UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial

---

DISSERTAÇÃO
apresentada à UTFPR
para obtenção do grau de

MESTRE EM CIÊNCIAS

por

MARCIA KOTELOK

MODELAGEM DE TRÁFEGO DE SERVIDOR WEB POR CLASSIFICAÇÃO DE CONTEÚDO

Banca Examinadora:

Presidente e Orientadora:	
Profa. Dra. Keiko Verônica Ono Fonseca	UTFPR
Examinadoras:	
Profa. Dra. Lúcia Valéria Ramos Arruda	UTFPR
Profa. Dra. Fátima de Lima Procópio Duarte Figueiredo	PUC-MG

Curitiba (PR), março de 2006

MARCIA KOTELOK

MODELAGEM DE TRÁFEGO DE SERVIDOR WEB POR CLASSIFICAÇÃO DE CONTEÚDO

Dissertação apresentada ao Curso de Pós Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de "Mestre em Ciências" – Área de Concentração: Telemática.

Orientadora: Profª. Drª. Keiko Verônica Ono Fonseca

CURITIBA
2006

Dedicatória

Dedico este trabalho

Ao meu marido:
Por sempre acreditar em mim

Aos meus pais:
Por me passarem os valores que me deram condições de realizar muito mais do que jamais ousei sonhar.

Agradecimentos

Agradeço ao UTFPR pela oportunidade e pela infra-estrutura disponibilizada, bem como por realizar um sonho da adolescência, que foi estudar nesta instituição.

Agradeço à Profa. Dra. Keiko Verônica Ono Fonseca pela oportunidade, pelo acompanhamento e direcionamento durante todas as etapas desta dissertação de mestrado.

Agradeço especialmente ao Mestre e Doutorando Carlos Marcelo Pedroso pela honra em dividir sua idéia comigo, por todo apoio, auxílio e incentivo durante o desenvolvimento do presente trabalho.

Agradeço às Professoras Dras. Lúcia Valéria Ramos Arruda e Fátima de Lima Procópio Duarte Figueiredo, por terem aceitado participar como membro da banca de avaliação desta dissertação de mestrado.

Agradeço a Profa. Dra. Cristina Duarte Murta por gentilmente disponibilizar os dados do IRCache que foram alvo de estudo desta dissertação.

Agradeço ao Professor Dr. Joel Correa da Rosa pelo apoio do Laboratório de Estatística da Universidade Federal do Paraná ao presente trabalho.

Agradeço aos diversos colegas que dividiram comigo horas e mais horas de estudo durante as disciplinas do curso.

E principalmente, ao meu marido por nunca deixar de acreditar que seria possível a conclusão deste projeto.

Meus agradecimentos

Marcia Kotelok

Sumário

Lista de Tabelas

Lista de Gráficos

Lista de Figuras

Lista de Siglas

ARPANET – Advanced Research Projects Network

DARPA – Defense Advanced Research Project Agency

DNS – Domain Name System

DSL – Digital Subscriber Line

FARIMA – Fractional Autoregressive Integrated Moving Average

FBM - Fractional Brownian Motion

FTP - File Transfer Protocol

GIF – Graphics Interchange Format

HTML –Hypertext Markup Language

HTTP – Hypertext Transfer Protocol

IANA – Internet Assigned Numbers Authority

ITA - Internet Traffic Archive

ICMP – Internet Control Message Protocol

ISDN – Integrated Service Digital Network

IP – Internet Protocol

JPEG – Join Photographic Experts Group

LAN – Local Area Network

LRD – Long Range Dependence

MIME - Multipurpose Internet Mail Extensions

MMPP – Processo de Poisson Modulado por uma Cadeia de Markov

MPEG – Moving Picture Experts Group

NLANR - National Laboratory for Applied Network Research

NNTP – Network News Transport Protocol

NS – Network Simulator

PIBIC - Programa Institucional de Bolsas de Iniciação Científica

QoS – Quality of Service

RFC – Request for Comment

SMTP – Simple Mail Transfer Protocol

SURGE – Scalable URL Reference Generator

TCP – Transmission Control Protocol

TELNET – Telnet Protocol

UDP – User Datagram Protocol

URL – Uniform Resource Locator

XML – Extensible Markup Language

WAN – Wide Area Network

WCA - World Wide Web Consortium's Web Characterization Group Repository

WWW – World Wide Web

Resumo

Este trabalho está inserido no contexto de melhoria do desempenho de servidores Web e foi desenvolvido no Laboratório de Comunicação de Dados do CPGEI/UTFPR. O objeto de estudo desta dissertação de mestrado é a modelagem do tráfego de saída de servidores Web. A nova proposta de modelagem baseia-se na análise da transmissão dos objetos acessados (arquivos transmitidos) durante uma sessão do usuário. A correta caracterização da distribuição do tamanho destes arquivos influencia na precisão da representação do tráfego de saída do servidor Web em foco. Este trabalho também discute as técnicas estatísticas aplicadas para esta caracterização. Os resultados obtidos permitem estabelecer um modelo para geração de tráfego sintético para servidores Web. Espera-se que este novo modelo contribua no sentido de reduzir a complexidade da geração de tráfego Web sintético, e que possa ser utilizado para fins de planejamento da capacidade de servidores e demais componentes da estrutura física (roteadores, circuitos de comunicação, etc).

Abstract

This work refers to the Web performance research area and was developed at the Data Communication Laboratory of the CPGEI/UTFPR. The main subject is the output traffic modeling of a web server. The new modeling proposal takes in account the accessed objects (transmitted files) during an user session. For a specific web server, the characterization of the size distribution of such files is fundamental to the model correctness. We discuss the adequacy of statistical techniques to this characterization. The achieved results could be applied to develop synthetic traffic generators for web server output, a fundamental tool for server design, capacity planning as well the deployment of network elements (cache servers, routers, etc.). The model validation also shows a reduced complexity when compared to the models found in the studied literature.

1 Introdução

Através do rápido desenvolvimento dos meios de comunicação e das tecnologias de rede, a Internet se tornou o mais difundido sistema de comunicação do mundo. É atualmente o mais importante e rápido sistema de troca de informações (voz, vídeo, emails, rádio e televisão online, entre outros serviços). Toda essa gama de opções tem alterado as relações entre as pessoas, as empresas e o mundo em geral.

No entanto, o uso atual da Internet está distante de sua concepção original. Principalmente no que diz respeito às características do tráfego gerado pelas transações entre os usuários e os destinos acessados. A variedade e a variação do conteúdo trafegado levam a problemas de desempenho. Estes problemas são alvo de estudos cujo objetivo é a proposição de soluções que eliminem ou amenizem seus impactos.

A Web ou World Wide Web [RFC1580] é a aplicação para acesso fácil e prático ao universo de informações disponíveis na Internet. O tráfego de rede da Internet, que já foi 50% baseado em FTP (File Transfer Protocol) [RFC959] no início da década de 90, hoje é 70% baseado em Web [KR01] [HWA05], como mostra a figura 1.1. Estes dados demonstram a grande importância da modelagem do tráfego dos servidores Web, no cenário atual do tráfego de rede da Internet.

	% bytes	% pacotes	% fluxos
HTTP	75	70	75
SMTP	5	5	2
FTP	5	3	1
NNTP	2	1	1
TELNET	1	1	1
DNS	1	3	18

Tabela 1.1 – Distribuição do Tráfego Internet por Aplicação [HWA05]

A utilização de geradores de tráfego sintético costuma ser uma das principais ferramentas nos estudos de tráfego de rede [CAO04]. Estes geradores injetam tráfego sintético

na rede de acordo com um modelo definido e permitem simulações de situações reais. Estas simulações, por sua vez, propiciam planejamento de capacidade e análise de desempenho dos servidores e da rede e seus elementos. Dentro deste contexto, os modelos para geração de tráfego sintético de servidores Web são componentes essenciais para simulação da Internet.

Esta dissertação, contribui no cenário Web com uma nova abordagem para modelagem de tráfego de rede, capaz de funcionar como uma alternativa aos modelos atuais. O modelo proposto é denominado Modelo por Classificação de Conteúdo e avalia o conteúdo do tráfego de rede do servidor Web em termos dos tipos dos arquivos transmitidos.

A estrutura desta dissertação foi organizada da seguinte forma:

A evolução do tráfego de rede, desde o seu nascimento na década de 60 até a forma atual, e diferentes características de tráfego durante este processo evolutivo são abordados no capítulo 2. O ambiente Web e seus componentes também são apresentados de forma a embasar a compreensão do experimento.

As diversas modelagens utilizadas para tráfego de rede são descritas e analisadas no capítulo 3, onde são apresentadas as características e aplicações de cada um dos modelos.

O capítulo 4 apresenta o modelo proposto para a geração de tráfego sintético e descreve sua aplicação (Modelo por Classificação de Conteúdo). Em seguida são descritas as amostras utilizadas na validação do presente modelo e os resultados obtidos com a aplicação do modelo aqui proposto. São destacadas as vantagens analíticas e matemáticas sobre os modelos existentes descritos no capítulo 3.

O estudo estatístico gerado para a validação matemática do modelo proposto é detalhado no capítulo 5, juntamente com as ferramentas estatísticas utilizadas para a obtenção dos resultados.

As conclusões da presente dissertação e as propostas de trabalhos futuros, a partir desta nova abordagem, são foco do capítulo 6.

2 Tráfego de rede

Para entender as características do tráfego Web atual, se faz necessária à compreensão da estrutura que permite que o usuário efetue a consulta a determinado documento no mundo Web. Para tanto, são apresentados a seguir os conceitos básicos para o entendimento desta dissertação e a evolução das redes de comunicação.

2.1 O início do tráfego de rede à arquitetura TCP/IP

No final dos anos 60 a agência de projeto e pesquisa do departamento de defesa americano (DARPA - Defense Advanced Research Project Agency) iniciou um projeto experimental chamado ARPANET (Advanced Research Projects Network). Esta rede tinha por objetivo permitir aos usuários do governo americano compartilharem os então escassos e caros equipamentos entre as diversas áreas existentes. Esta rede permitiu o início do compartilhamento de arquivos, trocas de informações e principalmente o desenvolvimento de pesquisas utilizando-se o compartilhamento de computadores remotos. [ALB98]

A transformação da ARPANET na Internet atual inicia com o desenvolvimento do TCP/IP (Transmission Control Protocol/Internet Protocol) [RFC 793] [RFC791]. Antes do TCP/IP as redes podiam somente comunicar-se internamente. O TCP/IP é responsável pela habilitação da troca de dados entre diferentes redes sem efetuar trocas de configuração entre elas. Permite endereçamento global e localização de computadores através de seus endereços numéricos sem qualquer correlação com sua localização geográfica.

A ARPANET original cresceu dentro da Internet, baseada na idéia de múltiplas redes independentes de forma arbitrariamente construídas. O ponto chave reside no conceito de uma arquitetura de rede aberta, com base em quatro tópicos principais:

- Cada rede distinta tem autonomia própria, desta forma, as informações internas não podem ser requisitadas de nenhuma outra rede sem antes se conectar a rede original;

- Comunicação seria o ponto chave, se um pacote não foi entregue no destino final ele deveria ser facilmente retransmitido de sua origem;

- Gateways e routers seriam utilizados para interligação entre as diversas redes;

- Independente do fluxo dos pacotes transmitidos, nenhuma informação seria retida pelos gateways utilizados. A simplicidade deste modelo evita adaptações complicadas e recuperação de vários modelos de falhas, impossibilitando assim, um controle global no nível operacional.

Uma das grandes vantagens do TCP/IP, parte da razão do seu sucesso, é que os protocolos envolvidos na arquitetura permitem a comunicação entre computadores independentemente do sistema operacional utilizado [TAN97].

O TCP/IP é um conjunto de protocolos de interconexão de sistemas, executado em ambiente aberto, utilizando uma arquitetura de quatro camadas, definidas como:

- **Camada de Aplicação:** fornece a interface do usuário de rede na forma de aplicativos e serviços de rede. Exemplos de protocolo: Simple Mail Transfer Protocol (SMTP) [RFC821], File Transfer Protocol (FTP) [RFC959] e Telnet [RFC854];

- **Camada de Transporte:** responsável por organizar as mensagens recebidas de camadas mais altas nos segmentos, por controlar os erros e pelo controle do fluxo fim-a-fim. Exemplos de protocolo: TCP e User Datagram Protocol (UDP) [RFC768];

- **Camada de Rede:** responsável pelo endereçamento dos equipamentos e pela transmissão (ou roteamento) dos dados em redes diferentes. Exemplos de protocolo: IP e Internet Control Message Protocol (ICMP) [RFC792];

- **Camada de Interface de rede:** responsável por controlar o fluxo de dados e organizar os bits da camada física.

Dentre estas quatro camadas e seus componentes merece destaque o protocolo IP devido a sua importância vital para o funcionamento das redes e conseqüentemente a ampliação da utilização da Web mundialmente.

O IP é um dos protocolos mais importantes da arquitetura TCP/IP e suas principais características são:

- Roteamento de pacotes, o que permite que estes possam chegar com rapidez e integridade ao endereço de destino;
- Endereçamento para cada equipamento conectado a um rede, fornecendo assim um identificador que permite que cada um deles possa se acessado de forma única e independente.

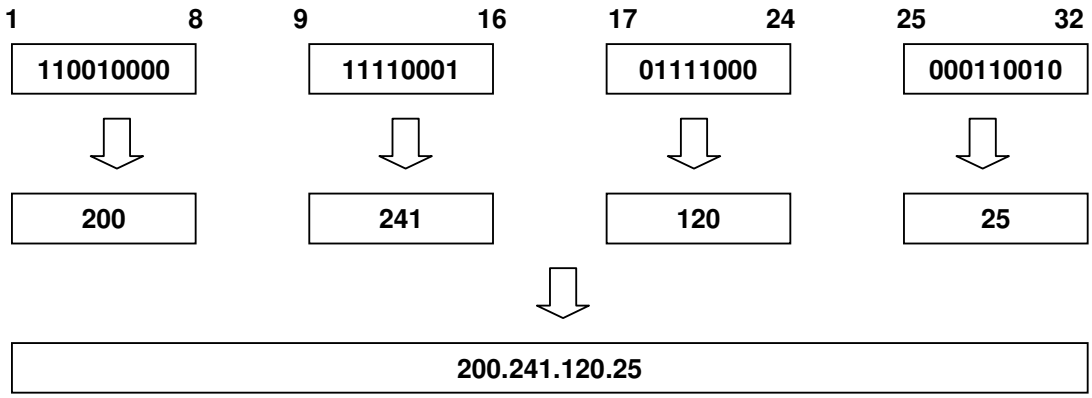


Figura 2.1 – Endereçamento IP

Este endereçamento é um número composto por 32 bits, representado por quatro campos de números decimais inteiros, que variam de 0 a 255, como descrito na figura 2.1.

2.2 A Web

A Web, também conhecida como WWW (World Wide Web), permite o acesso a documentos que estão armazenados em servidores espalhados por todo o mundo. Sua popularidade deve-se a facilidade em utilizar sua interface gráfica e pela imensa gama de informações que propicia a quem “navega na Internet”.

A Web teve início em 1989 na Europa [TAN97] como forma de compartilhamento do material de pesquisas sobre física de partículas. Foi um modo de rapidamente disponibilizar o acesso às informações que interessavam a todos e para as quais a distância geográfica dificultava a troca.

Primeiramente acessada em modo texto com o Hypertext (1991) evoluiu para o modo gráfico, tendo como primeira interface o Mosaic em 1993, que deu origem ao Netscape, e este a outros sucessivamente. Estes softwares foram batizados de browser.

A Web é basicamente um sistema cliente-servidor, como podemos ver na figura 2.3. No lado cliente, o usuário vê a Web como uma coleção de documentos ou páginas, que contêm ligações com outras páginas. Os browsers criam um cache local para guardar as páginas acessadas. Desta forma ao voltar à página anterior, esta não precisa ser novamente transmitida pela rede porque ainda está armazenada localmente.

No lado servidor o browser escuta a porta 80 do TCP e nela estabelece a conexão para receber o pedido do cliente e devolver a resposta. O protocolo utilizado é o HTTP (Hypertext Transfer Protocol) [RFC1945] e depois a conexão é liberada, na versão 1.0, ou mantida, enquanto houverem dados a serem transmitidos na versão 1.1. Na versão 1.0 do protocolo HTTP para cada objeto de uma página, nova conexão é estabelecida. Assim é grande o número de conexões para a transmissão de uma única página, mas a implementação fica simplificada. Na versão 1.1 [RFC2616] do HTTP, a conexão persiste até que todos os objetos de uma página sejam transmitidos.

Os servidores proxy têm por objetivo falar HTTP com o browser e com outros servidores onde as informações buscadas estão armazenadas. Muitas vezes também são usados como cache. Da mesma forma que o browser no cliente, o proxy no servidor armazena as páginas que passam por ele (cache). Quando uma página é pedida, o proxy verifica se já a possui, em caso afirmativo verifica se a mesma ainda é válida e a transmite para o solicitante. Em caso negativo a busca é encaminhada para a Web.

O protocolo padrão da Web é o HTTP, que é de fácil implementação. Primeiramente é estabelecida uma conexão entre o cliente e o servidor, e depois a página é solicitada. O protocolo HTTP 1.0 informa um código de status (cuja lista consta nos anexos) e uma mensagem MIME - Multipurpose Internet Mail Extensions [RFC822].

O HTTP é um protocolo do tipo requisição/resposta que foi projetado para transferir arquivos da aplicação Web. A página Web pode ser vista como um objeto HTTP composto de

um ou mais arquivos HTML (Hyper Text Markup Language) [RFC1942], juntamente com outros elementos que possam vir a compor uma página Web (imagens, sons, animações etc).

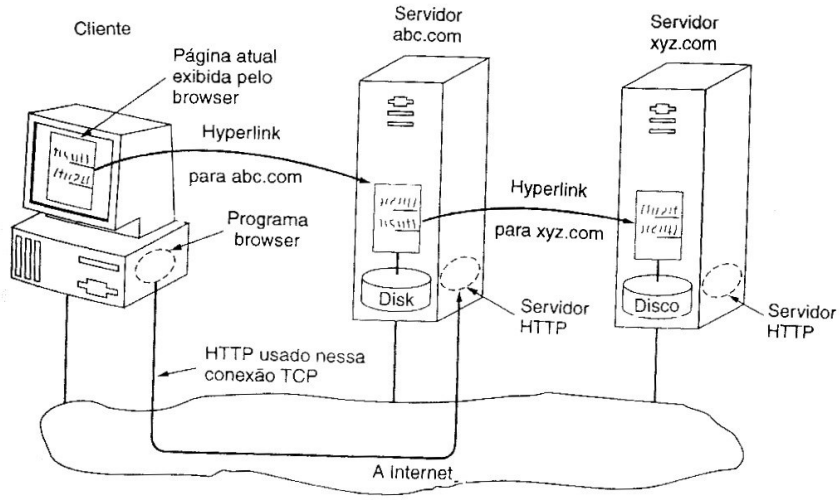


Figura 2.2 - Componentes da Web

O campo “content-id” do protocolo HTTP define três categorias para os arquivos da Web, listadas e explicadas abaixo. Estas categorias dizem respeito ao grau de suporte que o protocolo dará a estes tipos de arquivos em suas futuras versões.

- **conhecido:** o formato do arquivo é reconhecido, mas não se pode garantir o suporte integral;
- **não suportado:** o formato do arquivo não é reconhecido; será listado como "application/octet-stream", ou seja Desconhecido;
- **suportado:** o formato do arquivo é suportado integralmente.

O termo “Suportado” significa que este formato será mantido válido no futuro, usando qualquer combinação de técnicas apropriada (como migração, emulação, etc.). Para os formatos suportados pode-se optar por transformar o conjunto de arquivos de uma versão atual do formato para uma futura versão. Diz-se também que um formato de arquivo é “suportado”, se existir documentação e informação suficiente para saber como ele funciona. Mas infelizmente, isto significa que formatos proprietários, sobre os quais não exista essa informação disponível publicamente, não podem ser suportados pelo repositório. É provável que formatos

proprietários extremamente populares (como os .doc, .xls, .ppt da Microsoft) continuem a ser utilizáveis no futuro, simplesmente porque a sua ampla utilização torna mais provável a existência de ferramentas de migração de versões e conversão.

Em algumas situações o formato não é reconhecido. Sabe-se que há sempre mais formatos do que a viabilidade de suportá-los. Se um formato não é identificado, ele será registrado como “não suportado”, ou "application/octet-stream".

A tabela 2.1 referencia os tipos de arquivos definidos de acordo com o MIME [RFC822]. Os campos representam:

- **Descrição** é o nome que a maior parte das pessoas usa para designar o formato;
- **Extensões** são as extensões ao nome de arquivos mais freqüentes (a parte após o ponto). As extensões não são sensíveis a maiúsculas e minúsculas, por isso, quer teste.XML ou teste.xml serão reconhecidos como XML (Extensible Markup Language) [RFC3076];
- **Nível** é o nível de suporte no repositório para cada formato (suportado; conhecido e não suportado).

O MIME também divide em cinco tipos todos os objetos da Web, estes por sua vez se subdividem em inúmeros outros. A seguir os tipos macros serão apresentados, de forma a caracterizar o conteúdo transmitido pela rede quando mencionado cada um deles.

Text - indica principalmente o envio de texto puro. É formado por caracteres mas não inclui o subtipo "text/plain", que é a generalização de texto sem definição conhecida de formato, fonte, atributos, instruções, diretivas de interpretação ou conteúdo demarcado como o tipo Texto provê. “Text/plain” é apenas uma seqüência linear de caracteres, possivelmente interrompida por linhas ou páginas. Incluído como “plain text” há diversos formatos também conhecidos como "rich text". Uma característica interessante deste tipo é que há várias extensões que podem se lidas sem um software específico. Isto é útil quando, num nível mais alto, precisa-se distinguir entre dados de imagens, áudio ou texto de um formulário ilegível. Na ausência de um software apropriado para interpretar o formato, é razoável mostrar o subtipo texto ao usuário, mas o mesmo não deve ser empregado para tipos de dados que não sejam texto. Todos os formatos de dados textuais devem ser representados usando o subtipo "text";

TIPO MIME	DESCRIÇÃO	EXTENSÃO	NÍVEL
application/octet-stream	Desconhecido	qualquer não listado	não suportado
application/pdf	Adobe PDF	Pdf	suportado
text/xml	XML	Xml	suportado
text/plain	Text	txt, asc	suportado
text/html	HTML	htm, html	suportado
application/msword	Microsoft Word	Doc	conhecido
application/vnd.ms-powerpoint	Microsoft Powerpoint	Ppt	conhecido
application/vnd.ms-excel	Microsoft Excel	Xls	conhecido
application/marc	MARC	marc, mrc	suportado
image/jpeg	JPEG	jpeg, jpg	suportado
image/gif	GIF	Gif	suportado
image/png	image/png	png	suportado
image/tiff	TIFF	tiff, tif	suportado
audio/x-aiff	AIFF	aiff, aif, aifc	suportado
audio/basic	audio/basic	au, snd	conhecido
audio/x-wav	WAV	wav	conhecido
video/mpeg	MPEG	mpeg, mpg, mpe	conhecido
text/richtext	RTF	Rtf	suportado
application/vnd.visio	Microsoft Visio	vsd	conhecido
application/x-filemaker	FMP3	Fm	conhecido
image/x-ms-bmp	BMP	bmp	conhecido
application/x-photoshop	Photoshop	psd, pdd	conhecido
application/postscript	Postscript	ps, eps, ai	suportado
video/quicktime	Video Quicktime	mov, qt	conhecido
audio/x-mpeg	MPEG Audio	mpa, abs, mpega	conhecido
application/vnd.ms-project	Microsoft Project	mpp, mpx, mpd	conhecido
application/mathematica	Mathematica	Ma	conhecido
application/x-latex	LateX	latex	conhecido
application/x-tex	TeX	Tex	conhecido
application/x-dvi	TeX dvi	dvi	conhecido
application/sgml	SGML	sgm, sgml	conhecido
application/wordperfect5.1	WordPerfect	wpd	conhecido
audio/x-pn-realaudio	RealAudio	ra, ram	conhecido
image/x-photo-cd	Photo CD	pcd	conhecido

Tabela 2.1 – Formatos de arquivos Web [RFC822]

- **Image** - indica que o conteúdo do corpo enviado contém uma imagem. O formato inicialmente utilizado é o "jpeg" (Join Photographic Experts Group) [RFC2158]. A lista dos subtipos de imagem está registrada no IANA (Internet Assigned Numbers Authority) e são descritos na [RFC2048]. Porém esta lista não pára de crescer. Um dos mais comum é o GIF (Graphics Interchange Format) [RFC2158].

- **Audio** - indica o formato de áudio. Ainda não há um consenso do formato padrão de áudio para ser usado pelos computadores. Há uma pressão pela obtenção de formato que garanta um comportamento capaz de prover interoperabilidade;

  O subtipo "basic" foi especificado para encontrar um mínimo denominador comum nos requisitos do formato áudio. É esperado que formatos mais ricos com mais qualidades e/ou menor largura de banda serão definidos em um documento futuro. O conteúdo do subtipo "audio/basic" é um canal simples codificado usando 8bit ISDN (Integrated Service Digital Network) [RFC3057] e uma taxa de 8000 Hz;

- **Vídeo** - indica que os conteúdos são imagens, possivelmente coloridas e com som coordenado. O termo vídeo é usado no mais genérico sentido, sem referência a nenhum formato ou tecnologia particular, e não significa que não hajam subtipos com desenhos animados codificados compactamente. O subtipo "mpeg" referencia vídeo codificado de acordo com o padrão MPEG (Moving Picture Experts Group) [RFC3003];

- **Application** - é usado para dados discretos que não combinam com qualquer outra categoria, é formado principalmente por dados que são processados por algum tipo de programa aplicativo. Tipos de aplicações esperadas são: transferência de arquivos, emails e linguagens.

