

UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

**GUILHERME ALBERTO SOUSA RIBEIRO**

***AVALIAÇÃO DE DESEMPENHO DE UM SERVIDOR DE VÍDEO DE  
FLUXO CONTÍNUO (STREAMING)***

São Luís  
2013

**GUILHERME ALBERTO SOUSA RIBEIRO**

***AVALIAÇÃO DE DESEMPENHO DE UM SERVIDOR DE VÍDEO DE  
FLUXO CONTÍNUO (STREAMING)***

Monografia apresentada no Curso de Ciência da  
Computação da UFMA, como requisito parcial para  
a obtenção do grau de BACHAREL em Ciência da  
Computação.

Orientador: Prof. Dr. Mario Antônio Teixeira Meireles.

São Luís  
2013

Ribeiro, Guilherme Alberto Sousa.

Avaliação de desempenho de um servidor de vídeo de fluxo contínuo (streaming)/ Guilherme Alberto Sousa Ribeiro. – São Luís, 2013.

40 f.

Impresso por computador (fotocópia).

Orientador: Mário Antônio Teixeira Meireles.

Monografia (Graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2013.

1. Software – servidor multimídia. 2. Fluxo contínuo. I Título.

CDU 004.04'27

GUILHERME ALBERTO SOUSA RIBEIRO

**AVALIAÇÃO DE DESEMPENHO DE UM SERVIDOR DE VÍDEO DE  
FLUXO CONTÍNUO (STREAMING)**

Monografia apresentada no Curso de Ciência da  
Computação da UFMA, como requisito parcial para  
a obtenção do grau de BACHAREL em Ciência da  
Computação.

Aprovada em 06/03/2013.

Banca Examinadora



---

**Prof. Mario Antônio Teixeira Meireles (Orientador)**

Doutor em Ciência da Computação  
Universidade Federal do Maranhão



---

**Prof. Alexandre César Muniz de Oliveira**

Doutor em Ciência da Computação  
Universidade Federal do Maranhão



---

**Prof. Geraldo Braz Junior**

Mestre em Engenharia Elétrica  
Universidade Federal do Maranhão

*A Deus.*

*Aos meus pais, avós e irmão, por todo o apoio.*

*A minha namorada, Fernanda, por todo apoio,  
incentivo e compreensão a esta conquista.*

*Aos amigos, pelo incentivo, ajuda e compa-  
nheirismo.*

## Resumo

Este trabalho tem como objetivo realizar a avaliação de desempenho de um servidor multimídia para transmissões em fluxo contínuo. Com o uso do protocolo RTSP, é possível acessar vídeos deste servidor em tocadores multimídia ou por meio de navegadores através do uso do método URL (dentro de um código HTML) referente a este protocolo de transmissão em tempo real.

Palavras-chaves: Avaliação de Desempenho, Servidor Multimídia, Fluxo Contínuo, Protocolo RTSP.

## **Abstract**

This work aims to perform a performance evaluation of a multimedia server for streaming broadcasts. Using the RTSP protocol, you can access videos on this server or media players via web browsers by using the URL method (within a HTML) for this transmission protocol in real time.

Keywords: Performance Evaluation, Multimedia Server, Streaming, RTSP Protocol.

## Agradecimentos

Em primeiro lugar a Deus, por ter me dado saúde, paz e tranquilidade para chegar até este momento fundamental para meu desenvolvimento intelectual e profissional.

A minha família, por todo o apoio e compreensão pelas horas que estive ausente. Em especial minha avó Maria da Paz e o meu pai João Guilherme.

A minha namorada, Fernanda, por todo carinho, compreensão e paciência durante toda a minha jornada até a conclusão deste curso.

A todos os professores do departamento de informática por todo o conhecimento passado, em especial ao professor Alexandre César Muniz de Oliveira, por todo o aprendizado adquirido no PET e ao professor Mário Antônio Teixeira Meireles, pela orientação e paciência para o desenvolvimento deste trabalho de conclusão do curso.

A todos os meus amigos, dentro e fora da universidade, que estiveram comigo e me incentivaram durante toda esta caminhada.



*“Os covardes nunca tentam, os fracassados nunca terminam, os vencedores nunca desistem”.*

*Norman Vincent Peale*

# Sumário

<b>Lista de Figuras</b>	<b>8</b>
<b>Lista de Tabelas</b>	<b>9</b>
<b>1 Introdução</b>	<b>10</b>
<b>2 Fundamentação Teórica e Tecnologias Utilizadas</b>	<b>11</b>
2.1 Protocolo RTP . . . . .	11
2.2 Protocolo RTCP . . . . .	14
2.3 Protocolo RTSP . . . . .	15
2.4 Servidor de Mídia Contínua <i>Darwin</i> . . . . .	16
2.5 Apresentadores de Mídia ( <i>Players</i> ) . . . . .	18
2.5.1 <i>QuickTime</i> . . . . .	18
2.5.2 VLC . . . . .	18
<b>3 Avaliação de Desempenho do servidor <i>Darwin</i></b>	<b>19</b>
3.1 Objetivo . . . . .	19
3.2 Metodologia . . . . .	19
3.3 Características do Experimento . . . . .	20
3.4 Ambiente do Experimento . . . . .	21
3.5 Execução da Aplicação . . . . .	22
3.6 Resultados . . . . .	23
3.7 Discussões . . . . .	28

<b>4</b>	<b>Considerações Finais</b>	<b>31</b>
	<b>Referências Bibliográficas</b>	<b>32</b>
	<b>Anexos</b>	<b>34</b>

## Lista de Figuras

2.1	Arquitetura da Camada OSI com RTP . . . . .	12
2.2	Cabeçalho do Protocolo RTP . . . . .	13
2.3	Interação entre Cliente e Servidor RTSP . . . . .	15
2.4	Modelo Multicast (INC., 2002) . . . . .	17
2.5	Modelo Unicast (INC., 2002) . . . . .	17
3.1	Média de pacotes perdidos para 5 clientes. . . . .	24
3.2	Média de <i>bytes</i> entregues para 5 clientes. . . . .	25
3.3	Média de qualidade da transmissão para 5 clientes. . . . .	25
3.4	Média de pacotes perdidos para 15 clientes. . . . .	26
3.5	Média de bytes entregues para 15 clientes. . . . .	26
3.6	Média de qualidade da transmissão para 15 clientes. . . . .	27

## Lista de Tabelas

3.1	Fatores e seus respectivos níveis. . . . .	20
3.2	Tabela de sinais para $2^4$ fatorial. . . . .	21
3.3	Configurações do ambiente do experimento. . . . .	22
3.4	Influência de efeito dos fatores e de suas combinações. . . . .	28

# 1 Introdução

Com a utilização cada vez maior de dados que exigem grande quantidade de armazenamento e dos mais variados tipos, tanto em sistemas locais como em sistemas compartilhados, tem se buscado cada vez mais efetuar melhorias tanto no acesso quanto no processamento destes dados.

Para que tais melhorias possam ser realizadas é necessária a realização de medições através de indicadores que permitem avaliar o comportamento da aplicação e, caso necessário, modificações podem ser realizadas de forma a melhorar o desempenho de determinada aplicação.

Uma avaliação de desempenho pode ser realizada em sistemas computacionais ou em ambientes que demandem um alto poder de processamento, tanto para identificação de problemas que afetem o desempenho, como apenas para a realização de potenciais melhorias.

Neste trabalho foi realizada a avaliação de desempenho de um servidor de fluxo contínuo para vídeos através da utilização de um protocolo de transmissão de dados multimídia, o RTSP (*Real Time Streaming Protocol*).

No Capítulo 2, será apresentada toda a fundamentação teórica do trabalho, além das ferramentas utilizadas que tornaram o ambiente possível de ser avaliado.

No Capítulo 3 serão apresentados o objetivo, a metodologia, as características, o ambiente de avaliação, os procedimentos para execução da aplicação a ser avaliada, apresentação dos resultados e suas devidas discussões.

No Capítulo 4, considerações finais, apresenta-se os resultados alcançados com a realização do experimento.

## 2 Fundamentação Teórica e Tecnologias Utilizadas

Nos últimos anos, houve um crescimento extraordinário no desenvolvimento e na disseminação de aplicações de rede que transmitem e recebem áudio e vídeo pela Internet (KUROSE; ROSS, 2007). Apesar da maioria destas aplicações serem sensíveis ao atraso, um dos grandes desafios destas aplicações é reduzir cada vez mais a perda de dados ocasionais.

