Boston, 2001.

TANENBAUM, A. S. **Redes de computadores**. 4th.ed. [S.l.]: Editora Campus, 2003.

ViDEOHelp.com. Disponível em: <<http://www.videohelp.com/dvd>>. Acesso em: junho 2010.

ANEXO A RESOLUÇÕES DE VÍDEO

A figura A.1 apresenta um comparativo entre as resoluções de vídeo mais conhecidas.

1

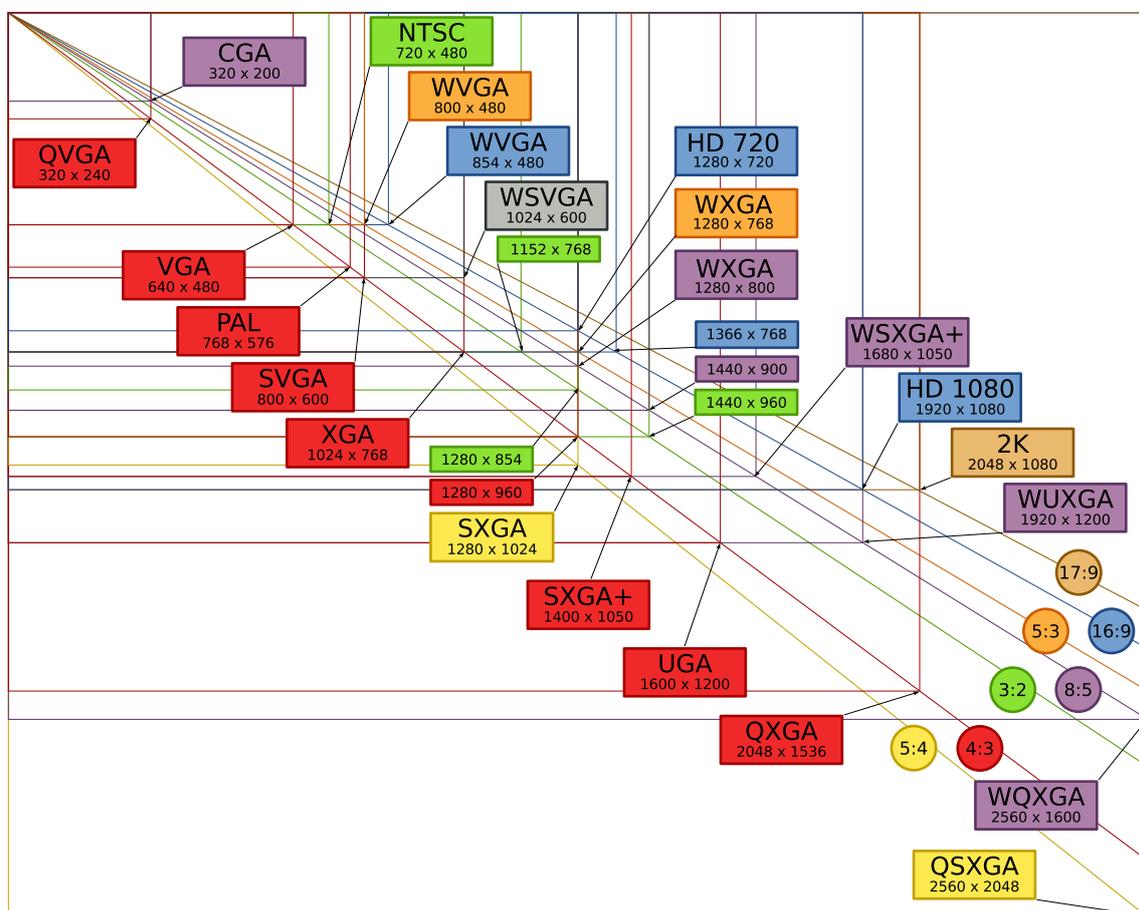


Figura A.1: Comparação de resoluções de vídeo

¹http://commons.wikimedia.org/wiki/File:Vector_Video_Standards3.svg

APÊNDICE A EXEMPLO DE MENSAGEM SDP

```

1 Session Description Protocol
2   Session Description Protocol Version (v): 0
3   Owner/Creator, Session Id (o): GoogleStreamer 1112396192 1408893144
4   IN IP4 74.125.211.182
5   Owner Username: GoogleStreamer
6   Session ID: 1112396192
7   Session Version: 1408893144
8   Owner Network Type: IN
9   Owner Address Type: IP4
10  Owner Address: 74.125.211.182
11   Session Name (s): Video
12   Connection Information (c): IN IP4 0.0.0.0
13   Connection Network Type: IN
14   Connection Address Type: IP4
15   Connection Address: 0.0.0.0
16   Bandwidth Information (b): AS:86
17   Bandwidth Modifier: AS [Application Specific
18   (RTP session bandwidth)]
19   Bandwidth Value: 86 kb/s
20   Time Description, active time (t): 0 0
21   Session Start Time: 0
22   Session Stop Time: 0
23   Session Attribute (a): control:*
24   Session Attribute Fieldname: control
25   Session Attribute Value: *
26   Session Attribute (a): range:npt=0-361.083000
27   Session Attribute Fieldname: range
28   Session Attribute Value: npt=0-361.083000
29   Media Description, name and address (m): video 0 RTP/AVP 97
30   Media Type: video
31   Media Port: 0
32   Media Protocol: RTP/AVP
33   Media Format: DynamicRTP-Type-97
34   Bandwidth Information (b): AS:54
35   Bandwidth Modifier: AS [Application Specific
36   (RTP session bandwidth)]
37   Bandwidth Value: 54 kb/s
38   Media Attribute (a): rtpmap:97 MP4V-ES/90000
39   Media Attribute Fieldname: rtpmap
40   Media Format: 97
41   MIME Type: MP4V-ES
42   Sample Rate: 90000
43   Media Attribute (a): control:trackID=2
44   Media Attribute Fieldname: control

```

```

45 Media Attribute Value: trackID=2
46   Media Attribute (a): cliprect:0,0,144,176
47 Media Attribute Fieldname: cliprect
48 Media Attribute Value: 0,0,144,176
49   Media Attribute (a): framesize:97 176-144