Os tipos desconhecidos de imagem, áudio e vídeo, assim como as aplicações genéricas são tratadas como "application/octet-stream" (que são seqüências binárias arbitrárias).

A eficiência da arquitetura da Internet possibilitou que o uso das redes fosse amplamente difundido e que novas aplicações surgissem. Dentre elas a Web sem dúvida foi a que revolucionou o mundo das comunicações, do comércio mundial e principalmente da divulgação das informações. A revolução no tráfego das redes de comunicação de dados fez surgir o desafio de buscar novas e melhores formas de modelagem do tráfego, que permitam planejar e operar eficientemente os recursos de rede.

3 Caracterização do tráfego de rede

Com a popularização da Internet e a multiplicação dos serviços disponíveis aos usuários houve um aumento na demanda por acessos à Internet. No início muitos acessos eram discados porém está havendo uma substituição gradativa destes por serviços que permitem taxas melhores de transmissão bem como alta disponibilidade e confiabilidade nos serviços prestados, como por exemplo: DSL (Digital Subscriber Line).

Porém, devido às características e peculiaridades das aplicações que trafegam na Internet, não basta mais apenas aumentar a banda dos circuitos de comunicação. O alto consumo prejudica as aplicações que exigem garantia mínima de qualidade. Hoje se tornaram necessárias garantias de alocação de largura de banda (bandwidth), controle de latência (delay) e da variação do atraso (jitter). O atendimento destes e outros requisitos são objetos de estudo de QoS (Quality of Service). Uma rede garante QoS quando consegue atender parâmetros mínimos para determinado fluxo de informações que serão mantidos sob quaisquer circunstâncias. Normalmente estes parâmetros mínimos estão associados às exigências das aplicações. No entanto, a provisão de QoS requer o conhecimento do padrão de tráfego a ser gerado pela aplicação.

As ações de identificar as características do tráfego Web e avaliar novas técnicas para melhorar seu desempenho são de extrema importância para o projeto de aplicações de Web “sites”, gerenciamento de “proxies” e servidores da Web, além é claro da própria rede. Para este trabalho são necessárias basicamente três etapas: monitorar a rede; armazenar as coletas; e analisar os dados.

Um modelo de caracterização de tráfego consiste numa coleção de parâmetros que representam os pontos chave da carga que afetam a localização dos recursos e o desempenho do sistema. Estes modelos devem identificar problemas de desempenho, proporcionar a análise de componentes (por exemplo: avaliação de um novo proxy a ser comprado) e principalmente planejamento de capacidade.

Os modelos de caracterização de tráfego devem retratar a natureza do ambiente em que estão envolvidos e identificar as métricas que afetam seu desempenho. Segundo [KRI01] são três as categorias que podem fornecer parâmetros para esta caracterização:

1. as características das mensagens do protocolo HTTP;
2. as propriedades dos recursos Web;
3. e o comportamento do usuário.

A tabela 3.1 mostra os parâmetros associados a cada uma das 3 categorias:

CATEGORIA	PARÂMETRO
Protocolo	Método de pedido Código de resposta
Recursos	Tipo de arquivo Tamanho do recurso Tamanho da resposta Popularidade Freqüência de modificação Localidade Temporal Número de recursos embutidos
Usuário	Tempo entre sessões Número de “clicks” por sessão Tempo entre pedidos

Tabela 3.1 – Exemplos de parâmetros para modelos da Web

Para tentar garantir que um modelo represente o tráfego real, os parâmetros do modelo devem possuir certas propriedades [KRI01]:

- Não estarem ligados a um sistema ou outras características específicas que possam vincular e limitar os resultados a um cenário em particular. O contrário pode ser verdade: um modelo de geração de carga pode avaliar as métricas que particularizam o sistema;
- O parâmetro deve representar a carga no nível apropriado de detalhe que permita a avaliação do sistema.

Os elementos estatísticos: média, mediana e variância dos dados em análise capturam propriedades básicas dos parâmetros utilizados para caracterização do tráfego Web. Já as distribuições de probabilidade mostram a variabilidade dos dados. O ajuste de dados coletados de tráfego real a uma distribuição de probabilidade e o teste deste ajuste tem sido uma área de pesquisa bastante ativa durante anos [LAW99].

A seguir alguns modelos de tráfego de rede serão discutidos.

3.1 Modelos de Tráfego de rede

Os modelos utilizados para modelagem de tráfego de rede podem ser classificados em duas grandes categorias [DRA03]: modelos baseados em amostras e modelos analíticos.

O modelos baseados em amostras são baseados em dados reais e a principal vantagem do uso deste modelo é que eles são mais fáceis de usar e implementar. A desvantagem, no entanto, reside no fato de que um trabalho considerável deverá ser empregado para se efetuar as coletas, fazendo com que o modelo seja de adaptação mais difícil para a troca de cenários. Esta amostra pode ser vista como uma fotografia estática de um dado momento da rede. Assim sendo, o modelo pode não ter uma boa confiabilidade para identificar as causas do comportamento de um sistema, mas normalmente tem uma boa aceitação na simulação de um conjunto específico de condições.

Os modelos analíticos usam modelos matemáticos para a geração de simulação. Modelos matemáticos mais flexíveis são capazes de gerar diferentes cenários através da variação ou troca das características de determinado escopo. Eles podem, no entanto, serem de difícil construção. Primeiro é necessário identificar as principais características do tráfego, segundo estas características devem ser empiricamente mensuráveis para finalmente poder se escolher o melhor modelo matemático que será alimentado com os dados. O modelo analítico fornece uma maior flexibilidade no que diz respeito às variações e trocas nas condições da simulação do que os modelos baseados em amostras.

Dentre os vários modelos empregados para a representação do tráfego de rede, alguns merecem destaque e serão descritos a seguir.

3.1.1 Modelo de Poisson

Inicialmente a caracterização do tráfego tinha como base a rede de telefonia, introduzindo o modelo de Poisson [ADA97], onde o número de chamadas e o tempo entre as chegadas das requisições eram exponencialmente distribuídas, e as chamadas demoravam um tempo exponencial para serem completadas. Este modelo tem propriedades analíticas úteis para modelagem do tempo de rede.

Com base nesta distribuição, ao longo do tempo, têm sido propostos modelos e mecanismos para a modelagem do tráfego da rede. Estes estudos tiveram mudanças de rumo significativas com o advento da Internet que modificou drasticamente o comportamento do tráfego [PAX95].

Um processo Poisson é simplesmente um processo de entrada aleatória exponencialmente distribuída na escala de tempo [BAR99]. As características mais marcantes deste processo são sua aleatoriedade e o fato de não ter memória. Este modelo é eficiente na descrição, tanto do evento de chegada do pedido, quanto do seu tempo de atendimento, definido somente através de um parâmetro. Uma vez que a agregação de múltiplos processos de Poisson também converge para uma distribuição Poisson, ele também pode ser utilizado na concatenação de processos mais complexos.

O processo de Poisson pode ser caracterizado como um processo de renovação em que os intervalos entre chegadas $\{A_n\}$ são exponencialmente distribuídos com parâmetro λ :

$$P\{A_n \leq t\} = 1 - \exp(-\lambda_t)$$

Onde: P = probabilidade; A_n = Processo de Intervalos entre Chegadas; t = tempo; exp = distribuição Exponencial; λ = parâmetro da distribuição Exponencial.

O número de chegadas em intervalos disjuntos é independente. O processo de Poisson apresenta algumas características analíticas importantes como:

- A superposição de processos de Poisson resulta em um novo processo de Poisson cuja taxa é a soma das taxas individuais;

- É um processo sem memória, o que facilita os problemas de resolução de filas que envolvem este processo [FRO94].

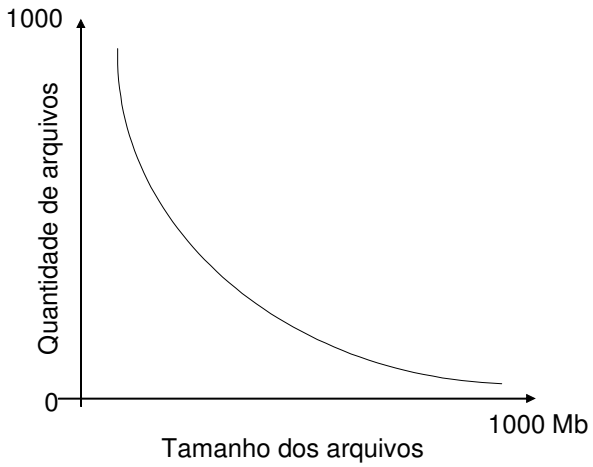


Figura 3.1 – Relação entre Tamanho e Quantidade dos arquivos transmitidos na Internet

Estudos realizados por [PAX95] apresentaram as falhas do modelo de Poisson na caracterização correta de grandes volumes de tráfego de rede da Internet. Os autores deste estudo afirmam que uma sessão, ou chamada, iniciada pelo usuário para um acesso remoto ou para transferência de arquivos poderia utilizar a premissa de Poisson, mas a tráfego da conexão desvia-se consideravelmente da distribuição de Poisson.

Uma das principais características do tráfego Web, que levaram à falha na utilização do Modelo de Poisson, é a relação entre o tamanho e a quantidade dos arquivos transmitidos na Internet. Esta proporção mostra que há muitos arquivos pequenos e em contrapartida poucos arquivos imensos [PAX95], conforme ilustrado na figura 3 1.

3.1.2 Modelo Markoviano (Modulado por uma Cadeia de Markov)

No contexto de modelagem de tráfego em redes, há uma grande utilização da Teoria de Filas, onde o tráfego é oferecido a uma fila ou a uma rede delas e várias medidas de desempenho são efetuadas, através de metodologias analíticas ou simulações computacionais [ADA97].

Os sistemas de telecomunicação têm sido tradicionalmente modelados como sistemas estocásticos utilizando-se modelos de Markov. Segundo a definição, o próximo estado de um processo Markoviano pode somente depender de seu estado atual. Nenhuma informação sobre as seqüências dos estados visitados anteriormente pode afetar a próxima transição [FRO94].

Os modelos de Markov introduzem dependência entre os elementos de uma seqüência **{**A_n**}**. Conseqüentemente eles podem "capturar" uma rajada do tráfego porque a auto-correlação da seqüência é diferente de zero.

Considerando uma cadeia de Markov de parâmetro contínuo t e com espaço de estados discreto M:

$$M = \{M(t)\}_{t=0}^{\infty} \qquad 0 < t < \infty$$

Neste caso, M comporta-se como se segue: a cadeia permanece no estado i por um tempo exponencialmente distribuído com parâmetro λ_i, que depende apenas de i . A cadeia então muda para o estado j com probabilidade p_{ij} , de acordo com a matriz de taxas infinitesimais de probabilidade:

$$P = [p_{ij}]$$

Em um modelo simples, cada mudança de estado indicaria uma chegada, então os intervalos entre chegadas seriam exponencialmente distribuídos e os parâmetros de taxa de chegadas seriam dependentes dos estados onde a cadeia estava antes da mudança. Isto resulta na dependência dos intervalos entre chegadas. A figura 3.2 representa a idéia descrita acima para uma cadeia com n estados.

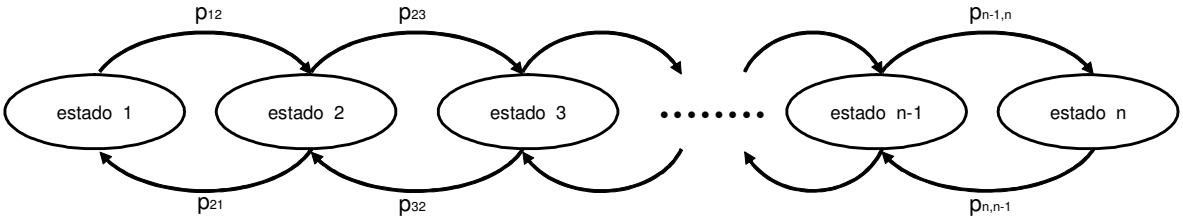


Figura 3.2 – Estados encadeados

No caso de tempo dividido em slots, o estado *i* representaria *i* slots vazios separando duas chegadas consecutivas e p_{ij} representaria a probabilidade da transição do estado *i* para o estado *j*. As chegadas podem ser unidades singulares, grupos de unidades ou uma quantidade contínua (uma certa quantidade de carga de trabalho chegando ao sistema) [FRO94].

A idéia, do modelo Markoviano, é utilizar os estados da cadeia para descrição do "stream" de tráfego enquanto um processo de Markov controla (modula) as probabilidades (uma associada a cada estado). Seja M = { $M(t)$} $_{t=0}^{\infty}$ um processo de Markov em tempo contínuo, com espaço de estados { 1, 2, 3, ..., m }. Assume-se que, enquanto M está no estado k , o processo de chegadas depende apenas deste estado e isso funciona para todos os estados 1 <= k <= m. Se ocorrer uma transição para outro estado, uma nova probabilidade irá reger o processo. A matriz P de transição $P = [p_{ij}]$ mantém a mesma função anterior, ou seja regular a transição entre os estados da cadeia [FRO94], conforme exemplo na figura 3.4:

$$P = \begin{bmatrix} P_{00} & P_{01} & \cdots & P_{0j} \\ P_{10} & P_{11} & \cdots & P_{1j} \\ \vdots & \vdots & \ddots & \vdots \\ P_{i0} & P_{i1} & \cdots & P_{ij} \end{bmatrix}$$

Figura 3.3 - Matriz de Transição Teórica

3.1.3 Modelo Auto-Similar

A auto-similaridade pode ser descrita através do conceito de fractal [TEL02]. Um fractal tem a característica de apresentar as mesmas formas apesar das diferentes escalas em que está sendo analisado. Estudos feitos na Internet concluem que as variações nas características do tráfego apresentam o mesmo comportamento apesar da escala de tempo que se está analisando. As variações no tráfego das redes são causadas pela mudança no comportamento dos intervalos entre os pedidos, variações dos tamanhos dos arquivos e do desempenho da rede no que diz respeito aos problemas relacionados aos congestionamentos e gargalos na largura de banda. A aparência é linear, apesar da variação na escala de tempo, e na verdade a variação pode ser infinita [LEL94], conforme vemos na figura 3.5 (a) até (e). Nesta figura os diferentes níveis de cinza são usados para identificar o mesmo segmento do tráfego em diferentes escalas de tempo.

Estatisticamente, auto-similaridade tem duas propriedades [GON05]:

- LRD (Long Range Depedence): Um processo tem dependência de longa escala se sua função de auto correlação *r(k)* não for uma somatória finita, isso quer dizer que $\sum_{k} r(k) = \infty$. Ou seja, o somatório dos valores da função de auto correlação *r*, para todos os valores *k* encontrados, tende ao infinito. Esta não "somabilidade" das correlações captura a intuição por trás da *LRD*; enquanto as correlações para grandes valores de *k* são pequenas, seu efeito cumulativo não pode ser desprezado e dá origem a características que são drasticamente diferentes das características dos modelos usados tradicionalmente.

- Variância diminuindo gradativamente: Implica que a variância média do modelo apresenta decréscimo mais lento se comparado com o tamanho da amostra.

Os primeiros artigos sobre tráfego auto-similar foram publicados no ano de 1994 [LEL94]. As coletas em LANs e WANs mostraram que as formas tradicionais de modelagem de tráfego de rede em geral e particularmente do tráfego da Internet não eram mais apropriadas. Foram assim abandonadas as formas Markovianas tradicionais para a adoção de novas modelagens baseadas em LRD.

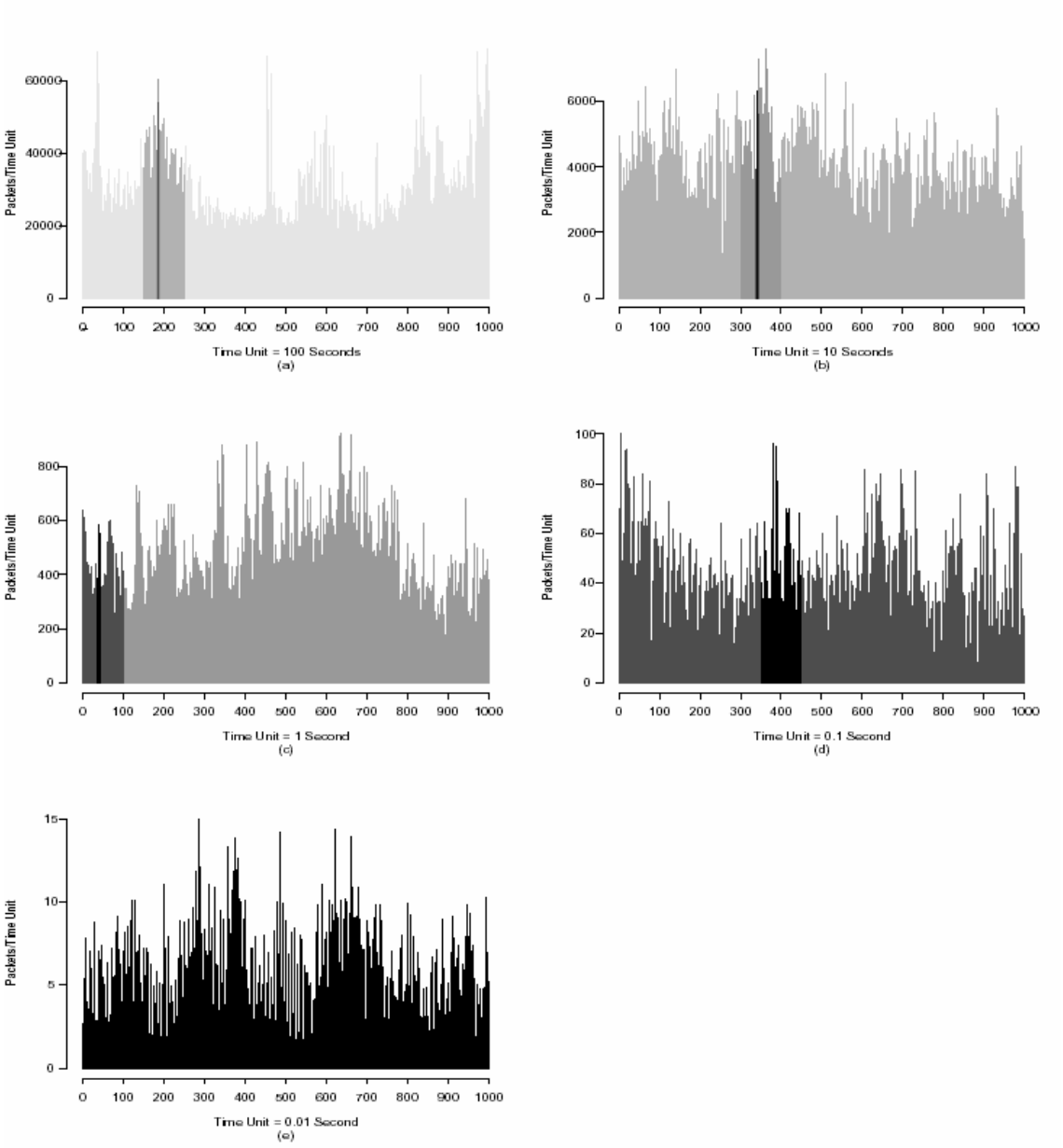


Figura 3.4– (a) – (e) Tráfego Ethernet (pacotes por unidade de tempo) em 5 diferentes escalas de tempo [LEL94]

Estudos recentes de medidas de tráfego de rede [GON05] têm revelado um novo fenômeno com ramificações em potencial para modelagem, projeto e controle das redes de banda larga. Isto inclui análise de alguns milhões de pacotes observados em uma LAN (Local Area Network) Ethernet em um ambiente de pesquisa e observação de alguns milhões de frames de dados provenientes de serviços de vídeo. Nestes estudos, o tráfego de pacotes parece ser estatisticamente auto-similar. O fenômeno auto-similar (ou fractal) faz com que o tráfego medido exiba uma estrutura similar sobre várias escalas de tempo. Em tráfego de pacotes, a auto-similaridade manifesta-se independentemente do tamanho das rajadas de dados em todas as escalas de tempo consideradas (de alguns milisegundos até minutos ou horas), a natureza de rajada do tráfego mostra-se similar.

Do ponto de vista matemático, o modelo de tráfego auto-similar difere dos outros modelos de algumas formas, considerando *s* uma unidade de tempo representativa de uma escala de tempo. Para modelos de tráfego tradicionais observa-se que à medida que *s* aumenta a "agregação" do tráfego tende para uma seqüência de variáveis aleatórias independentes e identicamente distribuídas similares ao ruído branco. Para modelos auto-similares, repetindo-se esta mesma operação, as seqüências resultantes não se distinguem entre si ("exatamente auto-similar") mas distingüem-se do ruído branco [1] ou convergem para séries de tempo com estruturas de auto-correlação não degenerativa ("assintoticamente auto-similar"). Os modelos tradicionais rapidamente convergem para o ruído branco após o aumento de normalmente duas ou três ordens de grandeza nas escalas de tempo [NES68].

Outra das principais características do modelo auto-similar, e provavelmente a de maior impacto no planejamento e dimensionamento, é o LRD (Long Range Dependence) [GON05] da distribuição de vários parâmetros do tráfego (tempo de recebimento dos pacotes, quantidade de dados transferidos por unidade de tempo, etc). Uma LRD significa que o tráfego tem alguma ordenação de memória, no entanto possui propriedades de forte correlação, e todas as outras características são significativas somente sobre uma quantidade limitada na escala de tempo. Qualquer relação menor que o tempo de transmissão de um pacote não tem significado físico.

As implicações em potencial do tráfego auto-similar em questões relacionadas com projeto, controle e desempenho de redes de alta velocidade vêm sendo estudadas exaustivamente nos últimos anos [GON05] [MEN98].

[1] O ruído branco, também associado ao ruído térmico, é provocado pela agitação dos elétrons nos condutores metálicos.

Os modelos mencionados até agora são analíticos e ao longo dos anos têm sido usualmente aplicados para a caracterização do tráfego de rede. Entre as aplicações de caracterização de tráfego destaca-se a geração de tráfego sintético para fins de simulação, a ser abordado a seguir.

3.1.4 Modelos de Geração de Tráfego Sintético

Os geradores de tráfego sintético são usados para injetar tráfego sintético na rede de acordo com um modelo que pode corresponder ao comportamento do usuário, ou das aplicações [PAX01]. Este paradigma é um contraste com os simuladores de rede que utilizam como base o tráfego no nível do pacote, e simulam o processo de chegadas de pacotes em um elemento particular da rede, de acordo com um modelo matemático. Os geradores no nível de pacote não podem ser usados como geradores de tráfego para aplicações que utilizem TCP, dependendo do que está sendo avaliado. Isto porque o TCP controla o congestionamento do início ao fim da conexão e ao fazer este controle causa uma mudança no tráfego original (por exemplo fazendo descarte de pacotes nos roteadores). Por esta razão, em [PAX01] a importância do uso de geradores de tráfego baseados em implementações ou simulações do TCP é amplamente discutida, assim como a questão de que os simuladores de rede devem gerar tráfego de rede sintético independente. Este tráfego deve corresponder a um modelo contemporâneo válido que possa ser utilizado tanto para o comportamento dos usuários como das aplicações.

Na maior parte, os geradores de tráfego sintético atualmente em uso são baseados nos estudos de [CRO98] e [MAH97] que capturaram o comportamento do usuário. Os modelos propostos por eles têm sido usados para geração de tráfego Web sintético para análise de desempenho de redes. Ambos foram implementados no [NS] (Network Simulator) software amplamente conhecido e empregado pelos pesquisadores da área. Os dois modelos utilizaram amostras com pequenas diferenças em seus estudos. Em [MAH97] os dados analisados refletem a coleta dos estudantes de graduação da Universidade de Berkeley do Departamento de Ciências da Computação e giram em torno de um milhão e setecentos mil pacotes TCP carregando o protocolo HTTP. Em [CRO98] a população analisada foram os estudantes de

graduação da Universidade de Boston do Departamento de Ciências da Computação e representam um milhão de objetos Web. Os dois conjuntos são hoje relativamente velhos pois datam de 1995 e 1998 respectivamente. É importante lembrar que os dados analisados foram coletados antes da versão HTTP 1.1 que implementou o conceito de conexão persistente e encadeamento. A conexão persistente possibilitou o uso de uma simples conexão TCP para transferir múltiplos objetos para um mesmo endereço IP (tipicamente os componentes embutidos em uma página Web). Já o encadeamento permitiu que o usuário fizesse uma série de pedidos, numa conexão persistente, sem precisar esperar pela resposta de cada pedido (o servidor porém precisa retornar os pedidos na ordem exata em que foram feitos). As conexões persistentes são importantes tanto para o servidor Web quanto para o browser no cliente mas o encadeamento é amplamente suportado apenas na implementação do servidor.

Os estudos utilizavam o conceito de página Web como base para a caracterização do tráfego, sendo influenciado pelo desenho da página, a localização dos componentes da página no servidor e o comportamento humano (suas ações) e a seleção da página que indiretamente controla a criação de novas conexões TCP e a transferência dinâmica de pedidos e respostas de dados sendo transmitidos nestas conexões. Nestes casos, modelar a chegada de conexões TCP e sua dinâmica interna de transferência implica em programas de geração de tráfego bastante complicados. As variáveis deste modelo são consideradas independentes e provenientes de populações com distribuições também independentes. As potenciais correlações entre as variáveis não são consideradas.