A utilização destes tipos de dados se dá através do uso de alguns protocolos, como o RTP (*Real transport protocol*), RTCP (*Real-time Transport Protocol*) e o RTSP (*Real Time Streaming Protocol*). Nas próximas seções estes protocolos serão apresentados. Além disso, serão apresentadas aplicações que servem e interpretam dados multimídia.

### 2.1 Protocolo RTP

O RTP (*Real-time Transport Protocol*), é definido (KUROSE; ROSS, 2007) como uma estrutura de pacote padronizada que inclui campos para dados de áudio/vídeo, números de sequência, *timestamp* (marcas de tempo), bem como qualquer outro campo potencialmente útil. Este protocolo encontra-se alocado entre as camadas de aplicação e transporte, portanto, devido a esta característica ele pode ser considerado como um *middleware*, conforme apresentado na Figura 2.1, que representa a localização deste protocolo no Modelo OSI.

Devido as suas características o RTP pode ser utilizado para transportar dados desde os formatos mais comuns, para áudio (i.e. *MP3*) e vídeo (i.e. *MPEG*), até as mídias de formato proprietário.

A utilização do protocolo está intimamente ligada ao protocolo UDP (*User Datagram Protocol*), sob o qual encontra-se o RTP. Para preparar a transmissão de dados, o lado do remetente encapsula uma parte da mídia dentro de um pacote RTP, que em seguida é encapsulado em um pacote UDP e então passado para um segmento IP. A partir de

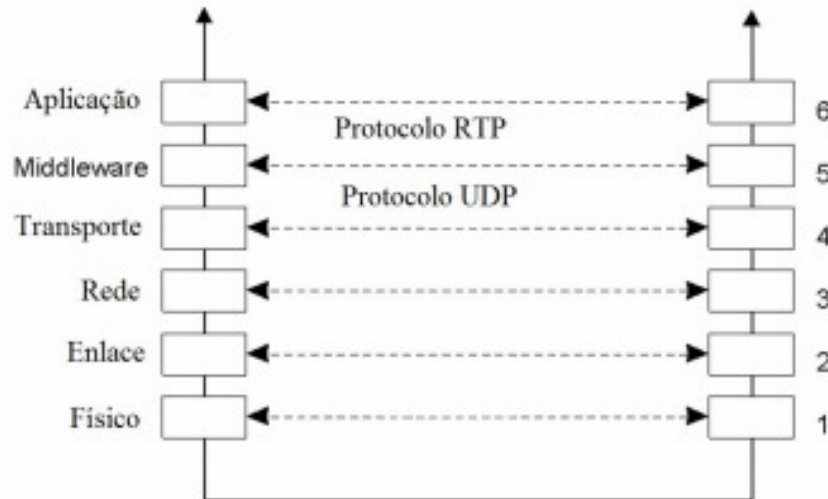


Figura 2.1: Arquitetura da Camada OSI com RTP

então, o receptor extrai o pacote referente ao protocolo RTP, dentro do segmento UDP e retira a parte da mídia que lhe foi passada, para que a mesma seja decodificada e exibida do lado receptor.

É válido ressaltar que o RTP não fornece nenhum tipo de mecanismo que garanta com restrições de temporização, nem que garanta a qualidade dos serviços solicitados (*QoS*). Nem mesmo a entrega dos pacotes, bem como sua ordem, podem ser garantidas pelo protocolo.

Em termos de comunicação, os pacotes RTP podem ser utilizados tanto em aplicações *unicast* como *multicast* (um-para-muitos ou muitos-para-muitos). Utilizando uma sessão *multicast*, no formato muitos-para-muitos, tanto emissores como fontes de uma sessão podem utilizar um mesmo grupo *multicast* para enviar seus dados na rede. Uma sessão do protocolo RTP suporta tanto dados de áudio como de vídeo (desde que apenas um destes seja transmitido por sessão).

De acordo com (GROUP et al., 1996) um pacote RTP é composto de um cabeçalho fixo, uma lista de fontes de contribuição opcional e um *payload*. A estrutura de cabeçalho será detalhada a seguir. A lista de fontes de contribuição, também conhecida como CSRC (*Contributing Source*), contém 32 *bits* e é comumente utilizado por *mixers*, que fazem a combinação destes dados com os dados identificados pelo SSRC (*Synchronization Source*). Por sua vez, o *payload* nada mais é que o formato do dado que se encontra no pacote RTP (p.e.: *jpeg*, *mp4*, *mov*, etc). O cabeçalho, assim como a lista opcional de fontes



contribuintes podem ser melhor visualizados na Figura 2.2.

Tipo de Carga Útil	Número de Sequência	Marca de Tempo	Identificador de Sincronização da Fonte	Campos Variados
--------------------	---------------------	----------------	---	-----------------

Figura 2.2: Cabeçalho do Protocolo RTP

De acordo com (KUROSE; ROSS, 2007), o cabeçalho RTP é dividido em quatro partes principais, que são os campos *payload*, número de sequência, *timestamp*, e os campos identificadores de fontes.

O *payload* (ou campo de carga útil) possui 7 bits de comprimento. Ele é utilizado para indicar o tipo de codificação que está sendo usada para os dados que estão sendo transferidos. Caso haja a necessidade de alterar a codificação dos dados durante uma sessão RTP (ex., para aumento da qualidade dos dados transmitidos), o receptor será informado através deste campo.

Outro campo principal do cabeçalho RTP é o número de sequência. Este campo possui comprimento de 16 bits e possui a característica de ser incrementado a cada vez que um pacote RTP é enviado. Esta característica permite ao receptor detectar perdas de pacotes e reorganizar a sequência dos pacotes.

Para demonstrar o instante de amostragem do primeiro byte no pacote RTP, utiliza-se o *timestamp*. Este campo de 32 bits utiliza um relógio que é inicializado no remetente e incrementado (em uma unidade) a cada vez que se avançar a uma próxima amostra. Caso uma aplicação gere 200 amostras, será incrementado o valor 200 ao *timestamp* para cada pacote.

Completando os quatro principais campos de cabeçalho do protocolo RTP, tem-se o identificador de sincronização da fonte ou SSRC. Com a necessidade de reconhecer o remetente de uma corrente de dados na rede, um número é atribuído de forma aleatória pela fonte cada vez que uma transmissão de dados é iniciada pelo remetente.

O desenvolvimento de aplicações utilizando o protocolo RTP se torna possível através de duas abordagens:

1. O desenvolvedor implementa todo o código do RTP (código e encapsulamento) no lado do remetente e decodifica o RTP no lado do receptor;

2. A utilização de bibliotecas que implementem o RTP e executem o encapsulamento e a decodificação para a aplicação.

## 2.2 Protocolo RTCP

O RTCP (*Real-time Transport Control Protocol*) é um protocolo que funciona como uma espécie de relatório, que contém dados estáticos, que podem ser utilizados em benefício da aplicação, a fim de manter um serviço de qualidade. Além disso, este protocolo não encapsula qualquer tipo de dado de áudio ou vídeo. Estes dados estáticos podem medir quantidade de pacotes, número de pacotes perdidos, dentre outras métricas. É opção do desenvolvedor utilizar tais características em sua aplicação, porém são informações essenciais para detecção de problemas, o que irá ajudar a manter um nível de qualidade no serviço que está sendo oferecido.

Existem relatórios diferentes para cada um dos lados da aplicação. No receptor, cada corrente RTP gera um relatório de recepção. Todos eles são agregados em um único pacote RTCP, que é enviado dentro da árvore *multicast*, onde todos os participantes da sessão poderão ter acesso. É bem vasta a quantidade de campos que se tem neste relatório, mas como principais podem-se citar: o identificador de sincronização da fonte dos dados RTP para o relatório que está sendo gerado; a quantidade de pacotes perdidos dentro da corrente RTP (nº de pacotes perdidos/nº de pacotes enviados), a fim de solucionar o problemas de congestionamento na rede (caso existam); último número de sequência recebido; variação de atrasos entre sucessivas chegadas de pacotes RTP. Diferentemente do receptor, o remetente que estiver na corrente RTP irá gerar o seu relatório e transmitirá o seu próprio pacote de remetente RTCP. As informações contidas nesse relatório são: o identificador de sincronização da fonte de dados; o *timestamp* e o tempo real do último pacote RTP que foi gerado; quantidade de pacotes enviados; quantidade de bytes enviados.