50 Media Attribute Fieldname: framesize
51 Media Attribute Value: 97 176-144
52   Media Attribute (a): fntp:97 profile-level-id=3;config=000001b003
53       000001b50900000100000001200086c40033182c2090a31f000001b25
54       876694430303339
55 Media Attribute Fieldname: fntp
56 Media Format: 97 [MP4V-ES]
57 Media format specific parameters: profile-level-id=3
58   [Level Code: Reserved (3)]
59   Media format specific parameters: config=000001b003000001b509
60   00000100000001200086c40033182c2090a31f000001b25876694
61   430303339
62   Configuration: 000001B003000001B50900000100000001200086C4003318
63   ...
64   0000 0000 0000 0000 0000 0001 = start code prefix: 0x00000001
65   1011 0000 = Start code: visual_object_sequence_start_code (176)
66   0000 0011 = profile_and_level_indication: Reserved (3)
67   0... .... = visual_object_identifier: 0
68   .000 1... = visual_object_type: video ID (1)
69   .... .0.. = video_signal_type: 0
70   .... ..01 = Stuffing: 1
71   0000 0000 0000 0000 0000 0001 = start code prefix: 0x00000001
72   0000 0000 = Start code: video_object_start_code (0)
73   0000 0000 0000 0000 0000 0001 = start code prefix: 0x00000001
74   0010 0000 = Start code: video_object_layer_start_code (32)
75   0... .... = random_accessible_vol: 0
76   .000 0000 1... .... = video_object_type_indication: Simple
77   Object Type (1)
78   .0.. .... = is_object_layer_identifier: 0
79   ..00 01.. = aspect_ratio_info: 1:1 (Square) (1)
80   ..0. .... = vol_control_parameters: 0
81   ...0 0... = video_object_layer_shape: rectangular (0)
82 Media Description, name and address (m): audio 0 RTP/AVP 96
83 Media Type: audio
84 Media Port: 0
85 Media Protocol: RTP/AVP
86 Media Format: DynamicRTP-Type-96
87   Bandwidth Information (b): AS:32
88   Bandwidth Modifier: AS [Application Specific
89   (RTP session bandwidth)]
90   Bandwidth Value: 32 kb/s
91   Media Attribute (a): rtpmap:96 MP4A-LATM/22050/1
92 Media Attribute Fieldname: rtpmap
93 Media Format: 96
94 MIME Type: MP4A-LATM
95 Sample Rate: 22050/1
96   Media Attribute (a): control:trackID=3
97 Media Attribute Fieldname: control
98 Media Attribute Value: trackID=3
99   Media Attribute (a): fntp:96 profile-level-id=40;cpresent=0;
100   config=400027103fc0
101 Media Attribute Fieldname: fntp
102 Media Format: 96 [MP4A-LATM]