Recentemente um estudo feito por [CAO04] propõe um modelo que expressa o tráfego Web como uma coleção de conexões TCP independentes, caracterizada pelos parâmetros: tempo de chegada da conexão, o tempo gasto no cliente, o tempo gasto no servidor, o número de pedidos e respostas trocados, o tamanho de cada pedido individualmente, o tamanho de cada resposta individualmente e os atrasos do servidor. Os autores alegam que este modelo, baseado em conexões, é mais apropriado para a simulação do tráfego de rede, e que os modelos baseados em páginas são mais apropriados para simulação de carga de servidores. Assim, a geração de tráfego Web sintético é feita cuidadosamente pensando nos links de comunicação, roteadores e pilhas de protocolos quando adotado o modelo baseado em conexões TCP. Enquanto os modelos baseados em páginas afirmam serem melhores para gerar o tráfego sintético em termos de taxas de estabelecimento de conexões, tamanhos dos pedidos e respostas trocados nas conexões (incluindo as conexões persistentes) do protocolo HTTP. A razão mais importante para se preferir esta abordagem, segundo os autores, é escalonar melhor a grande variedade de aplicações. A premissa é que o protocolo HTTP apresenta facilidades para a modelagem por conexão TCP. Logo, se o processo entre chegadas de conexões e o tamanho dos dados

trocados, dentro da conexão, puderem ser modelados, então pode-se arbitrariamente misturar as conexões TCP sem precisar explicitar nenhum informação da aplicação. Dado a grande mistura de aplicações usadas na Internet atualmente, a abordagem de modelagem por conexão TCP, tenta obter escalabilidade ao excluir os detalhes da aplicação, como a estrutura da página Web.

Uma vez que o foco esteja direcionado para a geração de tráfego sintético de servidores Web, o modelo por conexões TCP acima proposto não é a melhor alternativa. Os modelos de geração de tráfego sintético, de especial interesse para este estudo, têm como base o modelo de fontes ON-OFF, conforme mostrado na figura 3.6.

Os modelos denominados modelos de fontes ON/OFF se mostram eficientes na reprodução da realidade pois capturam o comportamento do usuário e dos arquivos armazenados no servidor.

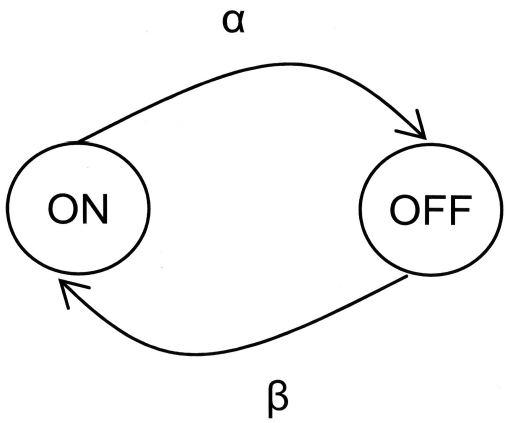


Figura 3.6 – Modelo ON-OFF

Estes modelos pregam que o fluxo de dados de uma fonte de tráfego é modelado como uma sucessão de períodos ativos (ON) e períodos de silêncio ou inatividade (OFF). A geração de dados, ou qualquer outra unidade de informação, ocorre apenas nos períodos de atividade (ON) e pode ser modelada para permitir a geração de tráfego.

Normalmente assume-se que os períodos de atividade (ON) e de silêncio (OFF) são independentes entre si e que seus tamanhos são variáveis aleatórias com distribuição Exponencial (para tempo contínuo) ou distribuição geométrica (para tempo discreto) [CRO98]. Existe uma probabilidade denominada α de ocorrer a transição do estado ON para o estado

OFF, assim como uma probabilidade β de ocorrer a transição do estado OFF para o estado ON.

3.1.5 Modelo SURGE - Scalable URL Reference Generator

É um modelo analítico baseado em diversas coletas reais de servidores Web, foi desenvolvido visando criar uma ferramenta de geração de carga, denominada SURGE, que permitisse reproduzir acessos reais de usuários aos servidores. Foi proposto por Paul Barford e Mark Crovella em 1998 [CRO98]. O referido artigo tornou-se um ponto de referência para estudos posteriores e apresenta inúmeros pontos em comum com o modelo de tráfego proposto nesta dissertação. Em virtude disto será feita, a seguir, uma explanação mais apurada deste modelo.

A ferramenta SURGE permite comparação para os seguintes parâmetros:

1) distribuição do tamanho de todos os arquivos que estão no servidor;

2) distribuição do tamanho do pedido (arquivos efetivamente transmitidos);

3) tamanho relativo do arquivo;

4) número de referências internas dos arquivos (referências embutidas);

5) referência de localização temporal;

6) período OFF (tempo inativo).

Na construção de uma carga Web analítica são encontrados três desafios. Primeiro descrever o conjunto de características dos fluxos Web para escolher o modelo e explicar porque este conjunto é importante. Segundo, descrever os resultados das novas medições da Web necessárias para popular os modelos do SURGE. Terceiro, descrever as questões que envolvem incorporar este modelo num único fluxo de saída, e como resolver estas questões.

A meta com o desenvolvimento do SURGE foi poder exercitar os servidores e a rede de modo a representar a Web. Nos servidores foram estudados: a fila de rede e o efeito da alta

variação do tráfego, o sistema de arquivos e sua associação com a "bufferização" do sistema, tanto no servidor quanto na rede. Estas motivações dirigiram as características modeladas no SURGE e as dividiram em duas categorias: equivalências do usuário e distribuições do modelo.

A idéia por trás da equivalência do usuário é que o SURGE possa gerar o tráfego de uma população com número conhecido de usuários. Então a intensidade de demanda do serviço, gerada pelo SURGE, pode ser medida em equivalência do usuário (UEs – User Equivalence).

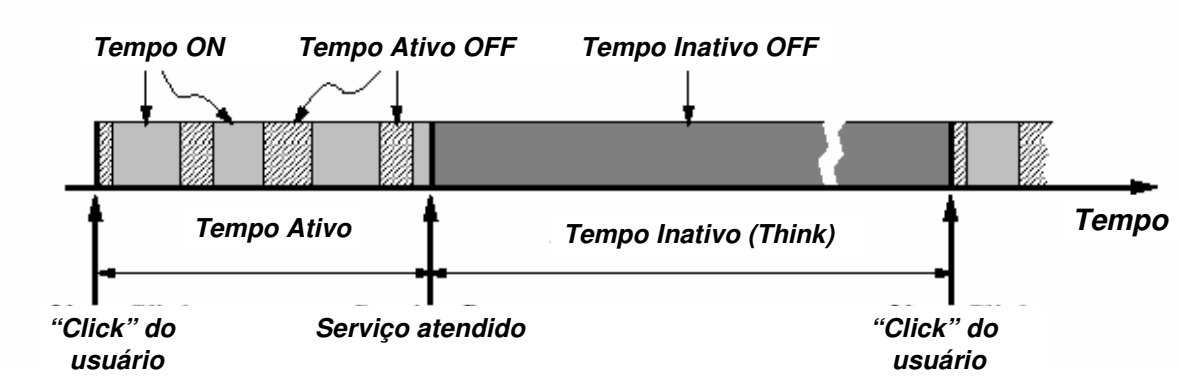


Figura 3.7 – Modelo ON-OFF com interação do usuário

Uma equivalência do usuário é definida como o processo entre o primeiro pedido por arquivos, requisição de uma página Web, até o período Inativo OFF. Tanto o período em operação quanto o período inativo mostram distribuições e propriedades de correlação que são características dos usuários reais da Web. Cada UE está entre o ON e o OFF de um processo. O período em que um arquivo está sendo transmitido é ON e quando o sistema está disponível é o OFF.

Os modelos de carga baseados em UEs têm efeitos importantes no desempenho, pois a UE tem períodos significantes em OFF (enquanto o usuário está inativo) e trabalha em processo de rajada (no início de um pedido). Ou seja, cada UE tem longos períodos de atividade seguidos por longos períodos de inatividade.

Os detalhes de como os componentes de um objeto Web são transferidos depende do "browser" e da versão do protocolo HTTP utilizados. Para o estudo original do SURGE a configuração usada foi o protocolo HTTP na versão 0.9, sem múltiplas conexões TCP. Este

estilo de transferência de objetos Web tem dois tipos de tempo OFF, como podemos ver na figura 3.7:

- OFF inativo – que corresponde ao tempo entre as sessões do usuário (entre a transmissão de objetos Web), ou seja, o tempo em que o usuário está pensando;
- OFF ativo - que corresponde ao tempo entre a transferência dos componentes de um único objeto Web, é o tempo de processamento gasto pelo "browser" para analisar gramaticalmente os arquivos Web e preparar para começar a nova conexão TCP.

Cada UE possui um conjunto de distribuição de probabilidades própria. Um aspecto importante de algumas destas distribuições é exibirem LRD.

Para que se cumpra o objetivo do SURGE, é preciso que sejam identificadas as distribuições de probabilidade dos itens a seguir:

1. Tamanho dos arquivos armazenados no servidor;
2. Tamanho do pedido transmitido pela rede;
3. Popularidade dos arquivos (distribuição dos pedidos por arquivo);
4. Referências embutidas no arquivo Web (ver 2.2 Web);
5. Localidade temporal (refere-se a probabilidade de que, uma vez que o arquivo foi solicitado, ele será solicitado novamente num futuro próximo);
6. Tempo OFF (como descrito na sessão anterior, para a modelagem correta do tempo OFF inativo é necessária a coleta das rajadas naturais do pedido de um usuário individual da Web. Para caracterizar propriamente o tempo OFF ativo é necessário replicar a transferência de objetos Web).

Há dois problemas básicos em construir um gerador de carga que atenda aos requisitos descritos. Primeiro, é necessário modelar cada uma das seis distribuições de probabilidade, e segundo identificar métodos que combinem estas distribuições em um único fluxo de saída.

Para a implementação deste modelo foram usados métodos estatísticos tradicionais: primeiro foram encontradas as função de distribuição de probabilidade dos 6 parâmetros definidos, segundo foram utilizados os testes de aderência ou a bondade do ajuste (Goodness-of-fit) Anderson-Darling (A^2) e Qui-quadrado (λ^2) para verificar quão bom era o ajuste à distribuição de probabilidade encontrada.

Para o tamanho de arquivos foi utilizada uma distribuição LRD. O modelo foi implementado de forma híbrida com uma nova distribuição para o corpo da distribuição, combinando com a distribuição de Pareto para a cauda. O teste λ^2 verifica o corpo dos dados contra alguns modelos de distribuições (Logo Normal, Weibull, Pareto, Exponencial e Log-Extreme) mostrando que o melhor valor λ^2 (o menor) é encontrado para a distribuição Logo Normal. Algumas técnicas de censura foram empregadas para determinar onde dividir a distribuição do corpo (Logo Normal) e a distribuição da cauda (LRD) . Foram selecionados todos os valores acima de um valor, definido como ponto de corte. Desta forma, o corpo da amostra pode assumir uma censura correta, desde que seja confirmado que está contaminado com a distribuição LRD da cauda. Foi utilizada a estatística A^2 para determinar o ponto de corte entre o corpo e a cauda.

O tempo OFF ativo, foi considerado como sendo tudo que fosse menor que um tempo estabelecido, o qual foi definido como sendo 1 segundo, baseado na inspeção dos dados. O teste λ^2 mostra para a distribuição de Weibull o melhor ajuste, entre as consideradas.

O número de referências embutidas em cada arquivo foi extraído da amostra, e definido como sendo o número de arquivos transferidos na seqüência idêntica dos arquivos buscados por um dado usuário para os tempos OFF entre transferência. Neste estudo foi apurado que este valor foi sempre menor que 1 segundo.

Os arquivos grandes causam um grande impacto na rede e no desempenho do servidor, uma vez que consomem maior quantidade de recursos e por mais tempo. Para tratá-los foi desenvolvido um método que permite combinar a cauda e o corpo da distribuição. Neste método os valores do conjunto Y são gerados da F(x) Função Densidade de Probabilidade da variável x, começando do menor para o maior valor em x. Este método resulta em ajustes muito próximos da cauda da distribuição, uma vez que diminuem a variância dos dados, e não inserem grande erro no corpo.

Como referência para gerar seqüências com localidade temporal, o SURGE começou selecionando cada nome de arquivo individualmente de uma pilha (a ordem inicial não foi importante). Esta seqüência de valores foi ajustada à distribuição Logo Normal, que

apresentou o melhor resultado. A seqüência foi invertida para obter a seqüência dos nomes. Isto foi repetido para os arquivos selecionados na pilha com a distância igual ao próximo valor da seqüência e a pilha foi reordenada após cada seleção.

O resumo do modelo de distribuição e os parâmetros usados no SURGE é dado na tabela 3.2 abaixo:

COMPONENTE	MODELO	FUNÇÃO DENSIDADE DE PROBABILIDADE	PARÂMETROS
Tamanho dos arquivos – corpo	Logo Normal	$p(x) = \frac{1}{z\sigma v 2\pi} e^{-(tnz-\pi)^2/2\sigma^2}$	μ = 9.357; σ = 1.318
Tamanho dos arquivos – cauda	Pareto	$p(x) = \alpha\kappa^{\alpha} x^{-(\alpha+1)}$	κ = 133K; α = 1.1
Popularidade	Zipf		
Localidade Temporal	Logo Normal	$p(x) = \frac{1}{z\sigma v 2\pi} e^{-(tnz-\pi)^2/2\sigma^2}$	μ = 1.5; σ = 0.80
Tamanho do pedidos	Pareto	$p(x) = \alpha\kappa^{\alpha} x^{-(\alpha+1)}$	κ = 133K; α = 1.1
Tempo Ativo OFF	Weibull	$p(x) = \frac{\sigma z^{6-1}}{\alpha^6} e^{-(z/a)^6}$	α = 1.46; *b* = 0.382
Tempo Inativo Off	Pareto	$p(x) = \alpha\kappa^{\alpha} x^{-(\alpha+1)}$	κ = 1; α = 1.5
Referências Embutidas	Pareto	$p(x) = \alpha\kappa^{\alpha} x^{-(\alpha+1)}$	κ = 1; α = 2.43

Tabela 3.2 – Estatísticas do modelo SURGE

Infelizmente este método simples resultou numa distribuição não uniforme dos nomes dos arquivos através da seqüência pedida, não refletindo assim a realidade. Desta forma o método foi modificado definindo uma pequena janela em cada lado do local especificado na seqüência Logo Normal. Assim, cada arquivo nesta janela foi assinalado com um peso cujo valor foi proporcional ao número de pedidos feito deste documento. A escolha de qual arquivo seria movido para o topo da pilha era baseado nos pesos de cada arquivo dentro da janela.

Este método de geração de seqüências resultou numa distribuição muito similar aos nomes ao longo da seqüência, e em valores distantes na pilha ainda seguindo a distribuição Logo Normal desejada.

O uso de múltiplos clientes apresentou o problema de localidade temporal. Quando um processo do UEs estava rodando num mesmo “host” eles dividiam um nome de arquivo comum na seqüência. Entretanto, quando processos UEs estavam rodando em múltiplos “hosts”, eles não conseguiam dividir uma lista comum sem uma sincronização com significativa

sobrecarga. Para manipular este caso, que representava um erro de implementação do modelo, foram geradas seqüências de nomes de arquivos com propriedades de distâncias independentes na pilha relacionada. Isto foi feito escalonando os valores de saída da distribuição Logo Normal com o número de clientes que foram usados na simulação. No caso em que os pedidos vinham de "hosts" separados, sofriam intercalação no servidor de uma forma regular, isto resultou na propriedade de localidade temporal.

O SURGE foi implementado em dois conjuntos de programas em C. Para comparação de resultados foram realizados testes usando o [SPECWeb96]. O SPECWeb96 é o conjunto de valores obtidos através de testes de estresse com servidores Web, e que são usados para comparação de desempenho.

A validação consistiu em verificar se a saída do SURGE estava de acordo com os seis modelos de distribuição (tamanho de arquivos, tamanho dos pedidos, popularidade, referência embutidas, localidade temporal e tempos OFF). Em todos os testes realizados, apenas a localidade temporal foi afetada quando o SURGE foi escalonado, ou seja, o número de UEs simultâneas foi incrementado.

O principal objetivo da comparação do SURGE com o SPECWeb96 foi o de descobrir qual dos dois geradores de tráfego originaria tráfego auto-similar. A variância do tempo no SURGE e do SPECWeb96 mostrou evidências de auto-similaridade quando a intensidade do tráfego é baixa mas se o tráfego aumenta, a auto-similaridade desaparece. Este efeito pode ser entendido pela análise de cada um dos casos como uma coleta individual de fontes do SURGE. Considerando cada instância do SURGE executando como uma fonte ON/OFF, a distribuição marginal é uma variável aleatória de Bernoulli. Esta variância é maximizada quando dois estados são equiprováveis, quando um estado inicia dominando o outro, a demanda da "thread" decresce.

O tráfego gerado pelo SURGE vem de uma variabilidade constante na fonte, o número de fontes é que aumenta quando a intensidade da carga aumenta. O SURGE produziu tráfego auto-similar sob condições de alta ou baixa intensidade da carga. Com alta intensidade de carga, o SURGE gera tráfego de rede auto-similar, o que não parece ser verdadeiro para o SPECWeb96. Estes resultados sugerem que a exatidão da carga Web gerada é importante, desde que comparada com as cargas reais ou com cargas de um gerador tradicional, como o SPECWeb96, e poderão otimizar a avaliação de desempenho do sistema.

3.1.6 Outros modelos

Além dos modelos de tráfego já descritos, outras abordagens de modelagem de tráfego da Internet merecem ser listadas pela sua ocorrência na literatura estudada: o Processo de Poisson Modulado por uma Cadeia de Markov (MMPP) [MUS96] [MUS05], o FBM (Fractional Brownian Motion) [NOR95], modelos Multifractais [TEL02], Decomposição Wavelet [VEN01], modelos de Mapa Caótico [ERA95] e FARIMA (Fractional Autoregressive Integrated Moving Average) [XUE99].

O (MMPP) Processo de Poisson Modulado por uma Cadeia de Markov é o modelo mais comum de um processo modulado por uma cadeia de Markov. Neste caso o mecanismo de modulação simplesmente estipula em que estado *k* de *m* possíveis estados *{1, 2, 3, ..., m}* as chegadas ocorrem de acordo com um processo de Poisson de taxa λ_k. A medida que o estado em que a cadeia se encontra é alterado, a taxa do processo de Poisson se altera também. Para mostrar o uso do MMPP consideremos uma fonte única de tráfego, por exemplo, cujo processo de geração de chamadas pode ser representado pela quantização em um número finito de taxas exponencialmente distribuídas. Assim cada taxa seria representada por um estado na cadeia de Markov. A matriz Q de transição entre os estados Q = [Q_{kj}] seria determinada de modo empírico, calculando a fração do tempo em que a cadeia comuta do estado *k* para o estado *j* [MUS96] [MUS05]. O MMPP simula a hierarquia do tráfego de rede e seus componentes são ajustados pelos objetos reais, como a distribuição dos fluxos TCP ou do tamanho das páginas Web, ou a distribuição da chegada destas páginas e fluxos. A principal dificuldade é manipular a complexa estrutura matemática dos processos estocásticos em que o modelo é baseado.

O FBM (Fractional Brownian Motion) é um modelo LRD que propõe um método Gaussiano para estudar o comportamento das filas. Entretanto apresenta uma estrutura de correlação restrita e falha na captura da correlação em termos curtos do tráfego real [NOR95].

Muitas pesquisas foram realizadas para o modelo de Multifractais [TEL02], cujo atrativo é sua propriedade de capturar ricamente o comportamento da variância nas escalas de tempo. As análises sugerem que estes modelos são os melhores para ajustar os dados. O Cascade é uma sub classe do Multifractal e também uma extensão dos modelos auto-similares que capturam o comportamento do tráfego em todas as escalas do tempo significativas. Estes modelos têm uma boa aproximação às propriedades LRD do tráfego da Internet porém são difíceis de gerenciar devido à sua complexidade analítica.

A Decomposição Wavelet tem sido usada como uma abordagem natural para estudar a invariância nas escalas mas apenas recentemente foi introduzido no campo de rede de comunicação. Estes modelos são computacionalmente muito eficientes mas são complexo e difíceis de ajustar, dificultando o mapeamento entre os parâmetros do tráfego e os coeficientes do modelo [VEN01].

Modelos de Mapa Caótico foram propostos como uma evolução determinista para os sistemas governados por um conjunto de regras de comportamento. Os modelos derivados são simples mas normalmente difíceis de entender a relação entre o modelo e os parâmetros do tráfego real [ERA95].

Os modelos FARIMA (Fractional Autoregressive Integrated Moving Average) são usados para modelagem de tráfego de vídeo e podem gerar seqüências LRD. Os coeficientes destes modelos são encontrados através de um filtro do ruído Gaussiano e capturam tanto dos períodos curtos quanto dos períodos longos o comportamento do tráfego. Entretanto, são modelos complexos e sua estrutura faz com que seja difícil a compreensão da relação entre os coeficientes filtrados e os dados do tráfego real [XUE99].

Todas estas abordagens chegam a conclusões similares usando técnicas diferentes. O objetivo é sempre ter a máxima acuidade quanto ao comportamento do tráfego real de modo a usar estas ferramentas de forma mais eficientes para o planejamento de capacidade da rede e relacionar corretamente a causa e o efeito dos fenômenos do tráfego de rede.

3.1.7 Conclusão

A modelagem do tráfego antes do advento da Internet fazia uso da distribuição de Poisson e de Cadeias de Markov [PAX95] [ADA97] para caracterizar o comportamento do tráfego gerado para a rede. Isto permitia que o planejamento da capacidade fosse não só possível mas também eficiente. Estudos recentes indicam que o atual tráfego da Internet é melhor caracterizado por distribuição de cauda pesada, também chamados de LRD (Long Range Dependence) [GON05] e apresenta auto-similaridade em diferentes escalas de tempo [LEL94] [CRO95] [WIL00].

Os modelos auto-similares reportam que o tamanho dos arquivos transmitidos pelos servidores Web é modelado por distribuições LRD, por exemplo, a distribuição de Pareto [CRO95]. Existe grande complexidade matemática no tratamento deste tipo de distribuição devido a sua enorme variabilidade. Esta variabilidade pode levar, em muitos casos, a modelos de estimações sub ou super dimensionadas.

Os modelos de tráfego que foram mostrados neste capítulo vêm sendo utilizados para a previsão do comportamento das redes de comunicação de dados digitais (como a atual Internet). Estes modelos podem ser utilizados em estudos analíticos ou nas simulações computacionais.

Nos próximos capítulos apresenta-se um estudo estatístico baseado na classificação de conteúdo do tráfego Web, levando em consideração o tipo de arquivo transmitido. A dinâmica usada no SURGE, de partir de uma amostra, descobrir as distribuições dos elementos de interesse e seus parâmetros, e utilizar estes dados para uma ferramenta de simulação são os passos seguidos na proposta de modelo desta dissertação. O modelo proposto será descrito em detalhes no capítulo 4 desta dissertação.

4 Modelo por Classificação de Conteúdo

Coletar e analisar dados é uma regra importante na avaliação dos protocolos, programas e tráfego da Web. Os pesquisadores interessados na Web freqüentemente utilizam “logs” e “traces” para caracterizar o tráfego Web e avaliar novas idéias para melhorar o seu desempenho. Conduzir estudos na Web requer conhecimento do protocolo HTTP além dos protocolos de rede, assim como, a habilidade em desenvolver softwares robustos e eficientes, que possibilitem tratar os imensos arquivos de dados. O HTTP, assim como outros protocolos, não foram desenvolvidos pensando em medições.

Tradicionalmente os estudos da Web têm sido feitos a partir de uma das três formas abaixo:

1. Monitorando os pacotes do tráfego HTTP. São obtidos através da coleta do tráfego dos servidores e proxies. Entretanto, estes não possuem um nível mais baixo de informações quando o interesse está nos vários estágios de uma transação na Web. Isto pode ser resolvido através da coleta de pacotes feita diretamente no circuito de comunicação. Existem programas que capturam os pacotes IP na rede, reconstroem o fluxo de bytes em cada conexão TCP e reconstroem o pedido e resposta das mensagens HTTP;

2. Analisando os “logs” dos servidores Web. Os “logs” dos servidores têm sido usados numa grande variedade de estudos e pesquisas. Para analisar estes “logs” surgem desafios práticos porque eles possuem milhares, se não milhões de registros e apresentam formatos variados. É necessário desenvolver programas que filtrem, transformem, formatem e analisem os arquivos coletados nos servidores;

3. “Logs” e “traces” públicos disponibilizados. A importância de caracterizar o tráfego Web para a comunidade de pesquisa fez surgir repositórios públicos de “logs” e “traces”. Dentre eles pode-se listar: o ITA (Internet Traffic Archive), o WCA (World Wide Web Consortium’s Web Characterization Group Repository) e o NLANR (National Laboratory for Applied Network Research).

Este capítulo descreve a abordagem do método utilizado na Modelagem Web por Classificação de Conteúdo [KOT05], e os elementos envolvidos. A proposta de um método de geração de tráfego sintético é apresentada e aplicada. Os resultados alcançados são mostrados, bem como as evidências de que o modelo traz contribuições ao cenário atual de modelagem de tráfego sintético para servidores Web.

4.1 Descrição do método utilizado

Com a crescente importância da Internet na vida profissional e pessoal da população mundial, cresce também a necessidade de modelos e ferramentas que reproduzam corretamente o tráfego típico da principal aplicação de Internet, a Web. É importante, em particular, a habilidade de gerar pedidos HTTP que imitem os usuários reais, para avaliação de desempenho e planejamento de capacidade dos servidores, "proxies" e redes [KR01].

O modelo aqui proposto foi baseado no modelo SURGE, desenvolvido por Crovella e Barford em [CRO98], que é um dos modelos mais citados para geração de tráfego sintético Web. Ambos são modelos de caracterização de carga, cujo objetivo principal é representar corretamente a carga em estudo e através da observação do tráfego real, caracterizar os elementos importantes para construir um método, implementado em software, para geração sintética do mesmo tráfego. Os modelos definidos a partir de fontes reais de tráfego são denominados de modelos naturais [MEN98].