De acordo com (KUROSE; ROSS, 2007) os relatórios de remetente podem ser usados para facilitar a sincronização de diferentes correntes de mídia dentro de uma sessão RTP (p.ex., uma vídeo chamada, que utiliza uma corrente de áudio e vídeo). Dentro de cada uma destas correntes existe a possibilidade de adição de informações relacionadas a fonte, como o e-mail do remetente, nome do mesmo, dentre outras.

O RTCP permite também o empilhamento de seus pacotes, ou seja, podem ser agre-

gados o relatório do receptor e o relatório do remetente em um único pacote. Depois disso, este único pacote é encapsulado em um pacote UDP e enviado a rede.

## 2.3 Protocolo RTSP

O protocolo RTSP (*Real Time Streaming Protocol*) permite ao usuário um controle sobre a transmissão de dados, oferecendo serviços de pausa, avanço, reposicionamento, dentre outros. Ele funciona como um *middleware* entre o servidor e o transdutor<sup>1</sup>, para que ambos possam trocar informações de controle de reprodução.

Um dos pontos a serem levados em consideração é que o RTSP é um protocolo ‘fora da banda’, ou seja, existe um outro canal de comunicação (porta 544) fora da corrente de mídia para utilização deste protocolo, o que o torna flexível para enviar mensagens tanto pelo protocolo TCP quanto pelo UDP.

A Figura 2.3 demonstra as interações entre cliente e servidor para uma requisição RTSP.

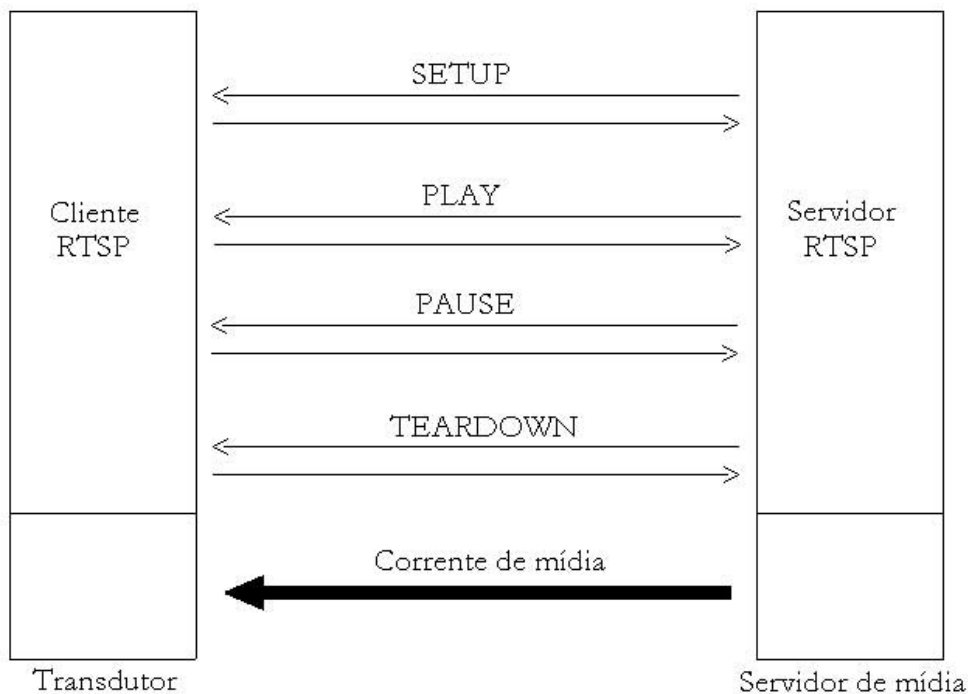


Figura 2.3: Interação entre Cliente e Servidor RTSP

De forma simplificada, as requisições a mídias através deste protocolo funcionam da

<sup>1</sup>dispositivo que recebe sinais ou dados e os retransmite em um formato correspondente.

seguinte forma:

- Uma referência é realizada ao arquivo de mídia, através de uma URL iniciada com ‘*rtsp://*’;
- O arquivo de descrição da mídia é encapsulado em uma mensagem de resposta HTTP, que é então enviada ao cliente;
- O cliente por sua vez, requisita o transdutor adequado para tocar a mídia recebida;
- O descritor do arquivo, apresenta as referências ao arquivo através do método ‘*rtsp://*’;
- Por fim, com a conexão estabelecida entre cliente e servidor, ambos passam a trocar mensagens (requisição e confirmação, respectivamente) com o intuito de inicializar a mídia solicitada.

É importante frisar que todas as mensagens de requisição e resposta (cliente e servidor) são padronizadas, o que facilita a comunicação entre as partes.

## 2.4 Servidor de Mídia Contínua *Darwin*

O *Darwin Streaming Server* (INC., 2007) (INC., 2002) é a versão *open source* da *Apple* para o *Quick Time Streaming Server* (QTSS), que permite enviar dados de forma contínua (via *streaming*) para clientes dentro da Internet (ou em rede local), utilizando os protocolos RTP e RTSP.

Para possibilitar o acesso aos dados armazenados no servidor, o *Darwin* possibilita o uso do método URL ‘*rtsp://*’, específico para dados multimídia que utilizam uma conexão não confiável para sua transmissão. Este método pode ser acessado por algum *software* cliente (p.e. VLC, *QuickTime*, etc) que possui suporte aos formatos *QuickTime* (.*mov*) ou *MPEG-4* (.*mp4*) ou através de um *web browser* que possua o *plug-in Quicktime* instalado. Os *softwares* clientes serão melhor detalhados na próxima seção.

Existem duas maneiras de se transportar dados no DSS (*Darwin Streaming Server*), uma é através do modelo *unicast*, a outra do modelo *multicast*. No modelo *multicast*, como demonstrado na Figura 2.4, existe uma única corrente na rede compartilhada entre

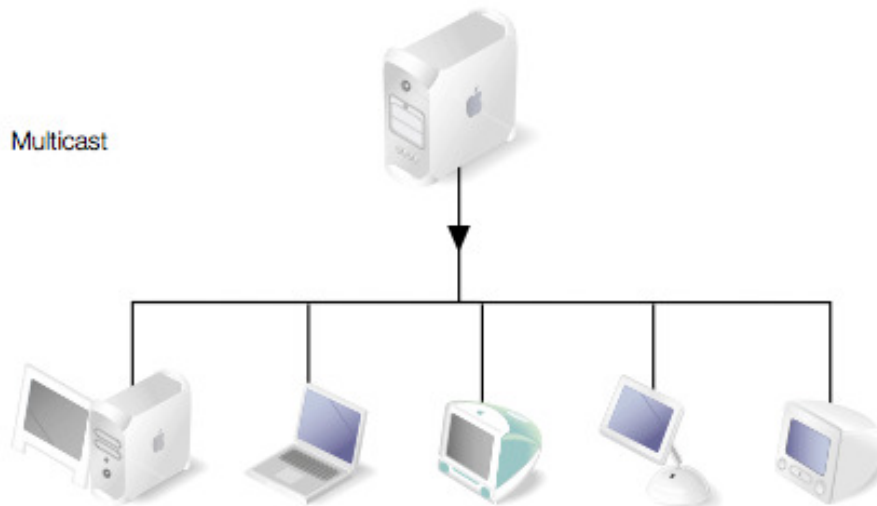


Figura 2.4: Modelo Multicast (INC., 2002)

todos os clientes, como uma espécie de rádio, onde várias pessoas se conectam a aquele conteúdo através de uma mesma frequência.

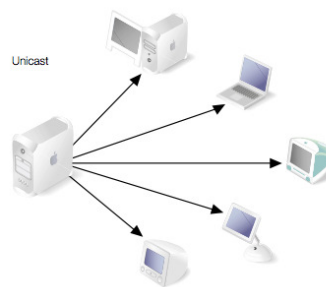


Figura 2.5: Modelo Unicast (INC., 2002)

Por sua vez no modelo *unicast*, demonstrado na Figura 2.5, cada cliente inicia sua própria corrente, o que resulta na geração de diversas conexões um-para-um entre cliente e servidor. Esta gama de conexões pode causar congestionamentos na rede, porém é a forma mais utilizada na internet, pois não necessita de nenhum tipo especial de transporte para entrega dos dados.

Além destas características, o *Darwin* possui uma série de outras opções (p.ex., transmissão de vídeo a partir de uma *webcam*). Por se tratar de um projeto *open source*, alterações podem ser feitas de acordo com a necessidade do desenvolvedor.