```

102	Media format specific parameters: profile-level-id=40
103	Media format specific parameters: cpresent=0
104	Media format specific parameters: config=400027103fc0

APÊNDICE B TESTE PRELIMINAR - CONFIGURAÇÃO DAS MÁQUINAS

TESTE - CONFIGURAÇÃO DAS MÁQUINAS	
ENTIDADE	DESCRIÇÃO
Apresentador 1	Desktop Processador Intel Core 2 Quad Q9450 2,66GHz 3,24GB de memória RAM Placa de vídeo ATI Radeon HD 3800 Microsoft Windows XP Service Pack 3 Professional 32 bits
Apresentador 2, Apresentador 4	Desktop Dell Optiplex 960 Processador Intel Core 2 Quad Q9550 2,83GHz 3GB de memória RAM Placa de vídeo ATI Radeon HD 3470 Microsoft Windows Vista Business Service Pack 1 32 bits
Apresentador 3, SSU	Desktop Processador Intel Core 2 Quad Q6600 2,4GHz 3,24GB de memória RAM Placa de vídeo NVIDIA GeForce 8500 GT Microsoft Windows XP Service Pack 3 Professional 32 bits
Apresentador 5	Notebook Asus W5F Processador Intel Core 2 Duo T5600 1,83GHz 1GB de memória RAM Placa de vídeo Intel GMA 950 (onboard) Microsoft Windows XP Service Pack 3 Professional 32 bits
Apresentador 6	Notebook Asus F9S Processador Intel Core 2 Duo T7300 2GHz 1GB de memória RAM Placa de vídeo NVIDIA GeForce 8400M G Microsoft Windows XP Service Pack 3 Professional 32 bits
EQUIPAMENTOS DE REDE	
Roteador Wireless D-Link DIR-655	
Switch Dell PowerConnect 2824	
Switch Dell PowerConnect 6224	
Filmadora Sony HDR-HC5	

Tabela B.1: Teste preliminar: descrição dos equipamentos que formaram o ambiente de testes

APÊNDICE C TESTE CONCLUSIVO - CONFIGURAÇÃO DAS MÁQUINAS

TESTE - CONFIGURAÇÃO DAS MÁQUINAS	
ENTIDADE	DESCRIÇÃO
Apresentador 1, SSU	Desktop Processador Intel Core 2 Quad Q6600 2,4GHz 3,24GB de memória RAM Placa de vídeo NVIDIA GeForce 8500 GT Microsoft Windows XP Service Pack 3 Professional 32 bits
Apresentador 2, Apresentador 3	Desktop Dell Optiplex 960 Processador Intel Core 2 Quad Q9550 2,83GHz 3GB de memória RAM Placa de vídeo ATI Radeon HD 3470 Microsoft Windows Vista Business Service Pack 1 32 bits
Apresentador 4	Desktop Processador Intel Pentium D 3,4GHz 2GB de memória RAM Placa de vídeo NVIDIA GeForce 6500 Microsoft Windows Vista Business Service Pack 2 32 bits
Apresentador 5	Notebook Asus W5F Processador Intel Core 2 Duo T5600 1,83GHz 1GB de memória RAM Placa de vídeo Intel GMA 950 (onboard) Microsoft Windows XP Service Pack 3 Professional 32 bits
Apresentador 6	Notebook Sony Vaio VGN-NS31M Processador Intel Dual Core T3400 2,17Hz 4GB de memória RAM Placa de vídeo ATI Radeon HD 3430 Microsoft Windows Vista Home Premium Service Pack 1 32 bits
EQUIPAMENTOS DE REDE	
Roteador Wireless D-Link DIR-655	
Switch Dell PowerConnect 2824	
Switch Dell PowerConnect 6224	
Filmadora Sony HDR-HC5	

Tabela C.1: Teste conclusivo: descrição dos equipamentos que formaram o ambiente de testes