O modelo SURGE é baseado no modelo de fontes ON/OFF que captura o comportamento do usuário e dos arquivos armazenados no servidor. Para entender o comportamento do período ON é necessário entender alguns detalhes da organização Web. Em particular, que os arquivos Web podem incluir outros arquivos, e estes arquivos são necessários para o resultado final a ser mostrado ao cliente. Normalmente estes arquivos são imagens ou gráficos. Assim, o pedido de um único arquivo Web, pelo usuário, resulta em múltiplos arquivos sendo transmitidos do servidor Web. O conjunto de arquivos solicitados, mais todos os arquivos transferidos é chamado de objeto Web. Assim quando o sistema está no estado ON, a sessão está ativa e enviando os objetos requisitados na sessão.

Os detalhes de como os componentes de um objeto Web são transferidos depende do "browser" e da versão do protocolo HTTP utilizados. Na versão 1.0 do protocolo HTTP, para

cada objeto de uma página, nova conexão é estabelecida. Assim é grande o número de conexões para a transmissão de uma única página, logo não há como discernir entre quais arquivos foram transmitidos por um mesmo usuário, ou sessão. Na versão 1.1 do HTTP a conexão persiste até que todos os objetos de uma página sejam transmitidos, desta forma é simplificada a tarefa de identificar todos os objetos transmitidos durante uma sessão, ou período ON. Em virtude disto, para o método aqui proposto, caso o tráfego coletado seja da versão 1.0, um valor para a sessão será inferido baseado na observação dos valores observados para o tráfego da versão 1.1.

No corpo da mensagem das requisições HTTP, entre diversos outros campos está o campo Content-Type, que será usado nesta dissertação para a classificação dos tipos de arquivos. Após a classificação dos tipos de arquivos, as classes serão utilizadas para determinar a distribuição de probabilidades do tamanho dos arquivos de cada uma. Os tipos de arquivos do campo content-id, considerados para este estudo, ficaram restritos ao grupo denominado suportado.

Apenas a variável "tipos de arquivos transmitidos" será remodelada, as demais variáveis, como popularidade, localidade temporal, e tempo active-off, continuam modeladas da mesma maneira proposta originalmente pelo modelo SURGE.

Ao separar o tráfego de um servidor Web pelos tipos de arquivos pretende-se verificar o comportamento das distribuições de probabilidade dos mesmos. Como hipótese principal desta dissertação espera-se que não apresentem distribuições LRD e possam ser modeladas por distribuições de decaimento Exponencial ou Sub Exponencial. Caso seja comprovada esta hipótese, isto acarretaria uma simplificação significativa na modelagem do tráfego Web.

Outra contribuição do modelo proposto é a geração de uma matriz de probabilidade de transições entre estados, que possibilite representar o comportamento dinâmico das requisições. A aplicação vislumbrada é poder deduzir, a partir desta matriz, os parâmetros de desempenho do sistema. A utilização deste modelo implica em que o estado futuro depende apenas do estado corrente e não dos estados anteriores, nem do tempo já gasto no estado atual. Isto restringe a variável aleatória que descreve o tempo gasto no estado a uma distribuição geométrica no caso discreto e a uma distribuição Exponencial ou Sub Exponencial no caso contínuo [TAY98].

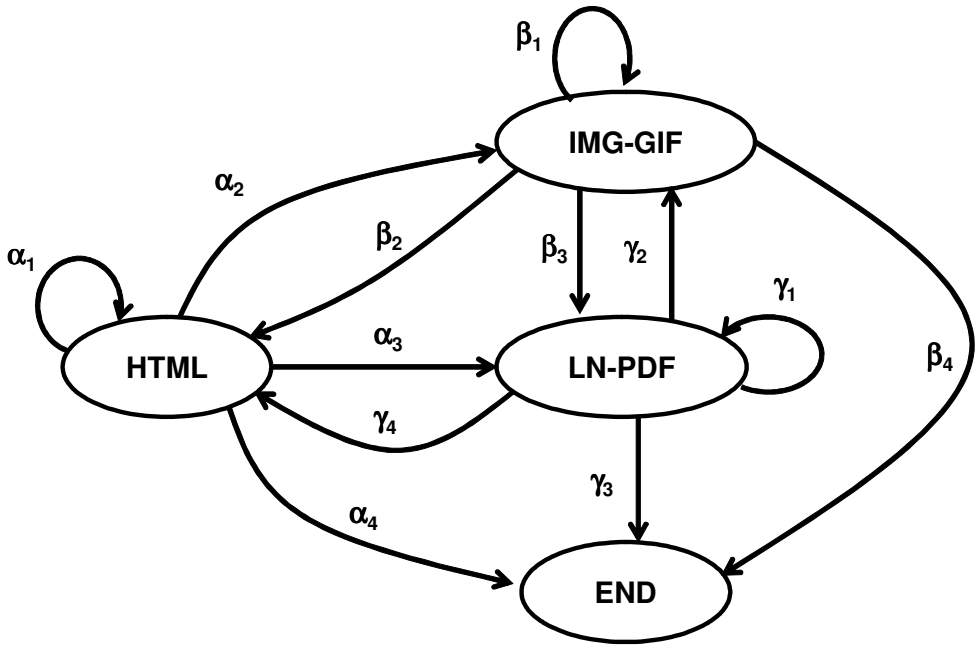


Figura 4.1 – Exemplo da transição entre tipos de arquivos

Numa página Web, após a primeira requisição ser produzida pelo usuário, muitos outros elementos são invocados. Basicamente, após a transferência de um objeto inicial, são produzidas várias requisições de acesso a outros objetos. Estas várias requisições são produzidas automaticamente pelo próprio programa cliente (browser) para transferir todos os objetos necessários para apresentar o conteúdo solicitado pelo usuário. A Figura 4.1 ilustra um diagrama de estados hipotético utilizando uma classificação de conteúdo com três tipos:

- arquivos em formato de hipertexto HTML;
- imagens em formato imagem IMG-GIF;
- arquivos em formato LN-PDF.

A figura 4.1 deve ser interpretada da seguinte maneira: Um acesso inicial a um objeto do tipo HTML será seguido por uma transmissão de um novo objeto de outro tipo ou em alguns casos do mesmo tipo. No caso específico deste exemplo, $\alpha 1$, $\alpha 2$ e $\alpha 3$, representam as probabilidades do próximo objeto transmitido a partir de um arquivo HTML ser do tipo HTML, IMG-GIF, LN-PDF, e $\alpha 4$ representa a probabilidade de não haverem mais requisições de transmissão de objetos para esta sessão, iniciando-se então um período OFF. A relação $\sum_{j=1}^{i} \alpha j = 1$ deve ser sempre verdadeira.

O mesmo ocorre para as probabilidades de transições a partir de cada um dos outros tipos de arquivos. Desta forma, β_1, β_2, β_3, β_4 são as probabilidades para as transições partindo de IMG-GIF para IMG-GIF, HTML, LN-PDF e OFF respectivamente. Assim como γ_1, γ_2, γ_3 e γ_4 representam as probabilidades de transição entre LN-PDF para LN-PDF, IMG-GIF, OFF e HTML respectivamente. No exemplo, $\sum_{j=1}^{i} \beta_j = 1$ e $\sum_{j=1}^{i} \gamma_j = 1$ são verdadeiros para os tipos de arquivos IMG-GIF e LN-PDF.

De modo genérico, pode-se representar o diagrama de estados da Figura 4.1 através de uma matriz quadrada P com as probabilidades de transição de estados, dada por:

$$P = \begin{bmatrix} P_{00} & P_{01} & \cdots & P_{0j} \\ P_{10} & P_{11} & \cdots & P_{1j} \\ \vdots & \vdots & \ddots & \vdots \\ P_{i0} & P_{i1} & \cdots & P_{ij} \end{bmatrix}$$

Figura 4.2 – Exemplo genérico de Matriz de Transição

onde j representa o tipo de arquivo, incluindo-se o estado representando o fim da sessão. Na montagem da matriz, o estado i é o estado que representa a transição para o fim da sessão. Para um estado k, $\sum_{j=1}^{i} P_{kj} = 1$ com $k \neq i$.

Esta matriz de transições de estados representa a probabilidade de sucessão de tipos de arquivos transmitidos, e não o seu tamanho transmitido. O tamanho dos arquivos transmitidos em cada estado deve ser determinado através do registro do tamanho dos mesmos nas coletas analisadas e será feito posteriormente.

A análise dos arquivos de "log" irá revelar a quantidade de tipos de arquivos. É possível obter muitos tipos, e utiliza-se como critério para a escolha dos tipos a serem analisados, o percentual em bytes transmitidos que este tipo representa na amostra. Ou seja, com a análise de um menor número de tipos, pretende-se representar o maior volume possível do tráfego da amostra.

Uma matriz de transições de probabilidade entre os tipos de arquivos para uma dada sessão também pode ser extraída dos arquivos de "log", de modo a completar o modelo.

Espera-se que o tempo inativo, estado OFF, seja exponencialmente distribuído e o processo de chegada de sessões seja um processo de Poisson [TAY98].

Na próxima sessão um método é proposto para a obtenção dos dados a partir das amostras de tráfego real.

4.2 Especificação do método para obtenção dos parâmetros do modelo

Usando uma heurística semelhante àquela desenvolvida em [CRO98], fez-se a especificação de um algoritmo para formatar amostras de servidores Web, com o objetivo de obter as matrizes de probabilidade de transição entre os tipos de arquivos. Este algoritmo pode, por exemplo, ser implementado em linguagem C.

Utilizando um arquivo de entrada com 10 campos em cada linha, separados por espaço, e cada linha representando uma requisição Web. Os campos serão descritos a seguir na ordem em que aparecem:

1. indica o tempo em que o pedido da conexão foi atendido. Totaliza os segundos transcorridos desde 01 de janeiro de 1970, com arredondamento em milisegundos. Usado para controle do período de tempo denominado Δt, que representa o intervalo de tempo em que a conexão será considerada a mesma, ou seja, a duração da conexão. Considera-se que a versão do protocolo HTTP é 1.0.

2. tempo de duração da conexão ou tempo de serviço, em milisegundos. Contabiliza o tempo entre o aceite da conexão até o último pacote transferido para o arquivo solicitado.

3. endereço IP do cliente. Para cada endereço IP, durante um Δt, deve-se mapear as transições. Após o Δt, o mesmo IP pode ser considerado como uma nova sessão, para um próximo período Δt.

4. mostra o código de resposta HTTP. Os códigos são descritos nos Anexos.

5. registra o tamanho em bytes da resposta enviada ao usuário. Usado para o ajuste da distribuição densidade de probabilidade.

6. descreve o método de requisição. Os métodos serão descritos nos Anexos.

7. é a URL requisitada pelo usuário.

8. identificação do usuário (se não identificado é substituído por um traço "-").

9. é a hierarquia e o endereço do servidor, descreve de onde e como a requisição e os objetos foram encontrados.

10. mostra o tipo do arquivo servido. Esta será a informação que irá alimentar a matriz de transição.

Neste exemplo serão considerados sete tipos de arquivos, encontrados com grande freqüência no tráfego Web, com as seguintes extensões: html, jpeg, gif, mpeg, pdf, xml e outros, nesta ordem. A partir destes, monta-se uma matriz com dimensão 8 x 8. O oitavo e último elemento é o estado FIM. Indica que a sessão do usuário acabou, seja pelo tempo definido de sessão ter transcorrido, ou por inatividade até o tempo de sessão ser esgotado. A escolha dos tipos de arquivos foi feita após a observação dos tipos mais freqüentemente utilizados na composição do tráfego Web.

	HTML	JPEG	GIF	…	FIM
HTML	[1,1]	[1,2]	[1,3]	...	[1,8]
JPEG	[2,1]	[2,2]	[2,3]	...	[2,8]
GIF	[3,1]	[3,2]	[3,3]	...	[3,8]
...	...	...	...	...	...
FIM	[8,1]	[8,2]	[8,3]	...	[8,8]

Figura 4.3 – Exemplo de Matriz de Transição

Segue o exemplo de preenchimento da matriz de transição, de acordo com as linhas da amostra abaixo, compostas por 4 requisições Web caracterizadas pelos 10 campos anteriormente citados:

[1]1101600018.615 [2]270 [3]165.102.49.118 [4]TCP_MISS/200 [5]645 [6]GET [7]http://www.jetaudio.com/ad/contentr.asp [8]- [9]DIRECT/216.133.247.218 [10]text/html

[1]1101600018.675 [2]171 [3]165.102.49.118 [4]TCP_HIT/200 [5]5179 [6]GET [7]http://free.hypnotrick.com/images/angelina/m01/title_youngname.gif [8]- [9]NONE/- [10]image/gif

[1]1101600018.754 [2]306 [3]9.93.52.80 [4]TCP_MISS/200 [5]4665 [6]GET [7]http://www.gimmie.tudelft.nl/homes/blok11b/TN_Groepopboot.gif [8]- [9]DIRECT/130.161.7.138 [10]image/gif

[1]1101600322.794 [2]99 [3]165.102.49.118 [4]TCP_HIT/200 [5]7237 [6]GET [7]http://free.hypnotrick.com/images/angelina/m01/clip1.jpg [8]- [9]NONE/- [10]image/jpeg

No primeiro registro, deve-se somar 1 na posição [1,1] HTML, pois um arquivo HTML foi transmitido. Para cada linha já tratada, um caractere de controle deve ser inserido na primeira posição da linha. No segundo registro, ainda é o mesmo cliente e a mesma sessão (mesmo endereço IP e o Δ_t não ultrapassou o limite estabelecido). Tem-se um arquivo GIF, iremos somar um na posição [1,3]. No terceiro recebe-se outro GIF, porém este registro não faz parte da sessão atual (IP diferente). Será ignorado por hora, ou seja, não será inserido o caractere de controle no início deste registro, e na próxima leitura do arquivo esta linha será contemplada. No quarto registro o arquivo transmitido é um JPEG, apesar do cliente (IP) ser o mesmo, o Δt foi ultrapassado, caracterizando que a sessão que estava em análise foi encerrada. Deve-se somar 1 na coluna do último elemento lido [3,8], indicando o início de um período OFF. Nova re-leitura do arquivo é feita até que seja encontrada a primeira linha sem o caractere de controle, e o mesmo processo se repete para o próximo IP e Δ_t, até o fim da sessão dentro do tempo estipulado, ou final do arquivo.

O programa deverá receber como parâmetro o arquivo de entrada com os dados da amostra coletada e o valor de Δt (em segundos). Baseado na idéia do modelo ON-OFF, foi usado o tempo ON (Δt), ou seja, uma sessão do cliente (cálculos baseados no campo 1 da

amostra) com 120.000 milisegundos (2 minutos). Se as amostras fossem da versão HTTP 1.1, através do protocolo tem-se exatamente cada sessão, mas para amostras obtidas com a versão HTTP 1.0, precisa-se inferir este valor.

O total de conexões tratadas deve ser armazenado na matriz, pois há alguns casos em que o registro não possui o campo 10 e portanto deverá ser descartado, ou, trata-se de algum outro tipo que não está sendo considerado e será somado à [linha,coluna] que representa o tipo outros.

Uma vez especificado o método para obtenção dos parâmetros do modelo, faz-se necessária a escolha de amostras que possuam os dados necessários. Em seguida o método foi implementado e aplicado às amostras selecionadas para a geração dos resultados.

4.3 Validação do método proposto através da análise de amostras de tráfego real

Utilizando o método já descrito, foi realizada a análise de duas fontes de dados reais. Ao realizar o experimento com mais de uma fonte, o objetivo foi de uma dupla validação do tráfego sintético Web gerado através do modelo proposto. Estas fontes de dados foram alvo de estudos em trabalhos publicados na literatura [ARL00] [DUT04] e portanto permitem comparar os resultados alcançados.

Primeiramente, as análises foram realizadas com arquivos de servidores de cache Web e depois, com o intuito de evitar, ao máximo, possíveis vícios e para confirmar os resultados já alcançados, as análises foram repetidas para amostras de servidores Web. As características dos dados são descritas a seguir.

4.3.1 Amostras de servidores cache Web - IRCache

Os caches Web surgiram baseados nos sistemas de cache de memória que os microcomputadores adotaram para aumentar a velocidade de acesso à memória principal e assim melhorar o desempenho do processamento.

Segundo [DUT04], a utilização dos servidores de cache Web foi intensificada nos últimos 10 anos, como alternativa para diminuir os problemas de desempenho da Web. Armazenar as páginas localmente aumenta a disponibilidade do serviço, reduz o tempo de atendimento e o tráfego da rede. Sendo o modelo Web do tipo cliente-servidor, armazenar os arquivos ou objetos mais acessados num servidor que atende a inúmeros clientes, significa reduzir x.n vezes a transmissão deste objeto, onde x é o número de clientes que solicitou o mesmo objeto n. Um algoritmo eficiente de cache Web melhora significativamente a qualidade do serviço oferecido [KR01].

Os primeiros resultados a serem apresentados neste trabalho foram obtidos através da análise de coletas feitas pelo IRCache. O IRCache é um projeto de cache Web do NLANR - National Laboratory for Applied Network Research [NLANR]. Os servidores cache do NLANR registram os acessos de toda a Internet e estão geograficamente distribuídos pelo Estados Unidos da América para balanceamento de carga e atendem a todos os continentes. Assim sendo, podemos afirmar que às milhares de requisições diárias destes servidores refletem o comportamento da Internet em geral, pois não se restringem a um grupo restrito de clientes ou a aplicações específicas.

A figura 4.4 mostra as ligações de rede entre os sistemas de cache da NLANR nos Estados Unidos, de onde se interligam com caches da África do Sul, Rússia, Brasil, Japão, Austrália, Holanda, Reino Unido, entre outros. Todas estas conexões são listadas em [NLANR].

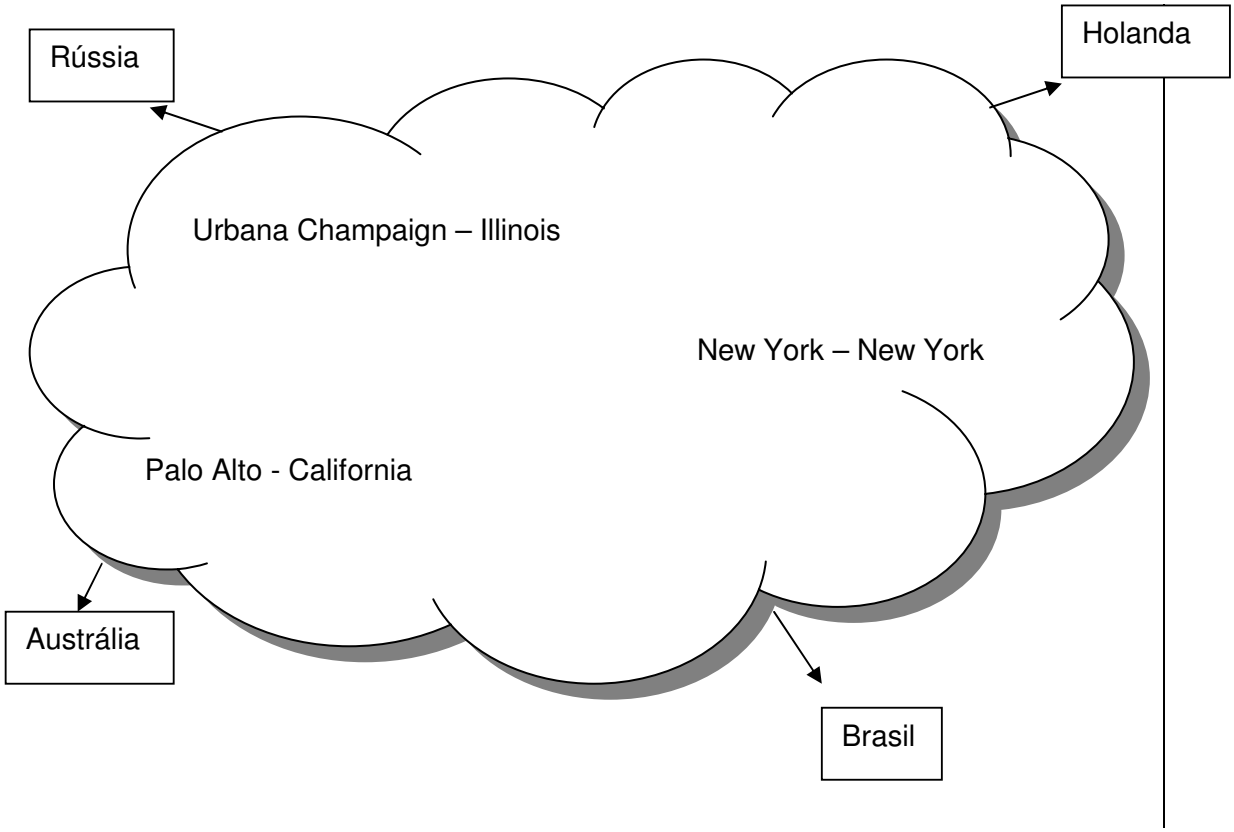


Figura 4.4 – Conexões da NLANR

A escolha das amostras foi aleatória, sem qualquer alusão ao dia da semana, mês, período do ano ou localidade. As coletas postadas na página do IRCache são constantemente substituídas pelas mais recentes. As quatro amostras, que compõem objeto de estudo neste trabalho, estavam entre as últimas postadas na ocasião.

Foram analisados quatro arquivos coletados no mês de novembro de 2004 e que totalizam mais de um milhão e quinhentas mil linhas, onde cada linha é o registro de um arquivo solicitado. Seguem mais informações na tabela 4.1, a seguir:

Amostra	Local do servidor de Cache Web	Data da coleta	Número de linhas do arquivo	Tamanho médio dos objetos em bytes
01	New York – New York ny.us.ircache.net	28/11/2004	366.234	10.404
02	New York – New York ny.us.ircache.net	29/11/2004	20.530	9.929
03	Palo Alto – California pa.us.ircache.net	29/11/2004	140.636	10.541
04	Urbana Champaign – Illinois uc.us.ircache.net	30/11/2004	1.064.800	6.676

Tabela 4.1 – Arquivos cache Web

Cada arquivo armazena todos os pedidos do respectivo servidor durante as 24 horas do dia coletado. Os arquivos são compostos por registros, sendo um por linha, e cada linha com 10 campos no seguinte formato:

```
[1]1101686604.435 [2]130 [3]68.230.165.32 [4]TCP_HIT/200 [5]5330 [6]GET
[7]HTTP://a708.g.akamaitech.net/7/708/33/914e22fe44c641/images.citysearch
.com/mw/images/a3/af/57409p1.jpg [8]- [9]NONE/- [10]image/jpeg
```

Campo 1. indica o tempo em que o pedido da conexão foi atendido. Totaliza os segundos transcorridos desde 01 de janeiro de 1970, com arredondamento em milisegundos.

Campo 2. tempo de duração da conexão. Contabiliza o tempo entre o aceite da conexão até o último pacote transferido para o arquivo solicitado.

Campo 3. endereço IP do cliente

Campo 4. código de resposta HTTP

Campo 5. tamanho da resposta enviada ao cliente em bytes

Campo 6. método de requisição HTTP

Campo 7. URL requisitada

Campo 8. identificação do usuário (se não identificado é substituído por "-")

Campo 9. hierarquia e endereço do servidor

Campo 10. tipo do arquivo servido (exemplo: jpeg)

Neste estudo os registros foram separados pelo campo 10 - tipo do arquivo servido - e para cada um individualmente armazenados todos os valores do campo 5 - tamanho da resposta enviada ao cliente em bytes. Os arquivos resultantes foram utilizados como amostras para a caracterização das distribuições de probabilidade.

4.3.2 Amostras de servidores Web – Copa 98

A proposta da presente dissertação foi aplicada também a dados coletados de servidores Web que hospedaram o "site" da Copa do Mundo de 1998. São quase trinta e oito milhões de requisições que foram analisadas com o intuito de validar os resultados já obtidos.

Os arquivos analisados nesse experimento foram coletados dos servidores que abrigavam o "site" http://www.france98.com e estão disponíveis para uso no endereço http://ita.ee.lbl.gov/html/traces.html.

O "site" http://www.france98.com era responsável por disponibilizar informações sobre a copa do mundo de futebol, realizada na França em 1998. Fornecia aos fãs do esporte uma gama grande de informações como acesso em tempo real aos resultados das partidas, resultados dos jogos já encerrados, estatísticas, biografia dos jogadores bem como fotos das partidas e entrevistas com os técnicos e jogadores.

Além dos serviços descritos, eram disponibilizados "downloads" de softwares gratuitamente como protetores de tela e papéis de parede para os "desktops" com motivos futebolísticos.

Ao analisar os arquivos de "log" dos servidores, percebe-se que a partir do dia 10 de junho de 1998, o tráfego cresce exponencialmente, uma vez que o campeonato estava em curso, permaneceu popular e com alto número de acessos até rapidamente cair no

esquecimento quando do término do campeonato. O dia de maior acesso ao "site" foi o dia 30 de junho, quando mais de 73 milhões de requisições foram efetuadas.

Algumas das características observadas nos servidores que geraram os arquivos de "log" são:

- Oitenta e oito por cento de todas as requisições foram geradas por arquivos de imagem, um adicional de 10 por cento foram por arquivos HTML, indicando que o interesse da maioria dos usuários estava em arquivos estáticos (arquivos armazenados em cache);

- Quase dezenove por cento de todas as respostas foram "Não Modificadas" indicando que o cache teve um grande impacto na carga da Copa do Mundo comparando com o Web Server;

- A carga caracterizou-se por ser quase sempre em rajada, ao longo da escala de tempo (exemplo: horas) e a chegada destas rajadas era quase previsível;

- Durante os períodos de pico de interesse do usuário no site da Copa do Mundo, o volume de tráfego do cache aumentava dramaticamente.