## 2.5 Apresentadores de Mídia (*Players*)

Esta seção será dedicada a detalhar melhor algum dos *players* que possuem suporte ao protocolo RTSP, bem como as requisições ao método URL deste protocolo.

### 2.5.1 *QuickTime*

O *QuickTime* (INC., 2006) é um interpretador de mídia bastante utilizado, principalmente em computadores *Apple*.

Ele possui um formato de vídeo próprio (*.mov*) que permite ao vídeo apresentar uma ótima qualidade de imagem em relação a maioria dos demais formatos. Esta característica fez com que o *QuickTime* fosse desenvolvido para outras plataformas, a fim de permitir a utilização deste formato de vídeo.

Além da característica anteriormente citada, este *player* oferece uma gama de outras características. Existe um serviço denominado *QuickTime Broadcaster* (INC., 2009) que permite criar uma transmissão ao vivo a qualquer usuário que possua acesso à Internet. Outro recurso é o *HTTP Live Streaming* (INC., 2011), que permite ao *QuickTime* fazer o streaming de áudio ou vídeo sem a necessidade de um servidor específico para este tipo de dados. Nos casos onde seja utilizado um servidor específico para *streaming*, o QT (como é popularmente conhecido) fornece um *plug-in* que interpreta os dados e permite que a mídia seja tocada em um *browser*. Caso o usuário deseje, a mídia pode ser tocada no próprio player através do método URL '*rtsp://*', que informará o caminho do arquivo multimídia no servidor.

### 2.5.2 VLC

O *Video Lan Client* (PROJECT, 2004), popularmente conhecido como VLC, é um *player* multimídia *open source*, multi-plataforma, que possui suporte a grande maioria dos formatos multimídia, bem como CD, DVD e protocolos *streaming*.

Além disso, o VLC permite a seu usuário realizar operações de gravação de vídeos, reprodução de arquivos compactados (não obrigando a descompactação dos mesmos), converter vídeos para os formatos suportados, reproduzir dados em ASCII, dentre outras.

## 3 Avaliação de Desempenho do servidor

### *Darwin*

Este capítulo apresenta a avaliação de desempenho realizada no servidor de fluxo contínuo *Darwin*, bem como todas as características que tornaram o ambiente proposto possível de ser avaliado e todos os resultados obtidos através desta avaliação. Além disso, algumas discussões em relação aos resultados obtidos também foi realizada.

#### 3.1 Objetivo

O experimento apresentado neste trabalho tem como objetivo avaliar o desempenho do servidor *streaming Darwin*, de forma a identificar dentre os fatores avaliados quais mais afetam o seu desempenho.

As seções seguintes serão responsáveis por demonstrar as características e informações relacionadas ao experimento que foi realizado, bem como seus resultados e algumas discussões sobre eles.

#### 3.2 Metodologia

Para gerarmos carga adequada com a intenção de avaliarmos da melhor forma possível o servidor, criou-se um código HTML que referencia os arquivos desejados no *Darwin*, fazendo com que estas mídias possam ser tocadas no *browser*.

O Fragmento de código 3.1 foi utilizado para referenciar as mídias no servidor *Darwin*. Como neste experimento vários vídeos deveriam ser executados ao mesmo tempo com a intenção de testar a capacidade de processamento do servidor, a referência aos vídeos foi replicada, afim de gerar carga sintética suficiente para realização dos testes.

Listing 3.1: Código para referenciar vídeos no servidor.

```

1 <object CLASSID="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B" width="1280" height="720" CODEBASE="http://www.apple.com/
  qtactivex/qtplugin.cab">
  <param name="src" value="nome_do_arquivo.formato">
3   <param name="qtsrc" value="rtsp://127.0.0.1/nome_do_arquivo.formato">
  <param name="autoplay" value="true">
5   <param name="loop" value="false">
  <param name="controller" value="true">
7   <embed src="nome_do_arquivo.formato" qtsrc="rtsp://127.0.0.1/nome_do_arquivo.formato" width="1280" height="720"
    autoplay="true" loop="false" controller="true" pluginspage="http://www.apple.com/quicktime/"></embed>
</object>

```

A exibição destes vídeos através do *browser* foi possível graças ao *plug-in* do *Quick-Time* compatível com o método URL *'rtsp://'*.

### 3.3 Características do Experimento

Para possibilitar a avaliação de uma aplicação, algumas métricas, com o intuito de realizar medições, precisam ser definidas. Neste experimento, foram utilizadas como métricas a quantidade de pacotes perdidos durante a transmissão, a quantidade de *bytes* recebidos pelo cliente durante a transferência dos dados e a qualidade do vídeo que foi entregue ao cliente. Os valores referentes a cada uma destas métricas foram coletados através de *logs* gerados em um local definido pelo servidor.

Quatro fatores foram definidos como de maior influência nos resultados do experimento realizado: o formato da mídia, a resolução da mídia, o browser que está sendo utilizado para acessar os dados e a quantidade de clientes que acessam o servidor ao mesmo tempo. Na Tabela 3.1, os fatores que compõem o ambiente avaliado podem ser melhor visualizados.

Tabela de Sinais	
Fatores	Níveis
A - Formato	mp4 (-1) mov (1)
B - Resolução	SD (-1) HD (1)
C - Qtde. de Objetos	5 (-1) 15 (1)
D - Cliente	Safari (-1) Chrome (1)

Tabela 3.1: Fatores e seus respectivos níveis.



Experimento	Fatores			
	A	B	C	D
1	1	1	1	1
2	1	1	1	-1
3	1	1	-1	1
4	1	1	-1	-1
5	1	-1	1	1
6	1	-1	1	-1
7	1	-1	-1	1
8	1	-1	-1	-1
9	-1	1	1	1
10	-1	1	1	-1
11	-1	1	-1	1
12	-1	1	-1	-1
13	-1	-1	1	1
14	-1	-1	1	-1
15	-1	-1	-1	1
16	-1	-1	-1	-1

Tabela 3.2: Tabela de sinais para  $2^4$  fatorial.

Cada um destes fatores possui dois níveis, o que traz ao experimento um total de 16 situações diferentes a partir da simulação. Estas situações são derivadas das interações realizadas entre os fatores mostrada na tabela de sinais de  $2^4$ , como demonstrado na Tabela 3.2.

Estes experimentos são do tipo fatorial completo, com um total de 30 replicações para cada uma das 16 situações, totalizando 480 execuções (sem contabilizar a quantidade de vídeos em cada requisição). Foram realizadas 30 replicações de forma a garantir a validade estatística dos dados através do cálculo do intervalo de confiança, garantindo que 95% dos resultados estariam dentro deste intervalo, a cada configuração do experimento.

### 3.4 Ambiente do Experimento

A infra-estrutura utilizada para realização do experimento é detalhada na Tabela 3.3.

As duas máquinas possuem configurações bem parecidas e foram conectadas em uma

Configurações de Hardware	
<b>Processador - Cliente</b>	Intel Core i3 2.13GHz Cache: 3MB L3
<b>Memória - Cliente</b>	4 GB
<b>Disco Rígido - Cliente</b>	7200 RPM 500 GB
<b>Processador - Servidor</b>	Intel Core i5 2.4GHz Cache: 256Kb (por núcleo) L2 3MB I3
<b>Memória - Servidor</b>	4 GB
<b>Disco Rígido - Servidor</b>	Seagate Momentus ST9500325ASG 5400 RPM 500 GB
Configurações de Software	
<b>SO - Cliente</b>	Windows 7 Home Premium
<b>Kernel - Cliente</b>	6.1.7601.17592
<b>SO - Servidor</b>	MacOSX 10.7.5
<b>Kernel - Servidor</b>	Darwin Kernel Version 11.4.2 x86 - 64

Tabela 3.3: Configurações do ambiente do experimento.

rede *Ethernet* com velocidade de 100 *mb/s*. No entanto, os sistemas operacionais de cada uma das máquinas são diferentes. O servidor foi executado sob o sistema operacional *MacOSX*, por sua vez cliente foi executado em um ambiente *Windows*.

Em relação aos *softwares* utilizados no cliente e no servidor, o *Chrome* (INC., 2008) que utilizava a versão 24.0.1312.57 e o *Safari* (INC., 2003) que utilizava a versão 5.1.7, foram utilizados como navegadores do lado do cliente. No lado do servidor, o *Darwin Streaming Server* foi utilizado como servidor multimídia, em sua versão 6.0.3 e o *Apache* (FOUNDATION, 1995) em sua versão 2.2, como o servidor HTTP, para requisição do código utilizado como gerador de carga sintética.

Apesar de serem citados no capítulo 2, na seção 2.5, os *players* não foram utilizados como clientes na realização dos experimentos devido algumas limitações no ambiente de realização dos experimentos.

### 3.5 Execução da Aplicação

Como citado na Seção 3.2, foi criado um código HTML que conectava o cliente aos dados desejados dentro servidor através do método URL '*rtsp://*'. As execuções foram realizadas de acordo com os fatores especificados na Tabela 3.1. O código utilizado no lado do cliente foi replicado de acordo com a quantidade de clientes necessários, em relação

a sua respectiva combinação. A aplicação não especificava nenhum tipo de ordem de execução das mídias, o próprio servidor multimídia foi o responsável pelo escalonamento dos vídeos.

Cada um dos experimentos, bem como suas replicações foram executados através da requisição do objeto HTML através dos clientes. Os principais passos da execução dos experimentos são:

1. O servidor multimídia e o servidor HTTP devem ser inicializados na máquina que funciona como servidor;
2. A URL que referencia a página HTTP que está em seu respectivo servidor é requisitada através do *browser*, no lado cliente;
3. Os objetos que referenciam os vídeos dentro da página HTML, através do uso do método '*rtsp://*', são interpretados pelo servidor multimídia;
4. O *plug-in* do *QuickTime* recebe estes dados e inicializa-os dentro do *browser*.

No fim de cada um dos experimentos, os *logs* eram gerados em um local específico dentro do servidor. Tanto os dados referentes ao servidor quanto ao cliente foram gerados em um único *log*. Os principais resultados apresentados nos *logs* tanto para o cliente quanto para o servidor são expressados através dos seguintes campos:

- Logs do Servidor: *IP, DNS, total de clientes, utilização da CPU, pacotes enviados.*
- Logs do Cliente: *status HTTP, tipo de CPU, tamanho da mídia recebida, quantidade de pacotes recebidos, quantidade de pacotes perdidos, qualidade da transmissão.*

O experimento completo foi realizado no período compreendido entre os dias 08/01 até o dia 15/01, de forma não uniforme e em horários alternados.

## 3.6 Resultados

Feita toda a coleta de dados, foram calculadas as médias das execuções, considerando um intervalo de confiança de 95% para cada média. Além disso, foram criados alguns

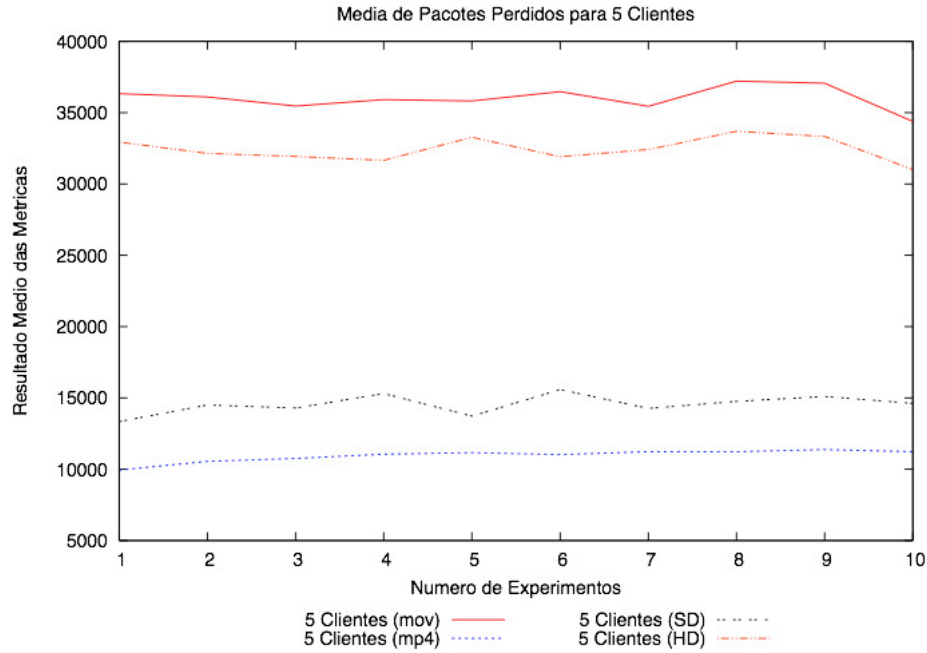


Figura 3.1: Média de pacotes perdidos para 5 clientes.

gráficos com características interessantes em relação aos fatores utilizados nesta avaliação (ver ANEXOS A, B e C).

Estes gráficos foram gerados para cada métrica avaliada, fixando o número de clientes (fator C) e alternando os níveis dos fatores A e B – verificar a Tabela 3.1. Inicialmente, demonstramos a média de pacotes perdidos para 5 clientes através da Figura 3.1. Para esta configuração podemos notar que os níveis mais influentes na métrica são o formato *.mov* e a resolução HD. No primeiro, o valor médio de pacotes perdidos foi de aproximadamente 36020 pacotes. Para as configurações voltadas a resolução HD a média foi de 32471 pacotes.

Na Figura 3.2 demonstramos a quantidade média de *bytes* recebidos pelo cliente. É notório que no formato *.mov* e na resolução standard (SD) houve uma maior eficácia na entrega dos dados. A média de *bytes* entregues para a extensão *.mov* foi de 85822 *bytes* e para a resolução SD foi de aproximadamente 77882 *bytes*.

A qualidade de transmissão do vídeo é demonstrada através da Figura 3.3. O formato e a resolução que possuem as maiores médias de qualidade são as que exigem menos processamento por meio do cliente (*.mp4* e SD, respectivamente). O formato *.mp4* obteve uma média de 58% de qualidade na transmissão e a resolução SD obteve média de 66%.

Para 15 clientes, podemos notar através da Figura 3.4 que o formato *.mp4* e a re-

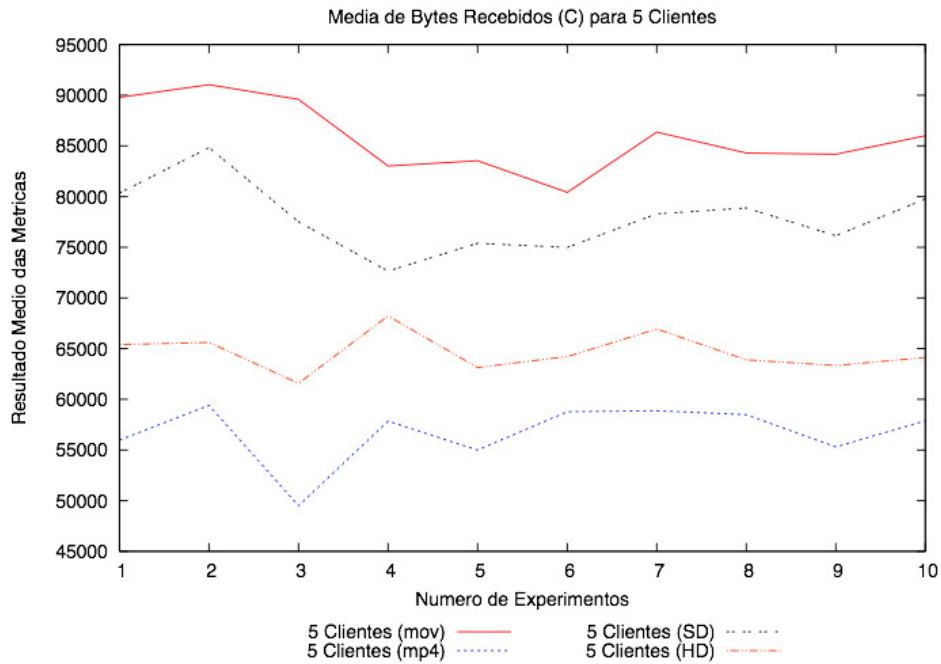


Figura 3.2: Média de *bytes* entregues para 5 clientes.