As amostras utilizadas neste estudo são "logs" das requisições feitas ao site Web da Copa do Mundo de 1998, entre os dias 30 de Abril de 1998 a 26 Julho de 1998. Todos os acessos de um determinado dia foram coletados de cada um dos vários servidores utilizados no "site" da copa do mundo de 1998 e condensados em um ou mais arquivo para o dia em questão.

Durante este período de oitenta e oito dias, trinta e três diferentes servidores HTTP, espalhados geograficamente entre: Paris, na França; Plano, no Texas; Herndon, na Virginia; e Santa Clara, na Califórnia, atenderam a mais de um bilhão, trezentos e cinqüenta e dois milhões oitocentos e quatro mil, cento e sete requisições.

Os arquivos do Web site da Copa do Mundo de 1998 foram coletados inicialmente em formato de log. Este formato de "log" é chamado de Common Log Format [KRI01]. Com o objetivo de reduzir o tamanho dos mesmos e o tempo de análise, estes arquivos foram convertidos para formato binário. Para retornar ao formato original foram disponibilizadas ferramentas que permitem a recuperação dos dados. Estas ferramentas são disponibilizadas no site, juntamente com os logs.

A escolha das três amostras foi aleatória, apenas buscando um dia de cada mês disponível, conforme observa-se na tabela a seguir:

Amostra	Data da coleta	Total de registros	Tamanho médio dos arquivos	Total de bytes
32	27-maio-1998	4.146.069	579.971	24.046.020.242
57	21-Junho-1998	17.224.132	427.838	74.539.469.668
72	6-Julho-1998	16.760.999	377.540	63.375.851.864

Tabela 4.2 – Amostras do servidor Web – Copa 98

Os registros dos arquivos têm o formato e o significado abaixo listados:

[1]34600 [2][30/Apr/1998:21:30:17 +0000] [3]"GET [4]/images/hm_bg.jpg [5]HTTP/1.0" [6]200 [7]24736

Campo 1. Identificação do cliente. É um número inteiro único que indica originalmente o endereço IP que solicitou o pedido ao servidor Web. Esta modificação foi feita para garantir sigilo aos clientes.

Campo 2. Data e hora da requisição, convertido ao GMT - Greenwich Mean Time (+0200) para permitir a portabilidade

Campo 3. Método da requisição do cliente (exemplo: GET)

Campo 4. Tipo do arquivo servido (exemplo: HTML, IMAGE, etc). Este é o campo que usaremos para fazer a identificação do conteúdo.

Campo 5. Identifica a URL solicitada

Campo 6. Status. Este campo tem duas informações: versão do HTTP (exemplo: HTTP/1.0); e os outros 6 bits indicam o código de resposta (exemplo: 200 OK).

Campo 7. Tamanho em bytes da resposta

Os registros foram separados pelo campo 4 - tipo do arquivo servido - e para cada um individualmente armazenados todos os valores do campo 7 - tamanho da resposta enviada

ao cliente em bytes. Os arquivos resultantes foram utilizados como amostras para a caracterização das distribuições estatísticas.

4.4 Matriz de transição das amostras

Com a implementação do algoritmo proposto, na linguagem C, foram obtidas as matrizes de probabilidade de transição para os dados das amostras descritas. A seguir mostra-se os resultados encontrados para a amostra 1 do IRCache. Para as outras 3 amostras os resultados estão listados nos Anexos.

Matriz do número de transição entre tipos de arquivos - Amostra 1 – IRCache								
	HTML	JPEG	GIF	MPEG	PDF	XML	OUTROS	FIM
HTML	39837	15096	11230	131	91	4035	30848	4408
JPEG	15689	33286	9641	88	62	1466	15741	2784
GIF	11178	9564	21623	46	27	977	12402	2232
MPEG	130	57	57	52	0	11	151	29
PDF	88	40	26	0	8	3	51	17
XML	3979	1671	1133	15	5	1417	4492	152
OUTROS	31558	15936	140	52	4225	55234	3971	0
FIM	0	0	0	0	0	0	0	0

Tabela 4.3 – Matriz do número de transições entre tipos de arquivos - Amostra 1 - IRCache

Os valores apresentados na matriz do número de transições entre tipos de arquivos indicam o número de vezes que ocorreu a transição em questão. Por exemplo: na tabela 4.5, o valor 39.837, que está na posição [1,1] ou seja [HTML, HTML] significa que na amostra 1 houve 39.837 transições de um arquivo HTML chamando outro arquivo HTML. Ou seja, das 366.234 linhas do arquivo, 39.837 são arquivos HTML transmitidos que levaram a outro arquivo HTML transmitido. Na posição [1,2] ou seja [HTML, JPEG] verifica-se que houve 15.096 transições de um arquivo HTML chamando um arquivo JPEG. Ou seja, das 366.234 linhas do arquivo, 15.096 são arquivos HTML transmitidos que levaram a que um arquivo JPEG fosse transmitido, e assim sucessivamente para cada um dos pares [m,n] da matriz.

Matriz de probabilidade de transição entre tipos de arquivos – Amostra 1 - IRCache								
	HTML	JPEG	GIF	MPEG	PDF	XML	OUTROS	FIM
HTML	0,3770	0,1429	0,1063	0,0012	0,0009	0,0382	0,291911	0,0417
JPEG	0,1992	0,4226	0,1224	0,0011	0,0008	0,0186	0,199868	0,0353
GIF	0,1926	0,1648	0,3725	0,0008	0,0005	0,0168	0,213647	0,0385
MPEG	0,2669	0,1170	0,1170	0,1068	0	0,0226	0,310062	0,0595
PDF	0,3777	0,1717	0,1116	0	0,0343	0,0129	0,218884	0,0730
XML	0,3093	0,1299	0,0881	0,0012	0,0004	0,1102	0,349192	0,0118
OUTROS	0,2840	0,1434	0,0013	0,0005	0,0380	0,4971	0,035737	0
FIM	0	0	0	0	0	0	0	0

Tabela 4.4 – Matriz de probabilidade de transição entre tipos de arquivos - Amostra 1 - IRCache

Os valores apresentados na matriz de probabilidade de transição entre tipos de arquivos mostram a probabilidade de ocorrência da referida transição. Estes valores são obtidos dividindo o valor da célula pela somatória de todos os elementos da linha. Assim, na tabela 4.6, para a amostra 1 tem-se 37,70% de probabilidade de, dado que o objeto transmitido seja um HTML o próximo arquivo transmitido será outro HTML. Há 14,29% de probabilidade de, dado que o objeto transmitido seja um HTML o próximo arquivo transmitido será um JPEG, e assim sucessivamente.

A seguir mostra-se os resultados obtidos para a amostra do dia 37, dos arquivos coletados do servidor Web da Copa de 98.

Matriz do número de transição entre tipos de arquivos - Dia 37 – WC98

	HTML	JPEG	GIF	OUTROS	FIM
HTML	667244	45356	322024	14233	361
JPEG	40332	357734	226166	9203	164
GIF	398775	214426	8160041	111911	1835
OUTROS	21459	357734	117585	188039	353
FIM	0	0	0	0	0

Tabela 4.5 - Matriz do número de transições entre tipos de arquivos - Dia 37 – WC98

Os valores apresentados na matriz do número de transição entre tipos de arquivos indicam o número de vezes que ocorreu a transição em questão. Assim na tabela 4.7, o valor 667.244, que está na posição [1,1] ou seja [HTML, HTML] significa que na amostra 1 houve 667.244 transições de um arquivo HTML chamando outro arquivo HTML. Ou seja, das 4.146.069 linhas do arquivo, 667.244 são arquivos HTML transmitidos que levaram a outro arquivo HTML transmitido. Na posição [1,2] ou seja [HTML, JPEG] verifica-se que houve 45.356 transições de um arquivo HTML chamando um arquivo JPEG. Ou seja, das 4.146.069 linhas do arquivo, 45.356 são arquivos HTML transmitidos que levaram a que um arquivo JPEG fosse transmitido. E assim sucessivamente para cada um dos pares [m,n] da matriz.

Matriz de probabilidade de transição entre tipos de arquivos – Dia 37 – WC98

	HTML	JPEG	GIF	OUTROS	FIM
HTML	0,6359	0,0432	0,3069	0,0136	0,0003
JPEG	0,0637	0,5646	0,3570	0,0145	0,0003
GIF	0,0449	0,0241	0,9182	0,0126	0,0002
OUTROS	0,0313	0,5221	0,1716	0,2744	0,0005
FIM	0	0	0	0	0

Tabela 4.6 – Matriz de probabilidade de transição entre tipos de arquivos – Dia 37 - WC98

Os valores apresentados na matriz de probabilidade de transição entre tipos de arquivos mostram a probabilidade de ocorrência da referida transição. Estes valores são obtidos dividindo o valor da célula pela somatória de todos os elementos da linha. Na tabela 4.8, para a amostra 1 tem-se 63,59% de probabilidade de, dado que o objeto transmitido seja um HTML, o próximo arquivo transmitido será outro HTML. Há 4,32% de probabilidade de, dado que o objeto transmitido seja um HTML o próximo arquivo transmitido será um JPEG, e assim sucessivamente.

O próximo tópico fará o relato da caracterização das amostras através do tamanho dos arquivos transmitidos.

4.5 Caracterização do tamanho dos arquivos transmitidos

A tarefa de ajustar para cada tipo de arquivo Web um modelo de distribuição requer a análise de medições de cargas Web coletadas. O modo mais comum de especificar um modelo estatístico para um conjunto de dados é através do método visual chamado "quantile-quantile" ou gráfico da distribuição de freqüência acumulada. A construção do Histograma também apresenta resultados conclusivos na identificação da forma da curva da distribuição amostral dos dados.

O desafio está em evitar que não seja feita a distinção entre duas distribuições próximas, pois tentar aplicar um teste de bondade do ajuste, para amostras do tamanho das aqui estudadas, apresenta uma série de problemas, conforme será relacionado no capítulo 6 no tópico sobre testes de bondade do ajuste. Métodos baseados em dados tabulados (como o teste Qui-quadrado) podem gerar falta de exatidão, se aplicados a grandes amostras, assim como métodos baseados em funções de distribuições empíricas (como o teste de Anderson-Darling) também falham quando aplicados a grandes conjuntos de dados. Como a maioria das amostras Web é composta por imensos conjuntos de dados, as dificuldades listadas acima permanecem ao utilizar os testes de bondade de ajuste como método para definir um modelo de distribuição.

A definição dos tipos de arquivos usados neste estudo foi determinada com base na seguinte premissa: explicar o maior volume trafegado em bytes da amostra, a partir do menor número de tipos de arquivos. O objetivo desta abordagem foi simplificar o modelo proposto. Por exemplo, considera-se que construir um modelo analisando 4 tipos de arquivos que representem 85% do volume trafegado é mais simples analiticamente que um modelo com 12 tipos que representem 90% do volume.

É importante salientar que a amostra será determinante para esta definição. Por exemplo, no caso das amostras extraídas do IRCache verifica-se uma gama maior de tipos de arquivos trafegados, e logicamente o volume trafegado fica pulverizado nesta diversidade. Mesmo assim ainda pode-se concentrar os tipos de maior peso e aplicar o modelo. Já nas amostras da Copa de 98 há uma concentração nos tipos trafegados, conforme explicado na descrição das amostras, ficando restritos a uns poucos tipos. Neste caso o modelo proposto irá atingir um volume maior do total trafegado.

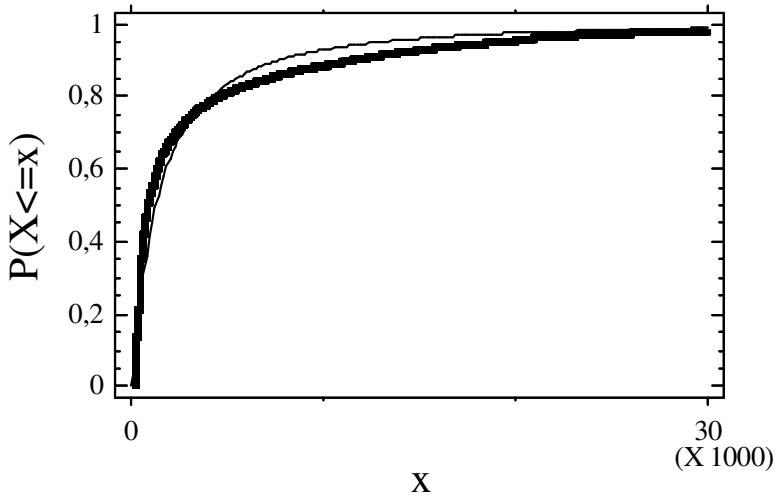
Desta forma, para alguns servidores Web, o modelo proposto será melhor que para outros, em virtude da caracterização do seu tráfego. Mas nos dois casos a simplificação em relação aos modelos atuais já denota um ganho significativo em virtude da eliminação das distribuições de cauda pesada.

Nada impede que sejam incluídos outros tipos de arquivos além dos aqui relacionados. Num estudo superficial com outros tipos de arquivos, que apresentaram um menor número de ocorrência, verificou-se que quase todas têm um bom ajuste à distribuição Logo Normal ou à distribuição Exponencial. Uma exceção foi o tipo octet-stream que pela sua própria definição, conforme a RFC 822 do MIME, engloba formatos não reconhecidos. Se um formato de arquivo não é identificado, ele será registrado como “não suportado”, ou "application/octet-stream". Assim sua variabilidade de “tipos” faz com que tenha o mesmo comportamento da análise do tráfego Web com um todo.

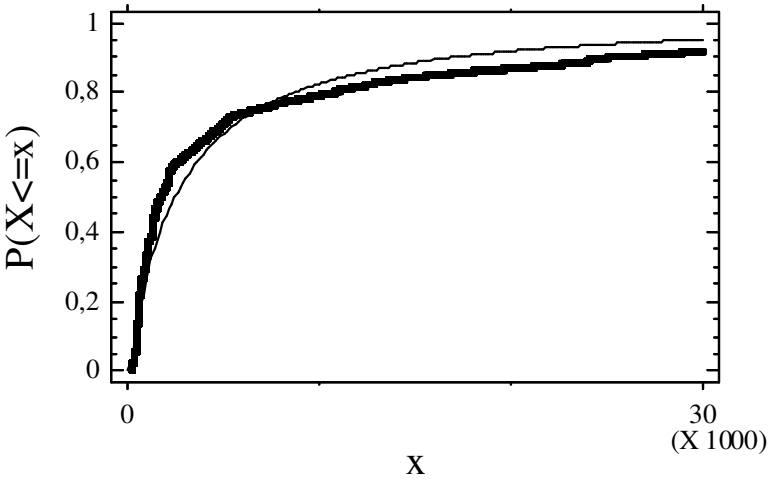
Outro ponto que merece destaque é com relação à subdivisão dentro de um mesmo tipo de arquivo. Este fato ocorreu para uma amostra da Copa de 98 e fez com que uma subdivisão do tipo ZIP trouxesse um melhor ajuste as distribuições amostrais. Ao estudar a amostra com os arquivos do tipo ZIP trafegados com até 350 bytes separadamente dos arquivos com mais de 350 bytes foram observadas duas distribuições distintas para cada conjunto. No primeiro caso uma distribuição normal e no segundo Logo Normal. Este comportamento se deve ao fato de que há componentes temporais no tráfego observado. Estes componentes temporais são comprovadamente característicos do tráfego Web [BOX94] [XUE99] [LIU99].

Ao analisar o tráfego das amostras na sua totalidade verifica-se a presença da auto-similaridade e a distribuição amostral dos dados apresenta LRD. Porém, ao separar os tipos de arquivos, como proposto nesta dissertação, consegue-se modelar grande parte do tráfego com distribuições sem cauda pesada, e isto traz inúmeras vantagens ao simplificar os cálculos matemáticos no processo de geração de tráfego sintético.

Foi realizada a caracterização do tamanho do objeto transmitido para alguns dos principais tipos de arquivos da amostra 1 do IRCache. O objetivo era o de realizar testes de aderência para a obtenção da distribuição de probabilidade acumulada da amostra. Os procedimentos adotados serão descritos no capítulo 5.



(a)



(b)

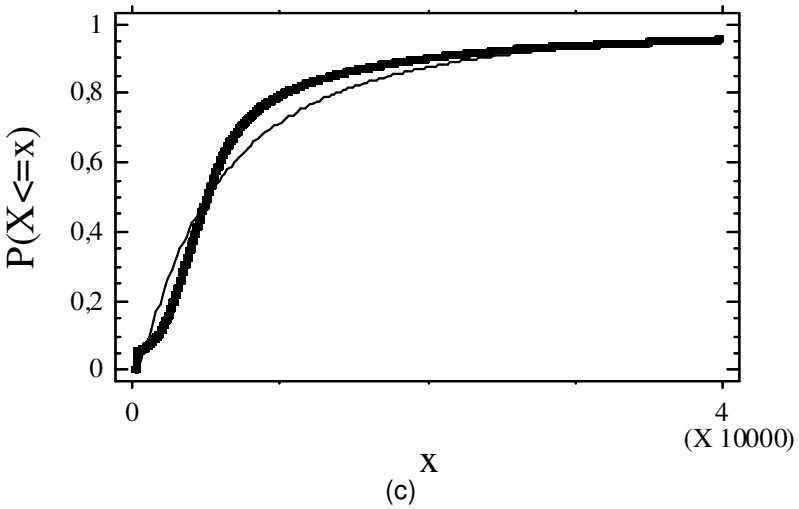


Gráfico 4.1 – Probabilidade acumulada dos tamanhos dos arquivos da amostra 1 – IRCache para os tipos GIF (a), HTML (b) e JPEG (c)

Nos gráficos 4.1 (a), (b) e (c), os dados da amostra foram comparados com a distribuição teórica Logo Normal. Conforme pode-se visualizar, o ajuste dos dados da amostra (x) à função densidade de probabilidade P (X <=x), para a distribuição Logo Normal é bastante aderente. Basta verificar que o traço mais escuro, que são os dados amostrados distribuídos de acordo com uma Logo Normal, usando a média e variância do próprio conjunto de dados, está muito próximo ao traço mais fino, que representa o ajuste da curva à distribuição Logo Normal teórica, utilizando também a média e variância da amostra.

Ou seja, as duas curvas possuem a mesma média e variância dos dados amostrados, porém no traço escuro o ajuste da curva é feito para os dados em si e no traço claro, para a distribuição Logo Normal teórica. Como a quantidade de dados das amostras aqui analisadas são consideradas grandes no conceito estatístico (> 500 [RAJ91]), o ajuste visual é muito bom.

4.6 Conclusão

Sob este novo prisma, que considera o tipo de arquivo trafegado, é possível contorna o problema de tratar variáveis com distribuição de cauda pesada. A abordagem proposta mostra que as distribuições encontradas para os tipos de arquivos, das amostras estudadas, não apresentam LRD e portanto não apresentam auto-similaridade. Mesmo assim o tráfego agregado, que corresponde ao agrupamento dos diversos tipos de arquivos transmitidos, apresentou auto-similaridade como era esperado.

O método aqui proposto e aplicado, mostra que é possível modelar o tráfego agregado a partir da distribuição dos tipos de arquivos com distribuições estatísticas sem LRD para as atuais amostras de carga Web. Isto foi provado para os tipos de arquivos mais comumente encontrados no tráfego Web, e que representam o maior volume de dados trafegados. O método pode ser aplicado a outros tipos de arquivos não estudados aqui.

A matriz de probabilidade de transição traz características comportamentais do usuário ao interagir com as páginas Web. Inicialmente pela sua escolha e posteriormente pela forma de agrupamento dos arquivos na construção da página, cujo resultado mostra ao usuário a informação solicitada.

A aplicação deste método para as amostras de duas fontes diferentes mostrou-se eficiente para obter os dados necessários para alimentar um simulador para geração de tráfego sintético Web. Tanto as matrizes de transições entre os tipos e conseqüentemente as matrizes de probabilidade de transições entre os tipos de arquivos, foram similares nos resultados encontrados. Ou seja, o comportamento mostrou repetir-se, guardadas as proporções de tráfego.

É importante ressaltar que o trabalho com as imensas amostras de tráfego Web foram um desafio à parte. Tanto a implementação do software (disponível no CPGEI como relatório técnico de número 1/2006), quanto a manipulação dos dados, são atividades que agregam um grau maior de dificuldade ao processo.

Os objetivos pretendidos foram atingidos, uma vez que todas as informações necessárias para os passos seguintes de simulação e geração de tráfego sintético, foram alcançadas. Foi possível mostrar também que os tipos de arquivos analisados individualmente não possuem LRD.

A aplicação do modelo aqui proposto não elimina a auto-similaridade do tráfego Web que está associada ao funcionamento das aplicações Web [NEV02], simplesmente traz uma nova abordagem que minimiza o tratamento dos efeitos apresentados pelo fenômeno da auto-similaridade (por exemplo: a não convergência do desvio padrão).

Além da geração de carga sintética Web, esta nova abordagem pode levar a criação de técnicas diferentes para melhoria de desempenho de servidores (por exemplo: algoritmos de descarte de dados em cache).

O fato das distribuições dos tipos de arquivos trafegados terem decaimento Exponencial ou Sub Exponencial traz a possibilidade de análise de desempenho do servidor através de cadeias de Markov.

Pode-se também caracterizar o tráfego de acordo com o servidor Web estudado. Ou seja, os tipos de arquivos a serem considerados para a montagem da matriz de probabilidade de transição podem ser escolhidos de acordo com o conteúdo do servidor alvo. Isto é interessante pois permite resultados mais apurados para cada servidor individualmente.

5 Estudo Estatístico

O estudo estatístico para o presente trabalho está dividido em duas etapas: primeiro encontrar as distribuições para os tamanhos dos arquivos transmitidos, para o qual são utilizadas técnicas ditas Paramétricas e Não-Paramétricas; e o segundo é tentar aplicar testes de ajuste de bondade para validar as distribuições encontradas anteriormente.

5.1 Estimação de Densidades

São abordados alguns métodos para estimação de funções de densidade de probabilidade, também conhecidas como função da distribuição dos dados da amostra. Existem duas formas de obter a distribuição da amostra de um conjunto de dados: utilizando técnicas Paramétricas ou Técnicas Não-Paramétricas.

A estimação Paramétrica é feita a partir da suposição de que um determinado modelo probabilístico é adequado para os dados em questão e uma vez assumido o modelo, é necessário estimar seus parâmetros.

Os métodos Não-Paramétricos baseiam-se nos próprios dados para encontrar o formato da distribuição de probabilidade. A forma mais simples de estimação é utilizar o próprio histograma de freqüência. O resultado dá uma noção da forma da curva da função densidade de probabilidade.

A teoria de Inferência Estatística está fortemente baseada na utilização de distribuições de probabilidades. Algumas destas distribuições já são conhecidas e amplamente utilizadas na literatura estatística. Entretanto, alguns fenômenos geram dados que não são compatíveis com estas distribuições.

Para melhor conhecer a distribuição de probabilidade de dados em uma amostra, o procedimento tradicional é a construção de um histograma de freqüências, no caso de variáveis aleatórias contínuas, e do gráfico de barras para variáveis aleatórias discretas.

O histograma pode ser visto como um método “ingênuo” de estimação da função de densidade de uma variável aleatória. Como todo procedimento de estimação, a sua qualidade deve ser monitorada e, por isto, alguns métodos de construção de histogramas são apresentados e discutidos na próxima sessão.

5.1.1 Construção de Histogramas

O histograma consiste na representação da freqüência de observações em um intervalo através de um retângulo. Se todos os intervalos têm igual comprimento, a altura do retângulo é diretamente proporcional à quantidade de observações.

A construção de um histograma é feita após a definição de 3 elementos :

- *L* - Número de intervalos de classe
- *h* - Comprimento dos intervalos de classe
- *W* - Amplitude amostral (máximo-minimo).

A relação entre estes 3 elementos é dada por:

$$h = \frac{W}{L} \qquad (1)$$

Em (1) o único elemento conhecido é o *W* (a amplitude da amostra), restando determinar um dentre os dois elementos restantes: *h* e *L*.

5.1.2 Fórmulas para a determinação do número de intervalos

Algumas fórmulas foram desenvolvidas para a definição do número ”ideal” de intervalos de classe para que seja feita uma primeira representação da função de densidade de probabilidade de uma variável aleatória com base em uma amostra de tamanho *n*. Um bom levantamento sobre os métodos desenvolvidos para construção de histogramas pode ser encontrado em [SCO92]:

Fórmula de Sturges

$$L = [1 + log_2 n] \qquad (2)$$

A Fórmula de Sturges foi criada sob o argumento de que *n*, o tamanho da amostra, é uma potência de 2, de forma que:

$$n = 2^{L-1}$$

Deste modo, Sturges idealizou que as freqüências dos intervalos de classe apresentariam um comportamento simétrico, conforme os coeficientes do binômio de Newton.

$$2^{L-1} = \binom{L-1}{0} + \binom{L-1}{1} + \binom{L-1}{2} + \ldots + \binom{L-1}{L-1}$$

Veja o exemplo para n=16:

$$2^{L-1} = \binom{5-1}{0} + \binom{5-1}{1} + \binom{5-1}{2} + \binom{5-1}{3} + \binom{5-1}{4}$$

Efetuando os cálculos acima,

$$2^4 = 1 + 4 + 6 + 4 + 1$$

As freqüências 1*;* 4*;* 6*;* 4*;* 1 são aquelas que Sturges idealizava encontrar nos dados e, portanto, o número de intervalos de classe *L*, para *n* = 16, é igual a 5.

Em seguida são apresentadas duas outras formulações para a determinação de *L*.