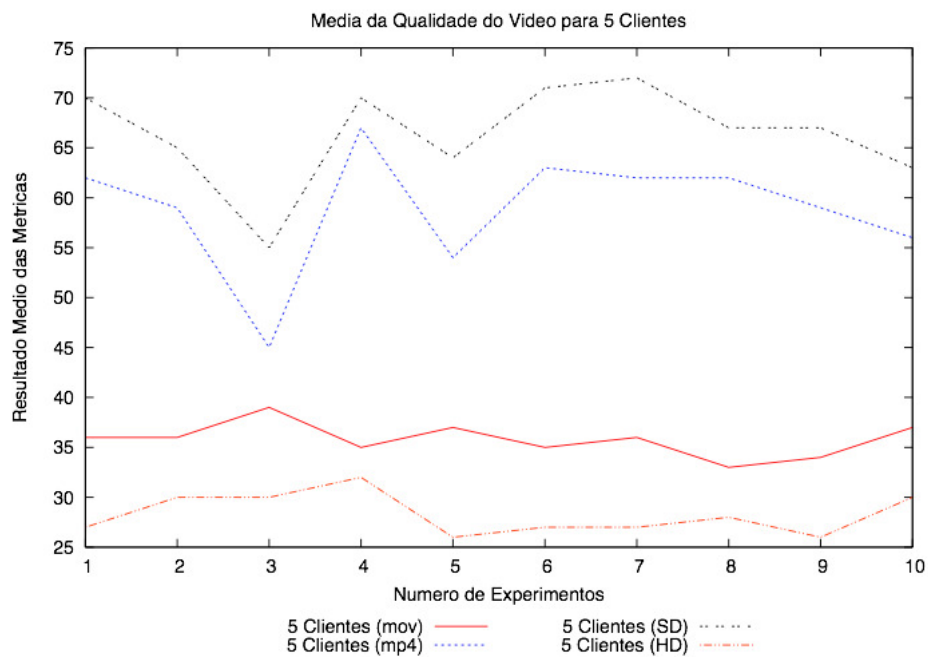


Figura 3.3: Média de qualidade da transmissão para 5 clientes.

solução HD com média de pacotes perdidos, respectivamente, de 15660 e 14024, foram os níveis que mais influenciaram esta métrica.

Na Figura 3.5 houve resultados bem próximos em relação ao formato, porém em relação a resolução, a SD obteve uma melhor média de *bytes* entregues ao cliente (42027 *bytes*). Nas médias relacionadas aos formatos, houve uma ligeira vantagem do formato

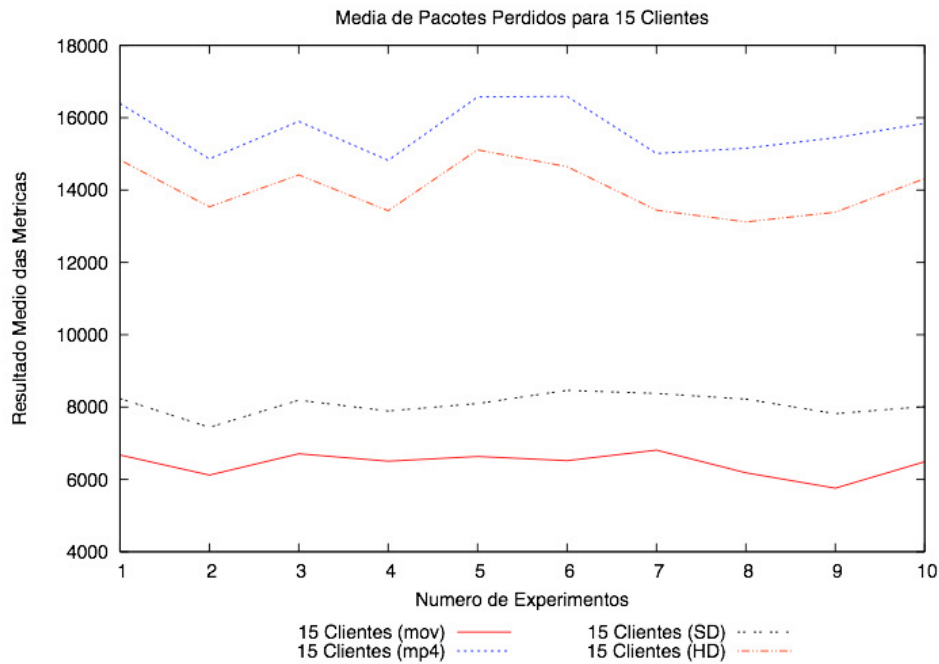


Figura 3.4: Média de pacotes perdidos para 15 clientes.

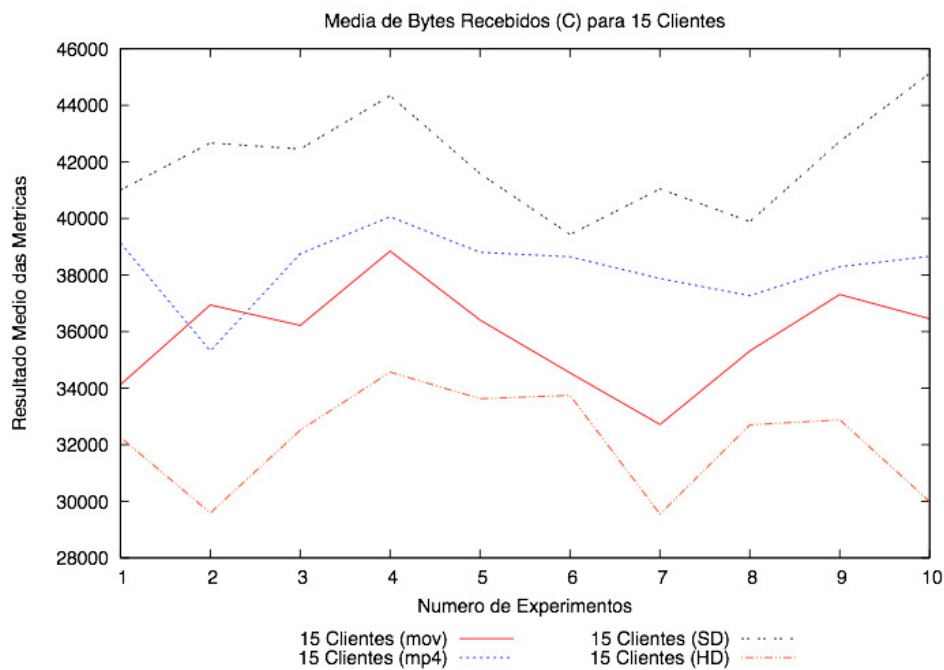


Figura 3.5: Média de bytes entregues para 15 clientes.

*.mp4*, que entregou aproximadamente 38282 *bytes* ao cliente, contra 35887 do formato *.mov*.

Em relação a qualidade de transmissão dos dados, a Figura 3.6 demonstra que o formato *.mov* com média de 47% e a resolução SD com 42%, apresentaram o melhor desempenho para esta métrica.

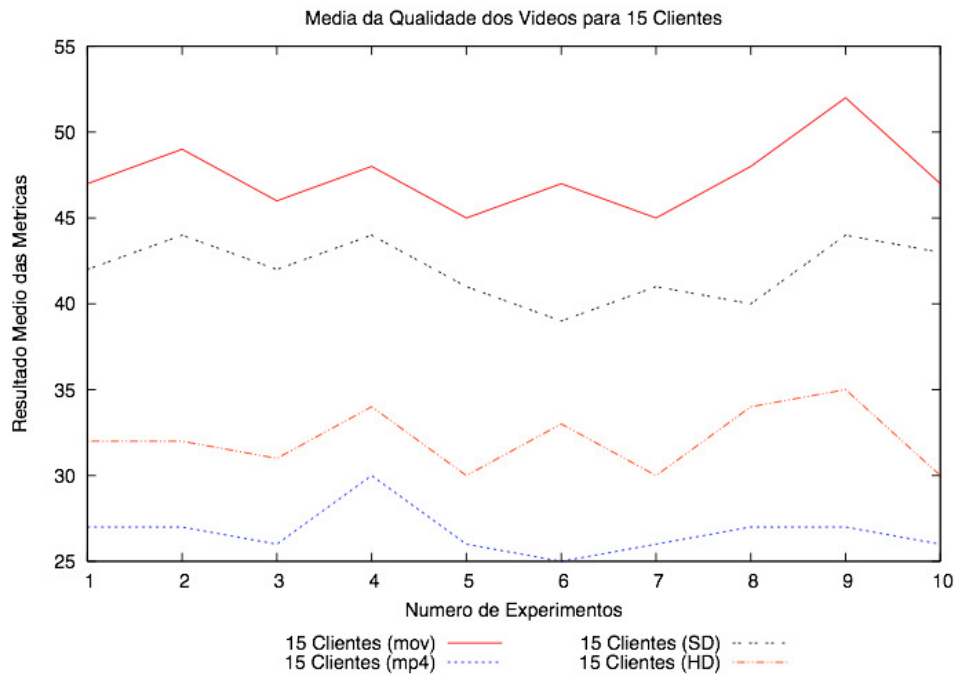


Figura 3.6: Média de qualidade da transmissão para 15 clientes.

Outro resultado encontrado através da avaliação do servidor foi o percentual de efeito dos fatores e de suas combinações. Na Tabela 3.4 demonstramos estes valores para cada uma das variáveis de respostas utilizadas. A primeira coluna demonstra todos os fatores e sua lista de combinações possíveis e as demais correspondem as variáveis de resposta. É possível notar que o percentual na perda de pacotes entre a comunicação de cliente e servidor ficou concentrado principalmente nos fatores referentes a resolução dos vídeos e a quantidade de clientes, junto da combinação do fator relacionado ao formato dos vídeos com a quantidade de clientes. Estes três fatores, são responsáveis por 82% de toda a influência sob o sistema para perda de pacotes.

Em relação aos *bytes* entregues ao cliente, diferentemente da métrica anterior, somente o fator referente a quantidade de clientes foi responsável por aproximadamente 59% dos efeitos sob estes dados a serem entregues dentro da aplicação. Além deste, outras duas combinações (formato do vídeo com a resolução e formato do vídeo com a quantidade de clientes) foram responsáveis por outros 25%, completando assim o conjunto dos fatores mais influentes a esta métrica.

Considerando o percentual de efeito dos fatores sob a qualidade de transmissão do vídeo, notamos que os valores estão melhor distribuídos em relação as métricas anteriormente apresentadas. O fator referente a resolução do vídeo e outras três combinações, que

<b>Fator</b>	<b>Pacotes Perdidos</b>	<b>Bytes P/ Cliente</b>	<b>% de Qualidade na Transmissão</b>
<b>A</b>	8,51%	6,86%	0,17%
<b>B</b>	18,76%	6,56%	29,13%
<b>C</b>	22,43%	58,60%	5,57%
<b>D</b>	0,01%	0,22%	0,00%
<b>AB</b>	4,27%	13,90%	23,23%
<b>AC</b>	40,63%	10,64%	26,07%
<b>AD</b>	0,07%	0,13%	0,12%
<b>BC</b>	5,02%	0,13%	12,09%
<b>BD</b>	0,01%	0,31%	0,01%
<b>CD</b>	0,07%	0,19%	0,02%
<b>ABC</b>	0,17%	1,37%	3,00%
<b>ABD</b>	0,02%	0,51%	0,27%
<b>ACD</b>	0,00%	0,18%	0,16%
<b>BCD</b>	0,01%	0,20%	0,00%
<b>ABCD</b>	0,02%	0,21%	0,14%

Tabela 3.4: Influência de efeito dos fatores e de suas combinações.

são o formato com a resolução, formato com a quantidade de clientes e a resolução com a quantidade de clientes, foram responsáveis por 90% de todos os efeitos sob a qualidade do vídeo.

### 3.7 Discussões

Deve ser frisado que a avaliação de desempenho foi realizada em cima de uma única tecnologia, porém diversas outras propuseram um ambiente adequado para a realização dos experimentos. Algumas destas tecnologias, além de prover este ambiente, possibilitaram a criação de diferentes situações para a avaliação deste trabalho.

Através da Tabela 3.4 podemos notar que os valores percentuais de efeito demonstram que o fator referente ao *browser*, bem como suas combinações ou não possuem influência, ou seu valor é desprezível em cada uma das três métricas. Além disso, podemos notar que o fator que possui maior percentual (somando as três métricas) é o que responde pela quantidade de clientes que acessam o servidor ao mesmo tempo. A partir dos resultados apresentados na Tabela 3.4, a avaliação foi voltada apenas para os fatores referentes ao



formato do vídeo, resolução, e quantidade de clientes.

Considerando a quantidade de clientes como o fator principal para esta avaliação, os gráficos apresentados na Seção 3.6 alternam os níveis do formato do vídeo e da resolução, fixando a quantidade de clientes, a fim de descobrir quais os demais fatores mais influentes na aplicação. Comparando os gráficos que demonstram os resultados para a métrica de pacotes perdidos podemos notar que quanto maior a quantidade de clientes, menor é a média de perda. Além disso, podemos ver que tanto na Figura 3.1 quanto na Figura 3.4 o nível do fator responsável pela resolução foi mantido (HD), porém em relação ao formato do vídeo para 5 clientes, o formato *.mov* foi o mais influente na perda de pacotes enquanto para 15, o formato *.mp4* foi o mais influente.

Em relação a métrica de *bytes* recebidos pelo cliente, podemos observar uma característica diferente da métrica anterior em relação ao formato para 15 clientes. Neste caso, os valores médios tanto para o *.mov* (35887 *bytes*) quanto para *.mp4* (38282 *bytes*) eram muito próximos, com uma ligeira vantagem do último citado. Com 5 clientes, a quantidade de *bytes* entregues é bem maior. O formato *.mov* e a resolução SD foram os que fatores que mais influenciaram na entrega destes dados. A resolução para 15 clientes, assim como para 5, obteve uma maior quantidade de *bytes* entregues na SD, porém em uma proporção diferente para cada uma das quantidades de clientes.

Analisando as Figuras 3.2 e 3.5 notamos que a resolução SD, em ambos, obteve o mesmo desempenho (proporcionalmente a quantidade de clientes). Já em relação ao formato, apesar da grande semelhança entre as médias do *.mov* e do *.mp4*, para 15 clientes, prevaleceu a alternância dos níveis deste fator para 5 clientes, que teve o *.mov* com maior quantidade média de bytes entregues e para 15, o *.mp4* foi ligeiramente melhor.

A terceira e última métrica utilizada para avaliar o desempenho da aplicação, foi a qualidade de transmissão dos vídeos. As Figuras 3.3 e 3.6 demonstram de forma clara que a qualidade de transmissão para 5 clientes é bem melhor para o formato *.mp4* e resolução SD, já para 15 clientes, apenas o formato é diferente (*.mov*) dentro das devidas proporções. Outro aspecto notado é que a resolução SD obteve um melhor desempenho em ambos os casos. Isto demonstra que além dos clientes o formato do vídeo é o fator de maior influência na qualidade do vídeo.

O comportamento para os formatos de vídeo *.mov* e *.mp4* de certa forma se inverteu para o caso de 5 e 15 clientes, conforme se pode perceber nas Figuras 3.1 e 3.4, 3.2 e

---

3.5 e também 3.3 e 3.6. No contexto dos experimentos realizados, não nos foi possível identificar exatamente o porquê deste comportamento, o qual será melhor investigado em trabalhos futuros.

## 4 Considerações Finais

Este trabalho apresenta a avaliação de desempenho do servidor de fluxo contínuo *Darwin*, através de requisições realizadas com o método URL ‘*rtsp://*’. Além do servidor multimídia, outras ferramentas como servidor HTTP e web browsers foram utilizadas.

Pode-se notar na apresentação dos resultados, principalmente na tabela que demonstra o percentual de efeito de cada um dos fatores sob o ambiente avaliado, que a quantidade de clientes utilizando o servidor foi o fator que mais influenciou os resultados nas três métricas utilizadas para medir este experimento.

A partir da observação referente a quantidade de clientes, decidiu-se encontrar dentre os demais fatores, quais deles associados ao número de clientes seria o mais influente. O que se pode notar através dos gráficos plotados nos resultados é que na maioria dos casos o fator que apresentou grande influencia no experimento foi o formato das mídias, diferentemente da resolução do vídeo, que obteve um baixo percentual de efeito em relação aos fatores anteriormente citados. O fator referente aos browsers não foi demonstrado em nenhum gráfico devido seus resultados terem sido considerados desprezíveis para o desempenho da aplicação.

Como um possíveis trabalhos futuros, a realização de uma nova avaliação de desempenho em cima deste mesmo ambiente, porém com aspectos de variação da banda com uma quantidade maior de clientes e o desenvolvimento do próprio servidor de fluxo contínuo para dispositivos móveis, além da avaliação de desempenho deste servidor.

## Referências Bibliográficas

FOUNDATION, A. S. *Apache*. 1995. [Online; acessado 20-Dezembro-2012]. Disponível em: <<http://httpd.apache.org>>.

GROUP, A.-V. T. W. et al. *RTP: A Transport Protocol for Real-Time Applications*. IETF, jan. 1996. RFC 1889 (Proposed Standard). (Request for Comments, 1889). Obsoleted by RFC 3550. Disponível em: <<http://www.ietf.org/rfc/rfc1889.txt>>.