Fórmula de Velleman

$$L = [2\sqrt{n}] \qquad (3)$$

Fórmula de Dixon & Kronmal

$$L = 10 log_e n \qquad (4)$$

As formulações apresentadas nas equações (2), (3) e (4) são algumas alternativas, dentre outras, para a determinação do número de intervalos de classe.

Abaixo são apresentadas algumas regras práticas, construídas através de evidência empírica, para realizar a melhor escolha dentre as apresentadas acima.

- A formula de Sturges é mais adequada para pequenas amostras $n < 50$. Em grandes amostras, ela tende a gerar um histograma demasiadamente suave.
- A fórmula de Velleman é mais adequada para amostras de tamanho médio $50 < n < 100$
- A fórmula de Dixon & Kronmal é mais adequada para amostras de tamanho $n > 100$
- A fórmula de Dixon & Kronmal também é um limite superior para o número L de intervalos de classe em gráficos do tipo histograma.

5.1.3 Fórmulas para a determinação do comprimento do intervalo de classe

Como o histograma é uma estimativa $\hat{\int}_n(x)$ da função densidade de probabilidade $f(x)$, [SCO79] sugere o seguinte princípio: escolher o comprimento do intervalo de classe que possa minimizar (5).

$$E\left[\hat{\int}_n(x) - \int(x)\right]^2 \qquad (5)$$

Entretanto, para minimizar a quantidade acima, é necessário o conhecimento da função densidade de probabilidade *f*(*x*). Utilizando a distribuição Normal como padrão, o comprimento do intervalo é aproximadamente:

$$h_n = 3.49\, s n^{-1/3} \qquad (6)$$

onde:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x)^2}{n - 1}}$$

é o desvio padrão amostral.

Como alternativa mais robusta, [FRE81] sugerem que tente-se minimizar o desvio absoluto médio em (7).

$$E\left|\hat{\int}_n(x) - \int(x)\right| \qquad (7)$$

Utilizando, novamente, a distribuição Normal como referência, o comprimento do intervalo é aproximadamente:

$$h_n = 1.66 s\left(\frac{\log_e n}{n}\right)^{1/3} \qquad (8)$$

Em [FRE81] os mesmos autores ainda propuseram utilizar uma medida mais robusta para determinação do comprimento do intervalo de classe com base na amplitude interquartílica conforme (9).

$$h_n = \frac{2AIQ}{n^{1/3}} \qquad (9)$$

Em (9), *AIQ* é a amplitude entre os quartis.

De forma resumida, a Figura 6.1 mostra as 3 alternativas apresentadas, sob a suposição da distribuição normal. É importante notar que ao determinar primeiramente h, ao invés de L, o histograma passa a ser um estimador (semi)-paramétrico para a densidade f.

Scott	Freedman e Diaconis I	Freedman e Diaconis II
$h_n = 3.49sn^{-1/3}$	$h_n = 1.66s\left(\frac{\log_e n}{n}\right)^{1/3}$	$h_n = \frac{2AIQ}{n^{1/3}}$

Figura 5.1 – Comprimentos dos Intervalos de Classe por 3 Diferentes Critérios

5.2 Estimação Paramétrica de Densidades

A estimação paramétrica é feita sob a suposição de que um determinado modelo probabilístico é adequado para os dados em questão.

Alguns modelos probabilísticos conhecidos são adequados para descrever determinados experimentos aleatórios. Um exemplo clássico é o tempo de espera de um cliente em uma fila de banco que é geralmente descrito, em termos probabilísticos, pela distribuição Exponencial cuja densidade é apresentada em (10).

$$f_X(x) = \lambda e^{-\lambda t} \qquad x > 0, \lambda > 0 \qquad (10)$$

Uma vez assumido o modelo probabilístico, basta estimar os parâmetros deste modelo, o que pode ser feito por diferentes métodos. Em (10), a densidade é completamente especificada se houver o conhecimento do parâmetro, geralmente estimado a partir de uma amostra.

Para citar os métodos mais conhecidos para estimação de parâmetros de uma distribuição: método dos momentos, método dos mínimos quadrados e método da máxima verossimilhança. Detalhes sobre estes métodos podem ser encontrados em [LIN96].

5.3 Estimação Não-Paramétrica de Densidades

Um dos problemas dos métodos de estimação paramétrica é assumir que um determinado modelo probabilístico seja adequado para representar os dados.

Recentemente, os métodos não paramétricos para estimação de densidades ganham força pois tem como base os próprios dados para estimar $f_X(x)$.

O método mais simples de estimação não paramétrica é utilizar o próprio histograma de freqüência conforme visto nas primeiras seções deste capítulo. Entretanto, o resultado pouco suave oferecido pelo histograma não é compatível com o comportamento real da função densidade de probabilidade.

Os métodos do tipo Kernel (palavra que pode ser traduzida como ”núcleo”) ajustam o valor da densidade a cada ponto, levando em consideração dois elementos básicos : o tipo de Kernel e a amplitude da vizinhança na qual é feito este ajuste local [ROS56] [PAR62].

5.3.1 Estimação Kernel

A Estimação Kernel é um método não paramétrico para estimação de densidades que pode ser visto como um aperfeiçoamento do histograma pois gera uma função suavizada.

Este método foi apresentado por [ROS56] e [PAR62] e posteriormente com o aumento da capacidade de processamento computacional tornaram-se mais acessíveis.

Para uma amostra $X_1, \ldots, X_n$, o estimador via Kernel, conforme [HÄR90] é definido através da equação (11).

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \qquad (11)$$

Em (11):

- $K(.)$ é uma função suave com as propriedades de uma função de densidade de probabilidade.
- h é o tamanho da janela (*binwidth)* utilizada para suavizar o valor de um ponto e corresponde ao comprimento do intervalo de classe no histograma.

Dentre as várias possíveis escolhas para a forma funcional de $K(.)$ estão as funções: uniforme, tricúbica, gaussiana e Epanechnikov.

O Kernel gaussiano é o mais utilizado na prática. Entretanto, a qualidade da estimação proporcionada dependerá mais do valor escolhido para h do que na forma funcional do kernel. Um valor muito baixo para a largura da janela pode produzir uma curva pouco suave, com várias modas (pontos de máximo local). Por outro lado, um valor alto estipulado para h pode gerar uma super suavização.

Uma das grandes vantagens da Estimação Kernel é ilustrar em uma curva os picos da distribuição dos dados, desde que escolhido o tamanho correto para a largura da janela.

5.4 Ajuste de Densidades aos dados das amostras

Nesta seção são ajustadas densidades paramétricas e não paramétricas ao conjunto de dados que ilustram o tamanho de arquivos de um determinado tipo acessado pela internet, conforme descrito no capítulo 4.

O software estatístico utilizado na obtenção dos resultados aqui apresentados é o [R]. Ele é amplamente conhecido no mundo acadêmico e tem distribuição livre.

A seguir é mostrado o resultado obtido da técnica acima descrita. As linhas finas nos gráficos apresentam a curva da distribuição densidade de probabilidade padrão e as linhas escuras mostram a aderência dos dados à distribuição.

Primeiro os dados de tamanho de arquivos transmitidos para o tipo GIF da amostra 1 do IRCache.

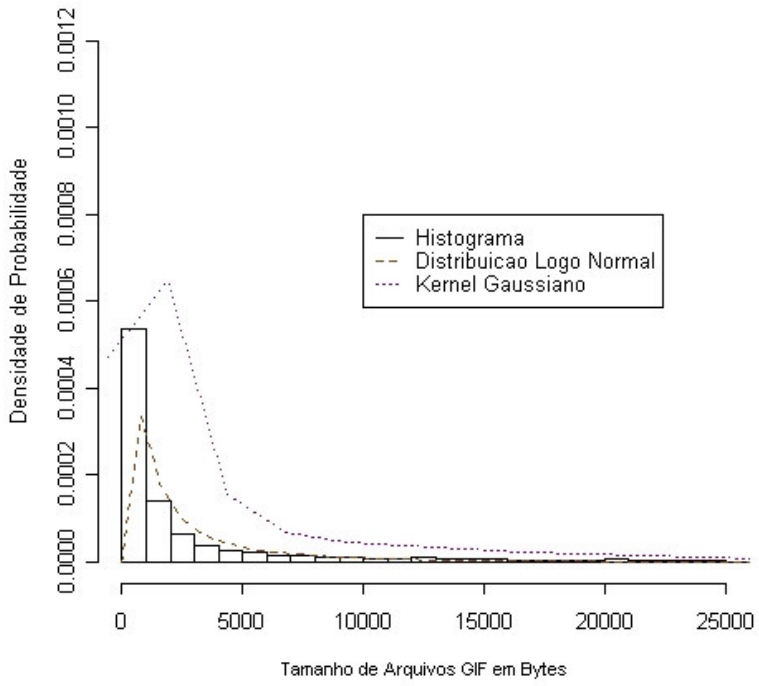


Gráfico 5.1 – Estimação Kernel do ajuste dos dados GIF amostra 1 – IRCache

Como pode ser observado no gráfico 5.1, o ajuste dos dados amostrados à curva da distribuição Logo Normal é excelente e apresenta apenas pequenas variações que são aceitáveis.

Os retângulos do histograma, no gráfico 5.1, naturalmente mostram uma distribuição de decaimento exponencial. Da mesma forma o ajuste da curva pontilhada, obtida pelo método Kernel Gaussiano, imita o ajuste da distribuição Logo Normal tracejada. Verifica-se que ajuste do Kernel Gaussiano acompanha de forma mais próxima a distribuição natural dos dados, uma vez que a curva gerada é ajustada gradativamente, conforme o método propõe.

Agora os dados de tamanho de arquivos transmitidos para o tipo HTML da amostra 1 do IRCache.

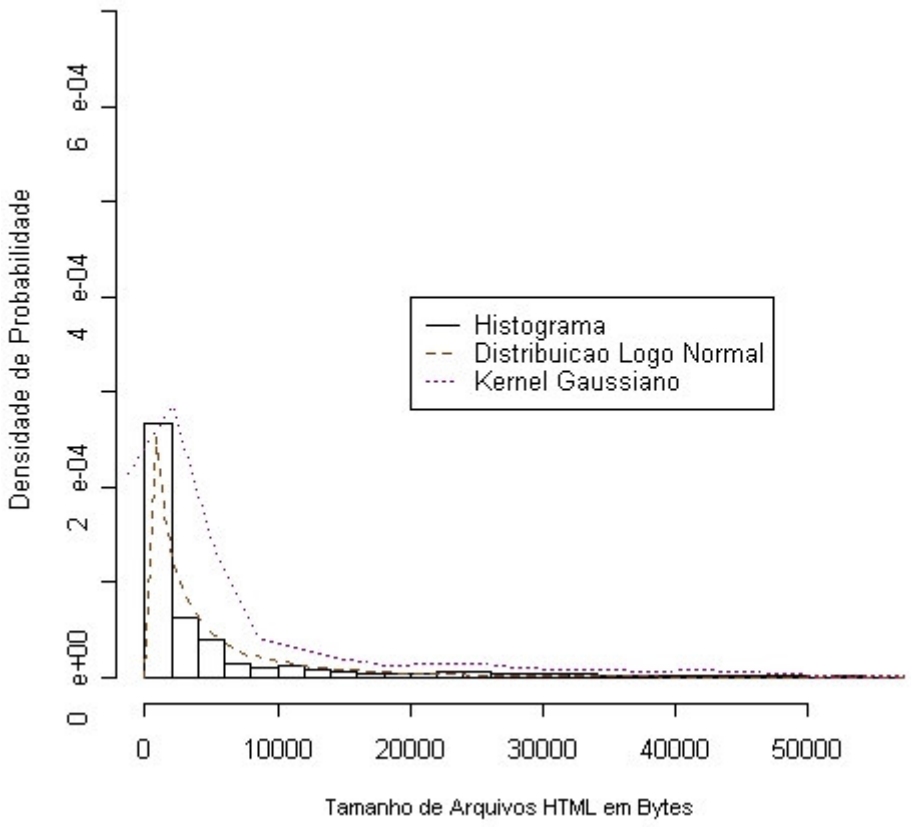


Gráfico 5.2 – Estimação Kernel do ajuste dos dados HTML amostra 1 – IRCache

No gráfico 5.2 verifica-se que o ajuste dos dados amostrados à curva da distribuição Logo Normal é visualmente bom. O histograma também denota que trata-se de uma distribuição Logo Normal. No ajuste Kernel é suavizada a queda da curva pois este método faz uma relação constante entre o valor anterior, com o atual e o futuro, levando em conta os parâmetros dos dados. No caso do histograma, um método Não-Paramétrico, apenas os valores da amostra são considerados. Desta forma, a característica do tráfego Web, de possuir uma imensa quantidade de pequenos arquivos e alguns grandes arquivos, fica evidenciada.

Em seguida os dados de tamanho de arquivos transmitidos para o tipo JPEG da amostra 1 do IRCache.

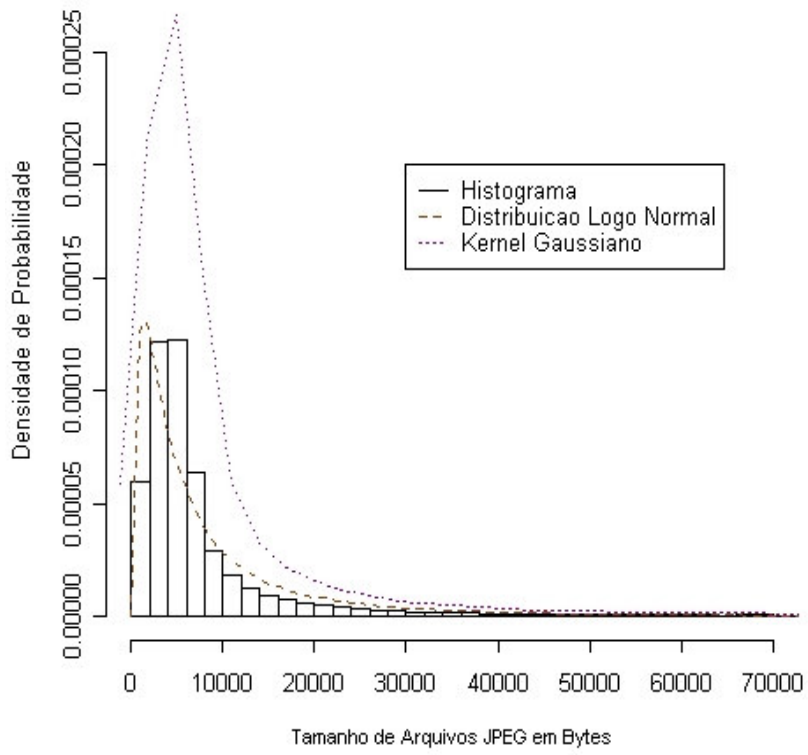


Gráfico 5.3 – Estimação Kernel do ajuste dos dados JPEG amostra 1 - IRCache

Como pode ser observado o ajuste dos dados amostrados à curva da distribuição Logo Normal é excelente apresentando apenas pequenas variações. Na amostra 3, que está

postada nos Anexos, é praticamente perfeito. Além disto, a tentativa de ajustar outras distribuições mostrou resultados piores, caracterizando que os conjuntos de dados estão mais próximos a Logo Normal.

A tabela 5.1 mostra o percentual que cada tipo de arquivo analisado representa, em bytes, sobre o tráfego total das amostras. Pode-se verificar que com apenas 3 tipos de arquivos, podem ser modelados, em média 50% do volume trafegado. Há pequenas variações dependendo do tipo de serviço prestado pelo servidor que este cache Web atende. Estas variações porém não são significativas dentro de um mesmo universo, ou seja, amostras retiradas de um mesmo servidor tendem a ter um comportamento similar. Assim a modelagem feita para um servidor A, poderá ser usada enquanto este servidor mantiver o mesmo tipo de serviço. Por exemplo: um servidor que atende a clientes de uma biblioteca terá o mesmo comportamento de tráfego enquanto seus clientes forem usuários de biblioteca. Se este servidor Web passar a atender também clientes de um banco, será necessário remodelar o tráfego pois este novo serviço pode implicar em tipos de arquivos novos e conseqüentemente a proporção poderá se alterar.

Não apenas o percentual de tráfego se manteve praticamente constante entre as amostras analisadas de um mesmo servidor, também a matriz de probabilidade de transição apresenta um quadro de similaridade entre amostras do um mesmo servidor e serviço.

Vale destacar que os resultados aqui obtidos não são influenciados por sazonalidade. Quando o tráfego aumenta, as transições e o volume dos tipos aumentam proporcionalmente. Outro fator que elimina a sazonalidade é o fato das amostras trazerem a coleta para as 24 horas de um determinado dia. Assim a unidade de trabalho dia apresenta o mesmo comportamento entre si. Todos este fatores se devem também ao fato do modelo aqui proposto captar o comportamento do usuário, que se repete para os serviços oferecidos por um determinado servidor.

TIPO DE ARQUIVO	% AMOSTRA 1	% AMOSTRA 2	% AMOSTRA 3	% AMOSTRA 4	MÉDIA
GIF	5,84	7,86	5,98	53,34	18,25
HTML	22,11	15,71	20,94	8,34	16,76
JPEG	20,88	25,86	10,25	0,96	14,49
Outros	51,17	50,57	62,83	37,36	49,50

Tabela 5.1 – Percentual por tipo de arquivo em relação ao volume trafegado em bytes - IRCache

Um fator que irá causar mudanças neste cenário é a implementação de outros tipos de arquivos na tecnologia de desenvolvimento Web. Por exemplo: outros tipos de imagens que passem a serem implementadas.

A tabela 5.2 mostra os valores dos parâmetros, média e variância, encontrados para as amostras do IRCache analisadas. Estes são os parâmetros que o software estatístico R utilizou para construir as curvas analisadas no ajuste dos dados a distribuição Logo Normal e ao ajuste Kernel Gaussiano. Estes parâmetros também serão utilizados para alimentar o simulador que irá gerar o tráfego Web sintético.

AMOSTRA	TIPO DE ARQUIVO	NÚMERO DE REGISTROS	INTERVALO DOS VALORES EM BYTES	μ	σ
1	HTML	102.321	159 – 1.673.083	7638,8	22793,7
	GIF	56.288	175 – 1.250.758	4156,8	12096,8
	JPEG	75.530	172 – 1.532.433	11068,0	27350,9
2	HTML	4.461	163 – 909.680	7666,5	21346,9
	GIF	4.383	204 – 162.775	3905,0	9090,46
	JPEG	6.650	201 – 315.708	8466,2	15501,7
3	HTML	56.740	164 – 639.150	5622.7	15540,1
	GIF	26.170	150 – 637.656	3482,0	13287,7
	JPEG	16.656	171 - 855.062	9374,7	21155,3
4	HTML	88.734	164 – 970.442	8934.5	21175,6
	GIF	567.921	200 – 4.076.239	1060.5	13482,0
	JPEG	10.169	217 – 1.746.341	6195,8	18054,2

Tabela 5.2 – Parâmetros da distribuição Logo Normal para as amostras IRCache

Para as amostras do site da Copa do Mundo de 1998 88,16% de todos os registros solicitados são arquivos de imagem (GIF, JPEG, etc) e 9,85% arquivos de HTML [ARL00]. Com relação ao percentual de bytes transferidos, as imagens correspondem a 35,02 % e arquivos do tipo HTML são 38,60%. Assim sendo, o presente estudo explica em média 98,01% dos pedidos feitos e 73,62% dos bytes trafegados.

Esta característica de quase 100% dos pedidos em servidores Web serem classificados como imagens ou HTML já foi observado anteriormente em outros trabalhos [ARL97]. Este é um dado importante e que confirma os resultados apresentados, uma vez que estamos trabalhando exatamente estes tipos de arquivos em nossas análises.

5.5 Teste de bondade do ajuste (Goodness-of-fit)

Para determinar a distribuição dos objetos para conjuntos de dados grandes, como as amostras aqui estudadas, é reportado que os testes de bondade do ajuste, conhecidos também por testes de aderência, como o Qui-quadrado (λ^2) ou o Kolgomorov-Smirnov (K/S) falham. E é por este motivo que muitos pesquisadores têm utilizado métodos empíricos em seus trabalhos, através da construção de histogramas. Existem várias explicações para a falha dos testes de aderência:

1. Para grandes conjuntos de dados o teste se torna muito preciso (e qualquer pequeno desvio faz o teste falhar);

2. A amostra pode não ser estacionária [ter estacionariedade forte (existe esperança e variância) e estacionariedade fraca (existe esperança)];

3. A amostra não é independente (então pode ser uma série temporal e a análise deve seguir os métodos ARIMA, AR, MA, etc.).

Se apenas um dos pontos acima for verdadeiro, isto é suficiente para que o teste falhe. Os conjuntos de dados aqui estudados são grandes e se enquadram no caso 1. Portanto, há grande chance dos testes de aderência falharem, mesmo assim, será feita uma tentativa de utilização dos testes de ajuste de bondade.

Inicialmente, é verificado o ajuste da distribuição acumulada de probabilidade dos dados da amostra, cujo resultado é apresentado no gráfico 5.4.

Para realizar o teste de aderência, a distribuição de probabilidade acumulada da amostra foi comparada com distribuições clássicas de probabilidade. A distribuição Logo Normal obtém uma boa aderência a praticamente todos os tipos de arquivos quando analisados separadamente. A gráfico 5.4 mostra uma comparação entre a distribuição acumulada teórica (Logo Normal) e a distribuição amostral dos tamanhos dos arquivos do tipo JPEG da amostra 3 do IRCache. O eixo horizontal x representa o tamanho do objeto transmitido e o eixo vertical P(X <= x) indica a probabilidade acumulada. Verifica-se visualmente que o ajuste é bom. Pela técnica não-paramétrica do histograma e pelo método paramétrico do ajuste de Kernel Gaussiano também foi comprovado que os dados tem distribuição Logo Normal.

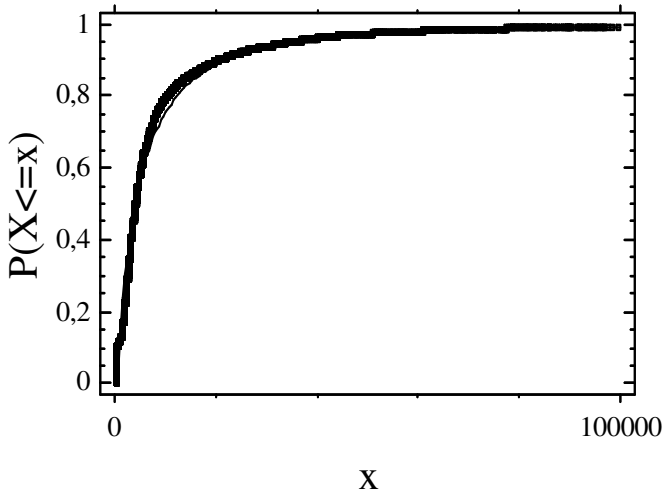


Gráfico 5.4- Probabilidade acumulada JPEG amostra 3 - IRCache

Ao aplicar os testes de bondade do ajuste, a hipótese de que os dados das amostras por tipo de arquivos provêm de uma distribuição Logo Normal, foi rejeitada com um intervalo de confiança de 99%. Os valores calculados nos testes de aderência, para o conjunto de dados JPEG da amostra 3 do IRCache, foi:

λ^2 = 3095,38 P-valor = 0,0

K/S = 0,095 P-valor < 0,1

A^2 = 259,92 P-valor < 0,1

O P-valor indica o valor do teste específico calculado a partir dos dados da amostra, comparados com os valores críticos das respectivas tabelas estatísticas, usando-se um intervalo de confiança de 99%.

O teste λ^2 dividiu a amostra em blocos e comparou o número de observações em cada classe com o número esperado, baseado na forma da distribuição alvo. Este teste pode ser usado para qualquer distribuição [RAJ91], porém no caso de distribuições contínuas, como a

Logo Normal, os valores encontrados são apenas aproximações. Sua aplicação ideal é para grandes amostras e distribuições discretas.

O K/S verifica a distância máxima entre a função distribuição acumulada dos dados da amostra 3 JPEG do IRCache e a distribuição teórica atribuída (Logo Normal). Neste caso a distância máxima foi de 0,095. Este teste é mais adequado a amostras pequenas e com distribuição contínua [RAJ91].

O teste K/S é baseado na diferença entre os valores observados e esperados das funções de probabilidade acumulada, enquanto o teste λ^2 é baseado na diferença entre o valor observado e a probabilidade hipotética para este valor. O K/S usa cada observação da amostra sem qualquer agrupamento, enquanto o teste λ^2 requer que as observações sejam agrupadas em células menores. Neste sentido, K/S faz um uso melhor dos dados da amostra. Um dos problemas em usar o teste λ^2 está na seleção das células. A escolha das células pode afetar a conclusão mas não há regras claras para guiar a escolha dos tamanhos apropriados. Com o teste K/S isto não é necessário. O teste λ^2 é sempre uma aproximação, enquanto o teste K/S provê todos os parâmetros para as distribuições conhecidas.

Mesmo a aplicação do teste A^2 , usado no SURGE, também não obteve resultado satisfatório.

Como em todos os testes, o P-valor é menor que 0,01, significa que pode ser rejeitada a hipótese de que os dados tenham uma distribuição Logo Normal, com 99% de confiança. Este resultado se repete para todos os conjuntos de dados verificados. Em um trabalho recente [GON05] são apresentadas as dificuldades na realização de testes de aderência entre dados reais e este tipo de distribuição. As tabelas estatísticas não estão preparadas para conjunto tão grandes de dados e portanto a tendência é de rejeitar a hipótese de pertencer à distribuição.

Estes testes de aderência são condições necessárias porém não suficientes para determinar de qual distribuição de probabilidade os dados da amostra provêm. Isto porque seu campo de atuação é limitado, de acordo com o tipo de dado e tamanho da amostra, como visto anteriormente.