INC., A. *Darwin Streaming Server Administrator's Guide (Manual)*. November 2002. [Online; acessado 15-Dezembro-2012]. Disponível em: <[http://manuals.info.apple.com/en\\_US/QuickTimeStreamingSrvrAdminGuide.pdf](http://manuals.info.apple.com/en_US/QuickTimeStreamingSrvrAdminGuide.pdf)>.

INC., A. *Safari*. 2003. [Online; acessado 20-Dezembro-2012]. Disponível em: <<http://www.apple.com/safari/>>.

INC., A. *QuickTime 7.1 User's Guide*. 2006. [Online; acessado 22-Dezembro-2012]. Disponível em: <[http://images.apple.com/quicktime/pdf/QuickTime7\\_User\\_Guide.pdf](http://images.apple.com/quicktime/pdf/QuickTime7_User_Guide.pdf)>.

INC., A. *Darwin Streaming Server*. 2007. [Online; acessado 10-Dezembro-2012]. Disponível em: <<http://dss.macosforge.org/>>.

INC., A. *QuickTime Broadcaster 1.5.3*. Janeiro 2009. [Online; acessado 23-Dezembro-2012]. Disponível em: <<http://support.apple.com/kb/DL764>>.

INC., A. *HTTP Live Streaming Overview*. 2011. [Online; acessado 23-Dezembro-2012]. Disponível em: <<http://developer.apple.com/resources/http-streaming/>>.

INC., G. *Google Chrome*. 2008. [Online; acessado 20-Dezembro-2012]. Disponível em: <<http://www.google.com/chrome>>.

JAIN, R. *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley, 1991. Disponível em: <<http://www1.cse.wustl.edu/~jain/books/perfbook.htm>>.

KUROSE, J.; ROSS, K. *Redes de computadores e a Internet: uma abordagem top-down*. [S.l.]: Pearson Addison Wesley, 2007.

PROJECT, V. *VLC user guide*. 2004. [Online; acessado 22-Dezembro-2012]. Disponível em: <<http://www.videolan.org/doc/vlc-user-guide/en/vlc-user-guide-en.html>>.

SANTOS, R. C. et al. Incremental validation of digital tv applications in nested context language. In: *Proceedings of the 10th European conference on Interactive tv and video*. New York, NY, USA: ACM, 2012. (EuroITV '12), p. 203–212. ISBN 978-1-4503-1107-6. Disponível em: <<http://doi.acm.org/10.1145/2325616.2325657>>.

# ANEXO A - Resultados da Métrica de Pacotes Perdidos

Métrica: Pacotes perdidos												
Apostas												
	1	2	3	4	5	6	7	8	9	10		
fxs.mov (1) hd (1) 15 Obj. (1) Chrome (1)	595	624	631	740	694	5402	654	5592	4439	623	5642,23	815,24
fxs.mov (1) hd (1) 15 Obj. (1) Safari (1)	634	587	617	540	601	6128	610	5462	5506	6618	6005,07	506,98
fxs.mov (1) hd (1) 5 Obj. (1) Chrome (1)	46388	43680	43641	46105	43839	38145	42454	45402	45722	37529	41844,52	3883,87
fxs.mov (1) hd (1) 5 Obj. (1) Safari (1)	45814	43279	41270	36540	44934	45575	42670	44938	42678	41983	43584,56	1582,24
goal.mov (1) sd (1)	7351	6994	6665	6238	6899	7652	6669	7039	6664	6321	6867,03	506,89
goal.mov (1) sd (1) 5 Obj. (1) Chrome (1)	683	6405	7251	6886	6676	6886	7739	6632	6421	6753	6886,05	506,39
goal.mov (1) sd (1) 5 Obj. (1) Safari (1)	2678	29463	30583	28474	23731	30461	25854	31023	30001	25953	28630,44	2065,11
goal.mov (1) sd (1) 5 Obj. (1) Chrome (1)	26640	23384	26602	32569	30795	31728	31018	27468	32608	32620	30408,32	2007,14
goal.mov (1) sd (1) 5 Obj. (1) Safari (1)	14689	13388	14689	14689	14689	14689	14689	14689	14689	14689	14689,00	14689,00
fxs.mpd (1) hd (1) 15 Obj. (1) Chrome (1)	20869	18355	21135	23451	23405	23445	20810	17865	21667	19884	20576,83	1993,84
fxs.mpd (1) hd (1) 5 Obj. (1) Chrome (1)	20869	21224	21330	21861	21486	22007	21779	22549	22118	22985	22159,68	298,28
fxs.mpd (1) hd (1) 5 Obj. (1) Safari (1)	18677	20317	21491	22140	22528	21933	21875	22816	22163	22310,60	455,06	327,39
goal.mpd (1) sd (1)	9485	9132	9476	9567	9410	9769	9460	9372	8859	9369	9365,71	827,39
goal.mpd (1) sd (1) 5 Obj. (1) Chrome (1)	9284	8128	9367	8878	9403	9544	9647	9834	9332	9616	9594,77	181,66
goal.mpd (1) sd (1) 5 Obj. (1) Safari (1)	0	136	148	179	192	152	336	127	95	22	146,44	116,57
goal.mpd (1) sd (1) 5 Obj. (1) Chrome (1)	126	30	57	23	165	14	48	393	409	424	275,32	227,06

Intervalo de Comparação	Méts +C	Méts -C
fxs.mov (1) hd (1) 15 Obj. (1) Chrome (1)	393	625
fxs.mov (1) hd (1) 15 Obj. (1) Safari (1)	393	625
fxs.mov (1) hd (1) 5 Obj. (1) Chrome (1)	46388	43680
fxs.mov (1) hd (1) 5 Obj. (1) Safari (1)	45814	43279
goal.mov (1) sd (1)	7351	6994
goal.mov (1) sd (1) 5 Obj. (1) Chrome (1)	683	6405
goal.mov (1) sd (1) 5 Obj. (1) Safari (1)	2678	29463
goal.mov (1) sd (1) 5 Obj. (1) Chrome (1)	26640	23384
goal.mov (1) sd (1) 5 Obj. (1) Safari (1)	14689	13388
fxs.mpd (1) hd (1) 15 Obj. (1) Chrome (1)	20869	18355
fxs.mpd (1) hd (1) 5 Obj. (1) Chrome (1)	20869	21224
fxs.mpd (1) hd (1) 5 Obj. (1) Safari (1)	18677	20317
goal.mpd (1) sd (1)	9485	9132
goal.mpd (1) sd (1) 5 Obj. (1) Chrome (1)	9284	8128
goal.mpd (1) sd (1) 5 Obj. (1) Safari (1)	0	136
goal.mpd (1) sd (1) 5 Obj. (1) Chrome (1)	126	30

$V_i = 27788,17$   
 $V_i/A_i = 1734,268$   
 $SS(abc...) =$   
 Variations =  
 $\% =$

Y	Qabd	Qbcd	Qacd	Qabd	Qabc	Qcd	Qbd	Qbc	Qad	Qac	Qab	Qd	Qc	Qb	Qa	q0
3642,23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6005,07	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
41844,52	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
43584,56	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
6867,03	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
6867,05	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
20630,44	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
30408,32	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2159,68	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
20576,83	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2210,60	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
9365,71	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
9594,77	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
146,44	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
275,32	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2978,95	2179,64	2179,64	739,13	2822,79	-894,95	-894,95	246,84	5496,97	-5700,71	-13668,75	-4428,87	-2082,87	-10150,19	-2082,87	-4428,87	-13668,75
18,18	136,23	136,23	46,20	-176,42	-559,68	-559,68	140,43	-3000,58	-85,29	-852,73	-2768,68	-128,93	-6343,95	-128,93	-2768,68	-852,73
554632,70	298295,91	298295,91	3044,88	49807,79	90194,86	90194,86	315518,12	14405665,37	201128,53	1166267,52	1226479,24	26963,68	64837056,47	26963,68	1226479,24	1166267,52
0,0001820	0,000189	0,000189	0,0001347	0,0001347	0,0005794	0,0005794	0,0000991	0,0501280	0,00070751	0,0625511	0,0472287	0,0009264	0,21480191	0,0009264	0,0472287	0,0625511
0%	0%	0%	0%	0%	0%	0%	0%	5%	0%	41%	4%	0%	22%	0%	9%	15%