Conhecendo esta limitação, os testes de aderência foram realizados para comprovar que grandes amostras implicam na falha dos mesmos. Em virtude disto, outras técnicas foram empregadas com o objetivo de validar os resultados obtidos (histograma, Kernel Gaussiano, etc).

6 Conclusão e trabalhos futuros

Nesta dissertação foram abordados: a evolução do tráfego de rede; o funcionamento da arquitetura TCP/IP; e os padrões usuais do tráfego da rede em relação à voz. A seguir foram apresentados os modelos de tráfego existentes para a rede atual de dados, e a motivação para a proposta de uma nova abordagem de modelagem de tráfego como extensão ao SURGE. Então, descreve-se esta abordagem para a modelagem do tráfego Web bem como uma validação da sua aplicação. Finalmente, apresenta-se o estudo estatístico que valida os resultados encontrados e as conclusões com as contribuições atingidas.

6.1 Conclusão

O modelo auto similar tem sido utilizado para caracterizar o tráfego produzido por servidores Web. Este modelo apresenta dificuldades para realização de estudos analíticos devido à extrema variabilidade [WIL00]. Uma explicação para o fenômeno da auto-similaridade do tráfego é a distribuição de probabilidade dos objetos transmitidos pelo servidor. Em muitos modelos, propostos atualmente, o conjunto de objetos transmitidos é caracterizado por uma LRD [CRO95].

O modelo proposto nesta dissertação captura as características do tráfego pelos tipos de arquivos transmitidos. Mostra-se aqui que cada tipo pode ser modelado de acordo com a distribuição Logo Normal. Cada sessão é mostrada então como uma seqüência de arquivos e estes arquivos são modelados através de uma máquina de estados. Este modelo mostrou-se adequado para representar de maneira muito precisa o comportamento do servidor. Para caracterizar o tráfego de um servidor Web, o modelo proposto necessita das seguintes informações:

1) Tipos de arquivos transmitidos pelo servidor;

2) Média e desvio padrão do tamanho para cada tipo de arquivo. A base de dados estudada, apresenta melhor aderência à distribuição Logo Normal;

3) Probabilidades de transição de estados para transmissão de tipos de arquivos em uma sessão;

4) Intervalo entre chegada de sessões.

Os dados necessários podem ser extraídos dos arquivos de log de servidores Web. Com estas informações, o modelo pode ser utilizado para realização de simulações e estudos analíticos (inclusive utilizando cadeias de Markov).

Estes dados estão sendo utilizados como entrada em um simulador que está em desenvolvimento pelo Doutorando Carlos Marcelo Pedroso, nesta instituição de ensino. O simulador utiliza o modelo proposto para calcular a curva de desempenho de um servidor (tempo médio de resposta vs. carga).

Um fator importante do presente trabalho foi a utilização de amostras de tráfego real da Web, dando maior veracidade ao estudo realizado. O uso de dados reais com imensas amostras (com milhares e em alguns casos milhões de registros) implicou em complexidade para o desenvolvimento do software para tratamento dos dados e obtenção dos resultados.

A nova abordagem de construção de modelagem aqui proposta contribui claramente em 3 pontos:

1. geração de carga sintética para simulações computacionais envolvendo servidores Web;

2. possibilidade de análise de desempenho do servidor através de cadeias de Markov;

3. pesquisa de novas técnicas relacionadas à melhoria de desempenho de servidores. Por exemplo: algoritmos de descarte de dados em cache.

6.2 Trabalhos futuros

O presente estudo é parte de um projeto maior. Os resultados aqui encontrados serão usados como entrada para um simulador de tráfego, conforme descrito na sessão anterior. A partir dos resultados desta simulação poderão ser realizadas modificações e melhorias na abordagem aqui empregada. Por exemplo:

- Criação de um mecanismo para encontrar o ponto ótimo entre número de tipos de arquivos analisados x maior volume explicado pelo modelo;
- Utilização de amostras coletadas com o protocolo HTTP1.1, que indica exatamente o início e fim da sessão do usuário;
- Estudar e caracterizar o tempo de permanência nos tipos de arquivos, para possibilitar o uso de cadeias de Markov;
- Desenvolvimento de ferramentas de geração de tráfego Web sintético baseado nos parâmetros aqui encontrados;
- Estudo para melhorar a forma de armazenamento de objetos no cache Web atual;
- Identificar a distribuição de densidade de todos os tipos de arquivos que trafegam na Web;
- Através do modelo de tráfego proposto e da geração de tráfego sintético gerado por este modelo, investigar como o conteúdo de um servidor poderia ser adaptado para que o padrão de acesso do usuário, em média, determinasse um tráfego de saída do servidor com correlação de longo termo;
- Sugerir uma recomendação para que os arquivos na Web fossem tratados de forma mais cuidadosa, evitando classificações genéricas, como por exemplo: octet-stream. Foi observado, durante a análise das amostras, que muitas vezes arquivos com tipos conhecidos, como pdf ou txt, são enviados como octet-stream. Esta recomendação indicaria a importância para o tráfego de saída do servidor Web gerado, uma classificação adequada.

Outro aspecto interessante sob o tema de desempenho de servidores Web está no mercado que possui atualmente. Inúmeras empresas oferecem o serviço de modelagem de servidores Web prometendo análise de desempenho e melhoria do mesmo, a fim de prover maior qualidade ao serviço prestado.

7 Referências Bibliográficas

[ALB98] Albitz, Paul & Cricket Liu. Dns and Bind Third Edition, O'Reilly, 1998

[ADA97] Adas, A. Traffic models in broadband network. IEEE Communicationns Magazine, 1997

[ARL97] Arlitt, M. e Williamson, C. Internet Web Servers: Workload Characterization and Performance Implications, IEEE/ACM Transactions on Networking, volume 5 (número 5), 1997

[ARL00] Arlitt, M. e Jin, T. A Workload Characterization Study of the 1998 World Cup Web Site, Hewlett-Packard Labs, 2000

[BAR99] Barret A. Traffic Models for hybrid Satellite-Terrestrial network. Dissertação de mestrado University of Maryland, 1999

[BOX94] G. Box, G. Jenkins e G. Reineel, Time Series Analisys, 3rd edition, New York. Prentice-Hall, 1994

[CAO04] Cao, J., Cleveland, W.S., Gao, Y., Jeffay, K., Smith, F.D. e Weigle, M. Stochastic Models for Generating Synthetic HTTP Source Traffic, IEEE INFOCOM, 2004

[CMP05] Pedroso, C. M., Kotelok, M. e Troian, R. A. S. Um modelo para avaliação de desempenho em servidores Web, X Seminário de Iniciação Científica do CEFET-PR Curitiba de 8 a 10 de agosto de 2005

[CRO95] Crovella, M. e Bestavros, A. Self-similarity in World Wide Web traffic: Evidence and possible causes, IEEE/ACM Transactions on Networking, volume 5 (número 6), 1995

[CRO98] Crovella, M. e Barford, P. Generating Representative Web Workloads for Network And Server Performance Evaluation, Boston University, 1998

[DRA03] Drakakis, K. A detailed mathematical study of several aspects of the Internet, dissertação de mestrado para a universidade de Princeton, 2003

[DUT04] Dutra, Géri Natalino. Caracterização de Parâmetros para Modelos de Serviço HTTP, dissertação de mestrado em Informática UFPR, 2004

[ERA95] Eramilli, A. Singh, R. P. An application of deterministic chaotic maps to model packet traffic, Queueing Systems 20 , páginas 171-206, 1995

[FRE81] Freedman, D. and P. Diaconis. On the histogram as a density estimator. Zeitschrift f¨ur Wahr. und verw.: Theory 57, 453–476, 1981

[FRO94] Frost, V.S. e Melamed, B. Traffic Modeling For Telecommunications Networks, IEEE Communication Magazine, páginas 70-81, 1994

[GON05] Gong, W. B., Liu, Y., Misra, V. e Towsley, D. Self-similarity and Long Range Dependence on the Internet: a second look at the evidence, origins and implications, Computer Networks, 48(3) páginas 377-399, 2005

[HÄR90] Härdle, W. Smoothing Techniques With Implementations in S. Springer-Verlag, 1990

[HWA05] Hwang, F., Bianchi G. R. e Lee, L. L., Impacto Gerado pelo Comportamento das Aplicações (Web, FTP e E-mail) e pelo Perfil das Redes na Característica Auto-Similar, IEEE (Institute of Eletrical and Electronics Engineers), 2005

[KOT05] Kotelok, M., Fonseca, K. e Pedroso, C. M. Um modelo para avaliação de desempenho de servidores Web utilizando classificação de conteúdo. I2TS 4th International Information and Telecommunications Technologies Symposium, 2005

[KRI01] Krishnamurthy, B. & Rexford, J. Web Protocol and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement. 1st Edition, Addison-Wesley, 2001

[LAW99] Law, A. e Kelton, D. Simulation, Modeling and Analysis. McGraw Hill, 3rd Edition, 1999

[LEL94] Leland, W. Taqqu, M. Willinguer, W.e Wilson, D. On the self-similar nature, of Ethernet traffic (extended version), IEEE/ACM (Institute of

Eletrical and Electronics Engineers/Association of Computing Machinery) Transactions on Networking, 1994

[LIN96] Lindsey, J. Parametric Statistical Inference. New York: Oxford Science Publications, 1996

[LIU99] J. Liu, Y.Shu, L. Zhang, F. Sue e O. Yang, Traffic modeling based on FARIMA models. IEEE 1999 Canadian Conference on Eletrical and Computer Engineering, Edmonton, Alberta, Canada, 1999

[MAH97] Mah, B. An Empirical Model of HTTP Network Traffic, IEEE INFOCOM, 1997

[MEN98] Menascé, D. A. e Almeida, V. A .F. Capacity Planning for Web Performane – Metrics, Models and Methods. Prentice Hall, 1998.

[MUS96] Muscariello, L.Markov Models of Internet traffic and a new hierarchical MMPP model, Computer Communications, 1996

[MUS05] Muscariello, L., Melia, M., Meo M., Marsan M. Ajmone, Cigno R. Lo Markov Models of Internet traffic and a new hierarchical MMPP model, Computer Communications, 2005

[NES68] Van Ness, J.W. e Mendelbrot, B.B. "Fractional Brownian Motions, Fractional Noises and Apllications", Computer Communications, vol. 10, páginas 422-437, 1968

[NEV02] Nevil, B. Understanding Internet Traffic Streams: Dragonflies and Tortoises, The University of Auckland, New Zeland, 2002

[NLANR] National Laboratory for Applied Network Research. http://ircache.nlanr.net

[NOR95] Norros, I. On the use of fractional Brownian motion in the theory of connectionless networks. IEEE Journal on Selected Areas in Communications 13, páginas 953- 962, 1995

[NS] http://www.isi.edu/nsnam/ns

[PAR62] Parzen, E. On estimation of a probability density and mode. Annals of Mathematical Statistics 33, 1065–1076, 1962

[PAX95] Paxson, V. e Floyd, S. Wide-Area Traffic: The Failure of Poison Modeling, University of California, Berkeley, 1995

[PAX01] Paxson, V. e Floyd, S. Difficulties in Simulating the Internet, IEEE/ACM Transactions on Network, volume 9, número 4, páginas 392-403, 2001

[R] R Development Core Team, R: A Language and Enrivonment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/

[RAJ91] Jain, R. The Art of Computer Systems Performance Analisys. John Wiley & Sons, 1991

[RFC791] http://rfc.net/rfc791.html

[RFC792] http://rfc.net/rfc792.html

[RFC793] http://rfc.net/rfc793.html

[RFC768] http://rfc.net/rfc768.html

[RFC821] http://rfc.net/rfc821.html

[RFC822] http://rfc.net/rfc822.html

[RFC854] http://rfc.net/rfc854.html

[RFC959] http://rfc.net/rfc959.html

[RFC1942] http://rfc.net/rfc1942.html

[RFC1945] http://rfc.net/rfc1945.html

[RFC2048] http://rfc.net/rfc2048.html

[RFC2158] http://rfc.net/rfc2158.html

[RFC2616] http://rfc.net/rfc2616.html

[RFC3003] http://rfc.net/rfc3003.html

[RFC3057] http://rfc.net/rfc3057.html

[RFC3076] http://rfc.net/rfc3076.html

[ROS56] Rosenblatt, M. Remarks on some non-parametric estimates of a density function. Annals of Mathematical Statistics 27, 642-669, 1956

[STATGRAPHICS] StatGraphics Plus 5.0 – versão Demo limitada

[SCO79] Scoot, D. On optimal and data based histograms. Biometrika 66, 605–610, 1979

[SCO92] Scoot, D. Multivariate Density Estimation: Theory, Practice, and Visualization. Oxford: John Wiley and Sons, 1992

[TAN97] Tanenbaum, Andrew S. Redes de Computadores, 3a edição, Editora Campus, 1997

[TAY98] Taylor, H.M. e Karlin, S. An Introduction to Stochastic Modeling, 3rd edition, Academic Press, 1998

[TEL02] Telek, M. e Horich, A. A Markovian point process exhibiting multi fractal behavior and its aplication to traffic modeling, 4th International Conference on Matrix-Analytic Methods in Stochastic Models, 2002

[VEN01] Vendictis, A. De, Baiocchi, A. Wavelet Based Synthetic Generation of Internet Packet Delays, in: Proceedings of International Teletraffic Conference ITC17, Salvador, Brasil, 2001

[XUE99] F. Xue e T. Lee, Modeling and prediction long-range dependent traffic with FARIMA processes. International Symposium on Communications, Keohsiung, Taiwan, 1999

[SPECWeb96] http://www.spec.org/osg/web96

[WIL00] Willinger, W. e Park, K. Self-similar network traffic and performance evaluation, 1st edition, Academic Press, John Wiley & Sons, 2000.

ANEXOS

A - Resultados das matrizes do número de transição e da probabilidade de transição das amostras 2, 3 e 4 do IRCache.

Matriz do número de transição entre tipos de arquivos - Amostra 2 – IRCache

	HTML	JPEG	GIF	MPEG	PDF	XML	OUTROS	FIM
HTML	1595	853	673	2	3	89	1065	354
JPEG	932	4242	979	1	3	17	755	300
GIF	692	952	2073	2	5	12	649	231
MPEG	4	1	1	0	0	0	5	0
PDF	5	2	5	0	1	0	4	2
XML	72	20	16	0	0	40	62	6
OUTROS	1087	651	625	4	4	48	2280	368
FIM	0	0	0	0	0	0	0	0

Matriz de probabilidade de transição entre tipos de arquivos – Amostra 2 - IRCache

	HTML	JPEG	GIF	MPEG	PDF	XML	OUTROS	FIM
HTML	0,344195	0,184074	0,145231	0,000432	0,000647	0,0192059	0,229823	0,07639
JPEG	0,128925	0,586803	0,135427	0,000138	0,000415	0,0023516	0,10444	0,0415
GIF	0,149913	0,206239	0,44909	0,000433	0,001083	0,0025997	0,140598	0,05004
MPEG	0,363636	0,090909	0,090909	0	0	0	0,454545	0
PDF	0,263158	0,105263	0,263158	0	0,052632	0	0,210526	0,10526
XML	0,333333	0,092593	0,074074	0	0	0,1851852	0,287037	0,02778
OUTROS	0,214525	0,128478	0,123347	0,000789	0,000789	0,0094731	0,44997	0,07263
FIM	0	0	0	0	0	0	0	0

Matriz do número de transição entre tipos de arquivos - Amostra 3 – IRCache

	HTML	JPEG	GIF	MPEG	PDF	XML	OUTROS	FIM
HTML	36956	3558	5489	34	238	224	9656	6854
JPEG	4020	6432	2886	22	62	72	3554	1195
GIF	6249	2989	12477	7	67	97	5051	1549
MPEG	44	7	9	14	0	0	29	14
PDF	257	75	76	1	142	18	198	52
XML	294	71	84	0	22	723	313	88
OUTROS	10386	3556	5114	26	196	299	19542	3237
FIM	0	0	0	0	0	0	0	0

Matriz de probabilidade de transição entre tipos de arquivos – Amostra 3 - IRCache

	HTML	JPEG	GIF	MPEG	PDF	XML	OUTROS	FIM
HTML	0,586519	0,056468	0,087115	0,00054	0,003777	0,003555	0,153248	0,10878
JPEG	0,220358	0,352574	0,158198	0,001206	0,003399	0,0039467	0,194814	0,0655
GIF	0,219371	0,104929	0,438005	0,000246	0,002352	0,0034052	0,177315	0,05438
MPEG	0,376068	0,059829	0,076923	0,119658	0	0	0,247863	0,11966
PDF	0,313797	0,091575	0,092796	0,001221	0,173382	0,021978	0,241758	0,06349
XML	0,184326	0,044514	0,052665	0	0,013793	0,4532915	0,196238	0,05517
OUTROS	0,245207	0,083955	0,120739	0,000614	0,004627	0,0070592	0,461375	0,07642
FIM	0	0	0	0	0	0	0	0

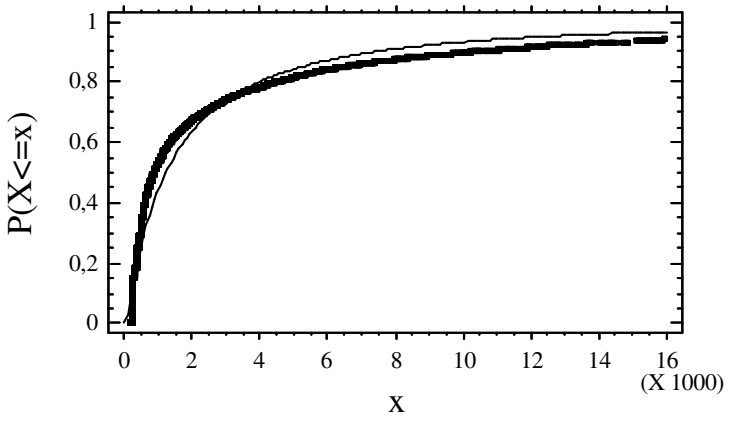
Matriz do número de transição entre tipos de arquivos - Amostra 4 – IRCache

	HTML	JPEG	GIF	MPEG	PDF	XML	OUTROS	FIM
HTML	33911	8495	18536	62	277	380	22525	4894
JPEG	9189	40933	32223	49	73	332	18088	2515
GIF	21095	33303	433547	55	178	1364	83077	5797
MPEG	67	37	40	25	0	1	53	29
PDF	264	74	147	1	217	10	317	51
XML	472	307	1189	1	8	553	1179	105
OUTROS	23834	18387	83291	64	292	1074	175240	5886
FIM	0	0	0	0	0	0	0	0

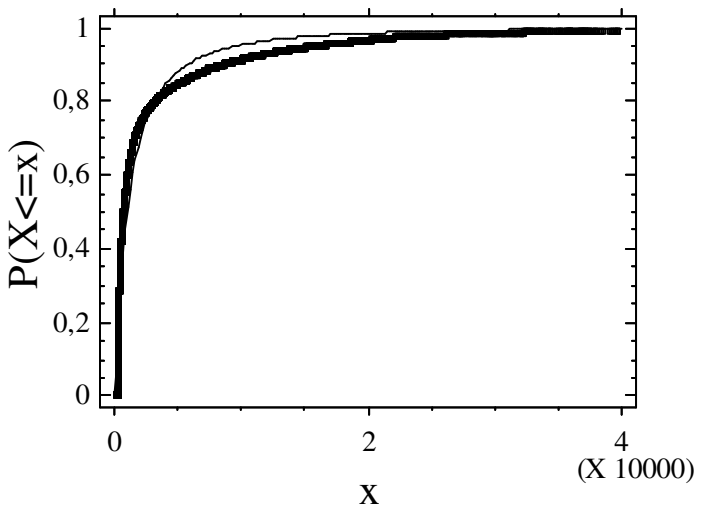
Matriz de probabilidade de transição entre tipos de arquivos – Amostra 4 - IRCache

	HTML	JPEG	GIF	MPEG	PDF	XML	OUTROS	FIM
HTML	0,38068	0,095364	0,208083	0,000696	0,00311	0,0042658	0,252863	0,05494
JPEG	0,088867	0,395863	0,311628	0,000474	0,000706	0,0032108	0,174929	0,02432
GIF	0,03647	0,057576	0,749542	9,51E-05	0,000308	0,0023582	0,143628	0,01002
MPEG	0,265873	0,146825	0,15873	0,099206	0	0,0039683	0,210317	0,11508
PDF	0,244218	0,068455	0,135985	0,000925	0,20074	0,0092507	0,293247	0,04718
XML	0,123755	0,080493	0,311746	0,000262	0,002098	0,1449921	0,309124	0,02753
OUTROS	0,077366	0,059685	0,270366	0,000208	0,000948	0,0034862	0,568835	0,01911
FIM	0	0	0	0	0	0	0	0

B - Caracterização do tamanho dos arquivos transmitidos - IRCache

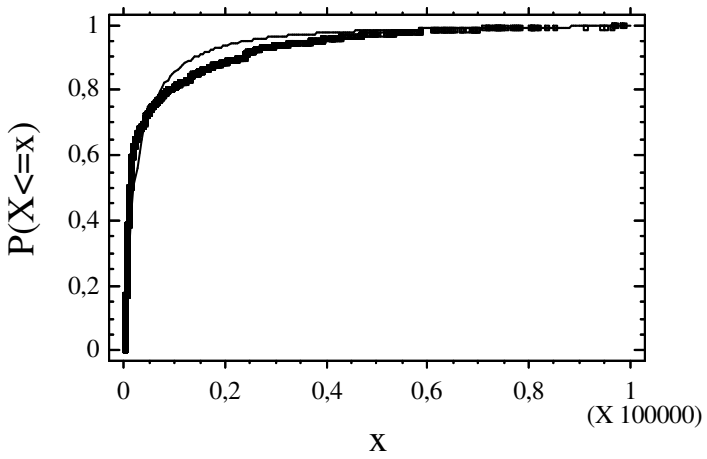


(2)

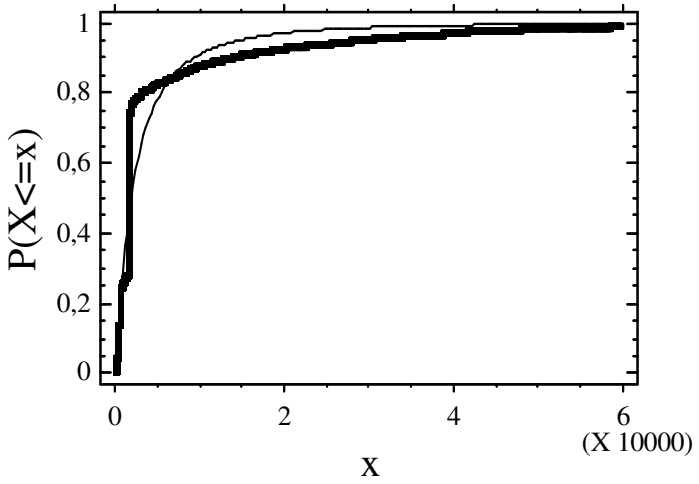


(3)

Gráfico B.1 – Probabilidade acumulada GIF amostras 2 e 3 - IRCache

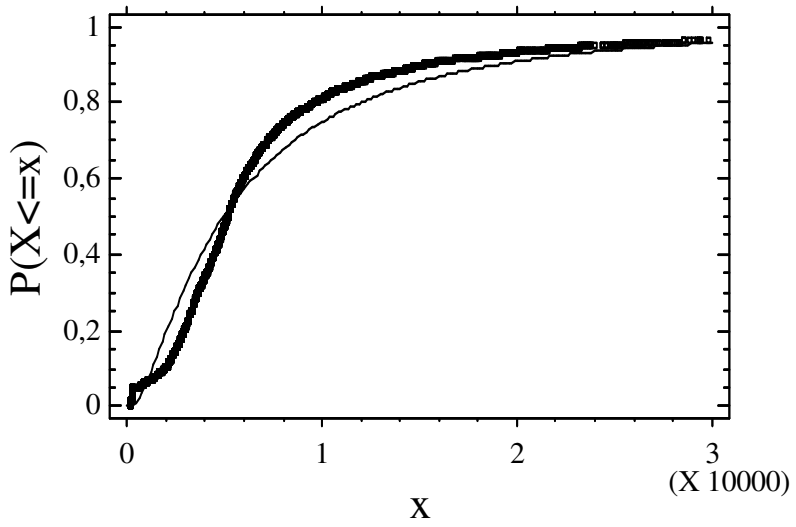


(2)

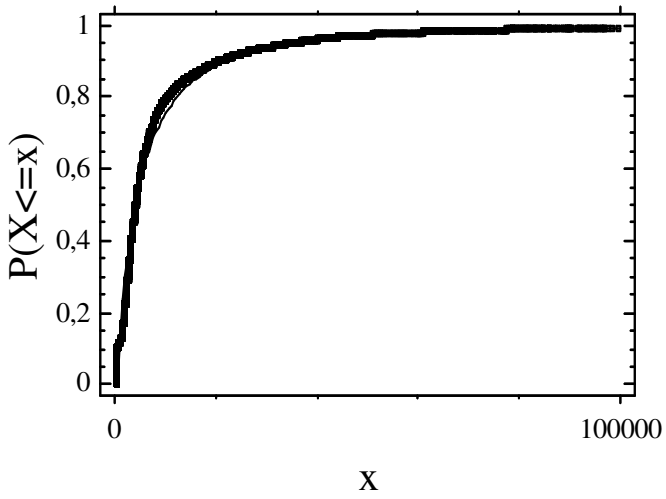


(3)

Gráfico B.2 – Probabilidade acumulada HTML amostras 2 e 3 - IRCache



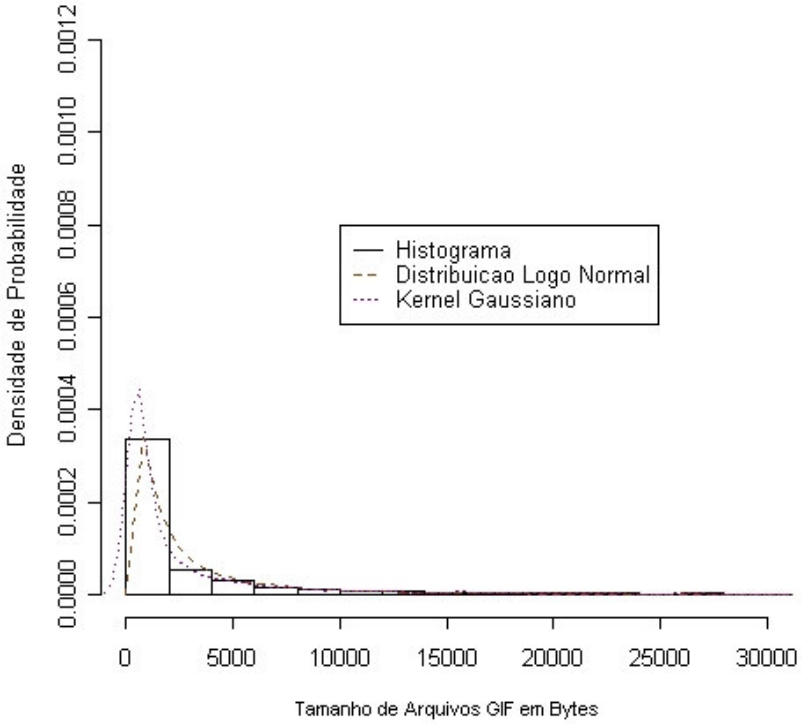
(2)

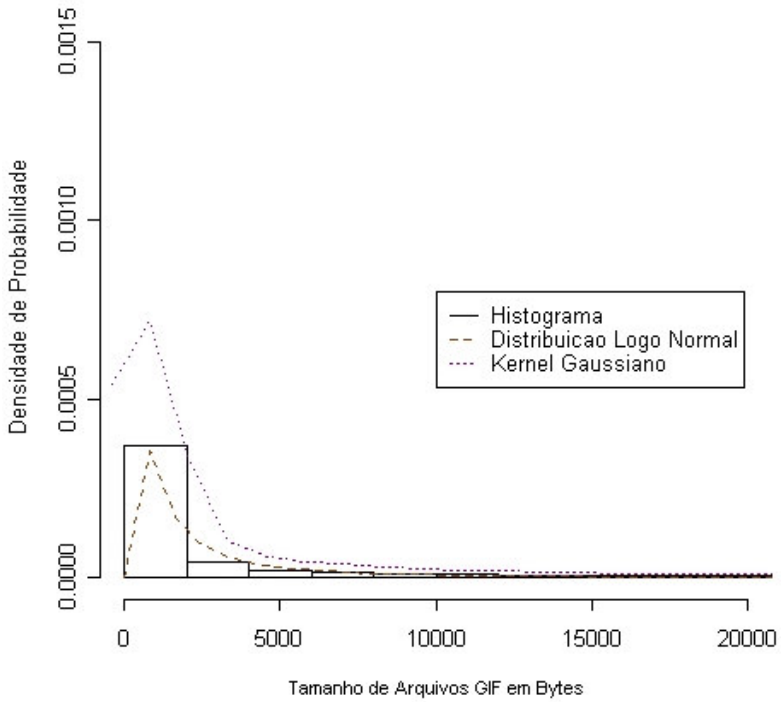


(3)

Gráfico B.3 – Probabilidade acumulada JPEG amostras 2 e 3 - IRCache

C – Ajuste à Distribuição Logo Normal e Kernel Gaussiano para amostras 2, 3 e 4 – IRCache





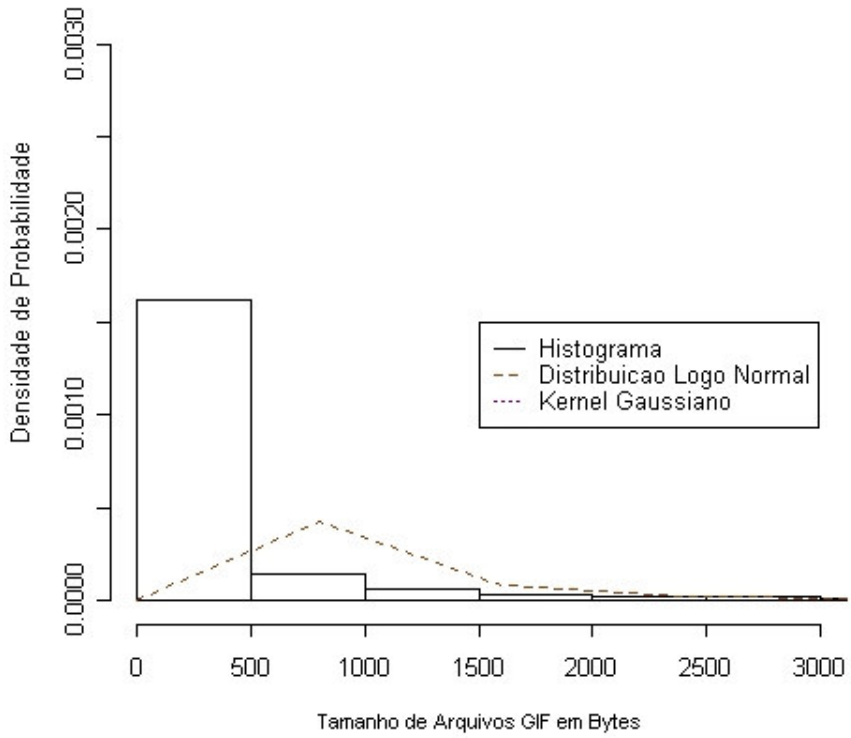
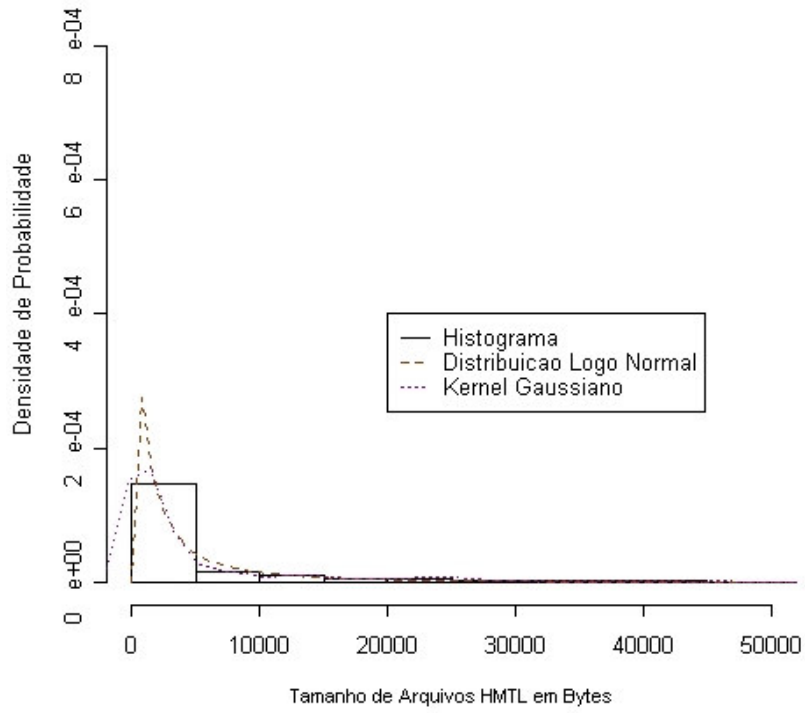
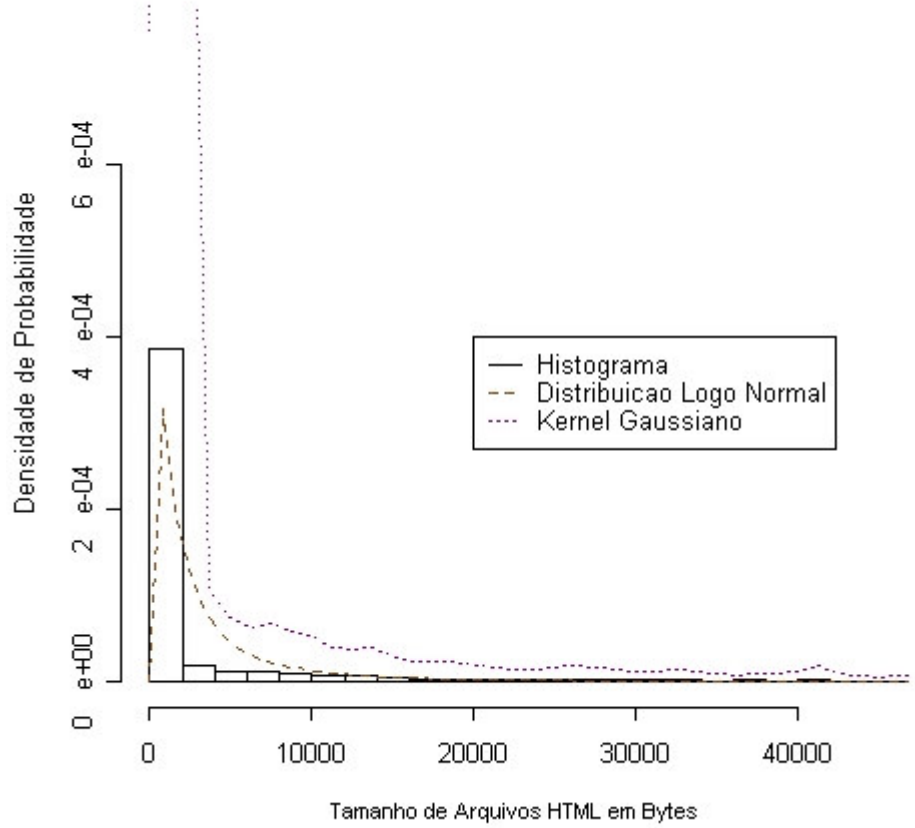


Gráfico C.4 – Ajuste Kernel Gaussiano tamanho de arquivos transmitidos para o tipo GIF das amostras 2, 3 e 4 do IRCache


Densidade de Probabilidade
0 e+00
2 e-04
4 e-04
6 e-04
8 e-04
0
10000
20000
30000
40000
50000
Histograma
Distribuicao Logo Normal
Kernel Gaussiano
Tamanho de Arquivos HMTL em Bytes


Densidade de Probabilidade
0 e+00
2 e-04
4 e-04
6 e-04
0
10000
20000
30000
40000
Histograma
Distribuicao Logo Normal
Kernel Gaussiano
Tamanho de Arquivos HTML em Bytes

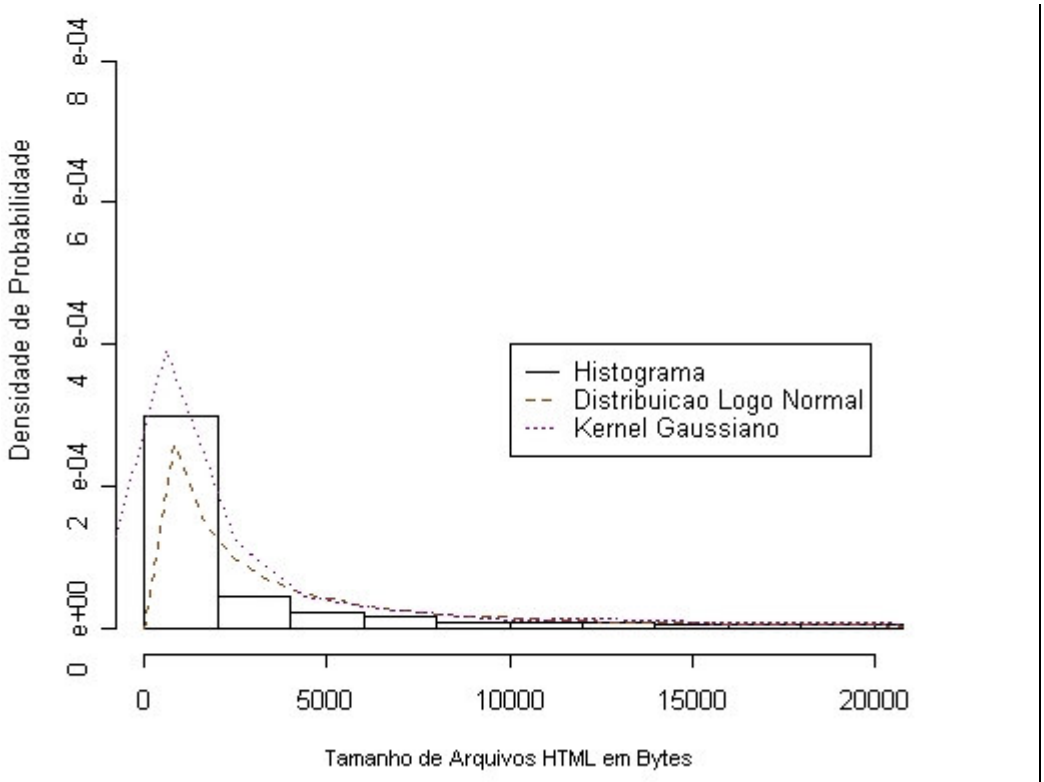
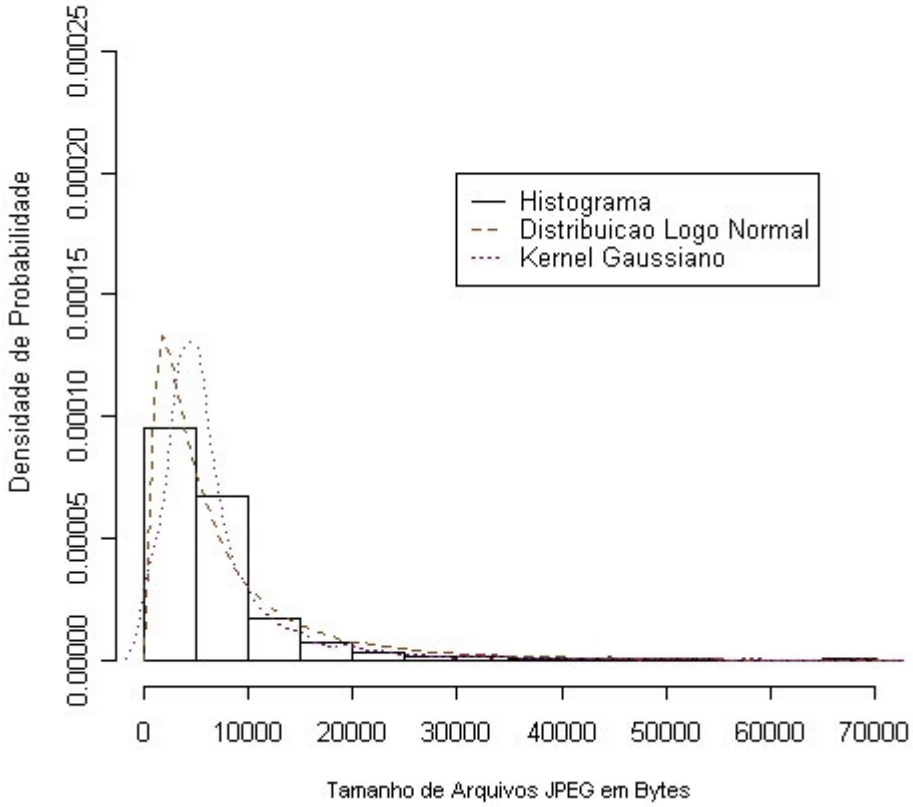
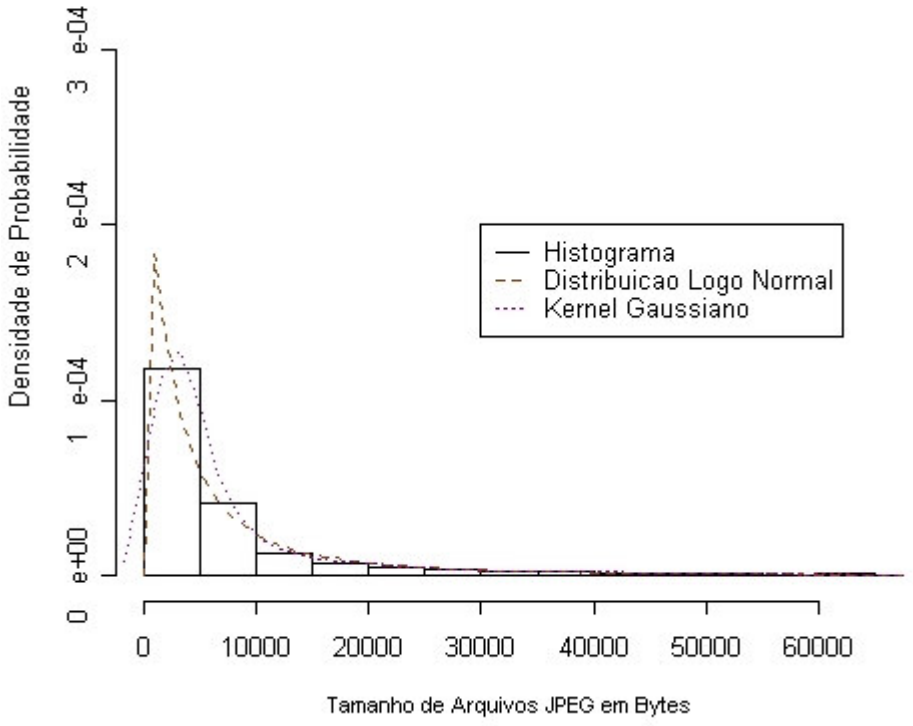


Gráfico C.5 – Ajuste Kernel Gaussiano tamanho de arquivos transmitidos para o tipo HTML das amostras 2, 3 e 4 do IRCache


Densidade de Probabilidade
0.00000
0.00005
0.00010
0.00015
0.00020
0.00025
Histograma
Distribuicao Logo Normal
Kernel Gaussiano
0
10000
20000
30000
40000
50000
60000
70000
Tamanho de Arquivos JPEG em Bytes


Densidade de Probabilidade
0 e+00
1 e-04
2 e-04
3 e-04
Histograma
Distribuicao Logo Normal
Kernel Gaussiano
0
10000
20000
30000
40000
50000
60000
Tamanho de Arquivos JPEG em Bytes

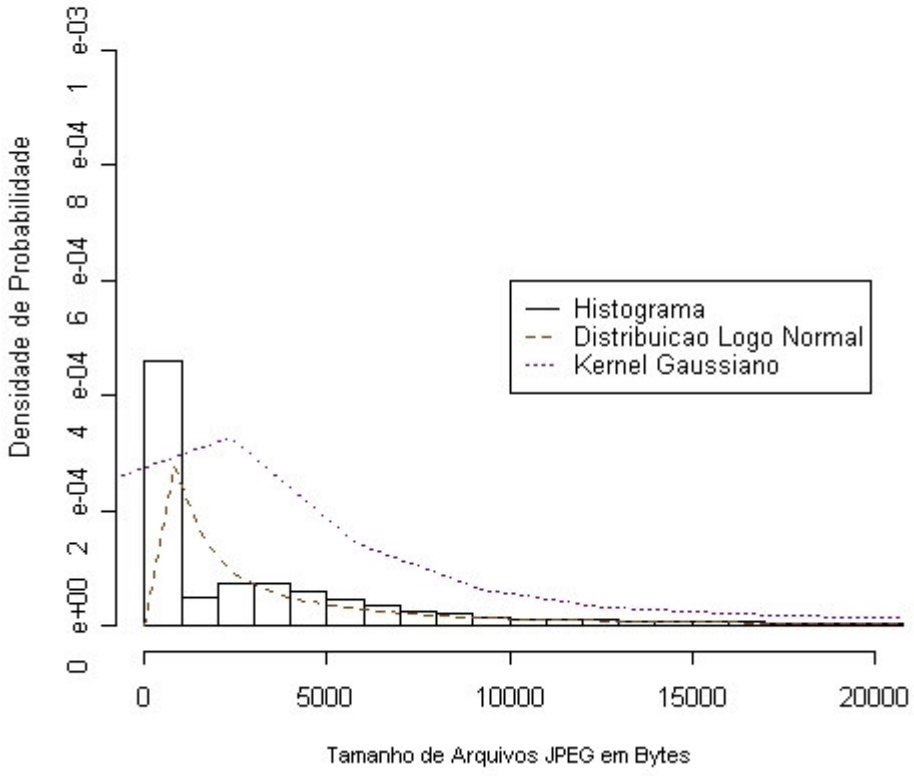


Gráfico C.6 – Ajuste Kernel Gaussiano tamanho de arquivos transmitidos para o tipo JPEG das amostras 2, 3 e 4 do IRCache

D - Protocolo HTTP - códigos de retorno, métodos de pedido, códigos de hierarquia

Existem mais de quarenta códigos de status de resposta no HTTP1.1, que incluem os dezesseis da versão do HTTP1.0 que ainda são significativos para a versão atual [KR01]. Estes estão descritos na [RFC2616]. A seguir listaremos os possíveis códigos de retorno às requisições HTTP:

000 Used mostly with UDP traffic

100 Continue

101 Switching Protocols

*102 Processing

200 OK

201 Created

202 Accepted

203 Non-Authoritative Information

204 No Content

205 Reset Content

206 Partial Content

*207 Multi Status

300 Multiple Choices

301 Moved Permanently

302 Moved Temporarily

303 See Other

304 Not Modified

305 Use Proxy

[307 Temporary Redirect]

400 Bad Request

401 Unauthorized

402 Payment Required

403 Forbidden

404 Not Found

405 Method Not Allowed

406 Not Acceptable

407 Proxy Authentication Required

408 Request Timeout

409 Conflict

410 Gone

411 Length Required

412 Precondition Failed

413 Request Entity Too Large

414 Request URI Too Large

415 Unsupported Media Type

[416 Request Range Not Satisfiable]

[417 Expectation Failed]

*424 Locked

*424 Failed Dependency

*433 Unprocessable Entity

500 Internal Server Error

501 Not Implemented

502 Bad Gateway

503 Service Unavailable

504 Gateway Timeout

505 HTTP Version Not Supported

*507 Insufficient Storage

600 Squid header parsing error

Os códigos de retorno acima fazem parte de cinco grupos, conforme listado abaixo:

- Informação:1xx

- Sucesso:2xx

- Redirecionamento:3xx

- Erro de cliente: 4xx

- Erro de servidor: 5xx

Os métodos de pedido HTTP são usados para informar o que deve ser feito sobre o recurso identificado no endereço solicitado. São apresentados a seguir os métodos possíveis:

Método	Versão disponível	Função
GET	HTTP/0.9	Recupera os objetos
HEAD	HTTP/1.0	Recupera o cabeçalho HTTP
POST	HTTP/1.0	Transfere os dados
PUT	HTTP/1.1	Armazena informações no destino
DELETE	HTTP/1.1	Exclui recursos
TRACE	HTTP/1.1	Permite saber se a mensagem foi entregue
OPTIONS	HTTP/1.1	Permite verificar a localidade de um servidor
CONNECT	HTTP/1.1	Reservado para uso futuro

Os códigos de hierarquia encontrados nas amostras são os que seguem:

NONE - For TCP HIT, TCP failures, cachemgr requests and all UDP requests, there is no hierarchy information.

DIRECT - The object was fetched from the origin server.

SIBLING_HIT - The object was fetched from a sibling cache which replied with UDP_HIT.

PARENT_HIT - The object was requested from a parent cache which replied with UDP_HIT.

DEFAULT_PARENT - No ICP queries were sent. This parent was chosen because it was marked ``default'' in the config file.

SINGLE_PARENT - The object was requested from the only parent appropriate for the given URL.

FIRST_UP_PARENT - The object was fetched from the first parent in the list of parents.

NO_PARENT_DIRECT - The object was fetched from the origin server, because no parents existed for the given URL.

FIRST_PARENT_MISS - The object was fetched from the parent with the fastest (possibly weighted) round trip time.

CLOSEST_PARENT_MISS - This parent was chosen, because it included the lowest RTT measurement to the origin server. See also the *closests-only* peer configuration option.

CLOSEST_PARENT - The parent selection was based on our own RTT measurements.

CLOSEST_DIRECT - Our own RTT measurements returned a shorter time than any parent.

NO_DIRECT_FAIL - The object could not be requested because of a firewall configuration, see also *never_direct* and related material, and no parents were available.

SOURCE_FASTEST - The origin site was chosen, because the source ping arrived fastest.

ROUNDROBIN_PARENT - No ICP replies were received from any parent. The parent was chosen, because it was marked for round robin in the config file and had the lowest usage count.

CACHE_DIGEST_HIT - The peer was chosen, because the cache digest predicted a hit. This option was later replaced in order to distinguish between parents and siblings.

CD_PARENT_HIT - The parent was chosen, because the cache digest predicted a hit.

CD_SIBLING_HIT - The sibling was chosen, because the cache digest predicted a hit.

NO_CACHE_DIGEST_DIRECT - This output seems to be unused?

CARP - The peer was selected by CARP.

ANY_PARENT - part of *src/peer_select.c:hier_strings[]*.

INVALID CODE - part of *src/peer_select.c:hier_strings[]*.

Abaixo seguem os códigos de resultado de como as requisições HTTP foram tratadas, em relação ao Cache Web:

- TCP_HIT --> uma cópia válida do objeto estava no cache
- TCP_MISS --> o objeto requisitado não estava no cache
- DIRECT / NONE --> (Hierarchy Data / Hostname) descrição de como e onde o objeto requisitado foi recuperado.
- TCP_HIT = NONE --> foi recuperado do cache
- TCP_MISS = DIRECT/x.x.x.x --> x.x.x.x é o IP do hostname de origem para o qual o pedido foi direcionado

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo