UNIVERSIDADE DO EXTREMO SUL CATARINENSE

CURSO DE CIÊNCIA DA COMPUTAÇÃO

CAROL PEREIRA GONZAGA

WEBLIMIT: APLICAÇÃO PROXY EXPERIMENTAL PARA CONTROLE DO USO DE BANDA EM REDES DE COMPUTADORES

CRICIÚMA, JULHO DE 2006

CAROL PEREIRA GONZAGA

WEBLIMIT: APLICAÇÃO PROXY EXPERIMENTAL PARA CONTROLE DO USO DE BANDA EM REDES DE COMPUTADORES

Trabalho de Conclusão de Curso para a Obtenção do Grau de Bacharel em Ciência da Computação da Universidade do Extremo Sul Catarinense.

Orientador: Prof. M.Sc. Rogério A. Casagrande

CRICIÚMA, JULHO DE 2006.

CAROL PEREIRA GONZAGA

WebLimit: Aplicação Proxy Experimental para Controle do Uso de Banda em Redes de Computadores

Submetido ao corpo docente do Departamento de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

_____

Profa. M.Sc. Ana Cláudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:

_____

Prof. M.Sc. Rogério A. Casagrande (UNESC)
Orientador

_____

Prof. M.Sc. Paulo João Martins (UNESC)

_____

Prof. Esp. Valter Blauth Junior (SATC)

Dedico este trabalho a minha família.

AGRADECIMENTOS

Agradeço a Deus por minha saúde e por ter condições de realizar todas as atividades que me fazem feliz.

Por minha família, que sempre foi incentivadora e colaborou para que eu pudesse obter o grau de Bacharel em Ciência da Computação.

Amigos e colegas que de uma forma ou de outra colaboraram para que este trabalho fosse desenvolvido.

Professores do curso, em especial ao professor e orientador Rogério Antônio Casagrande, que foi fundamental com suas sugestões consistentes e a professora Merisandra que sempre me guiou para atingir os objetivos.

A força não provém da capacidade física e sim de uma vontade indomável. (Mahatma Ghandi)

RESUMO

Alguns provedores de acesso à Internet estão propondo aos assinantes a cobrança sobre tráfego de dados, que consiste numa cota limite de *download* e *upload*, na qual os usuários que ultrapassarem tal limite deverão pagar cada *megabyte* adicional. Nesta pesquisa foi desenvolvida uma aplicação experimental baseada em *proxy*, chamada *WebLimit*, que realiza a contagem dos *bytes* transmitidos e não permite que a cota seja ultrapassada, já que possibilita o bloqueio de acesso à Internet. Dessa forma, esse trabalho põe em evidência a cobrança sobre tráfego de dados e sugere o controle do uso de banda em uma rede de computadores por meio de uma aplicação *proxy* experimental.

Palavras-Chave: Controle do Uso de Banda, Controle de Tráfego de Dados, Internet, Redes de Computadores, *Proxy*, *WebLimit*.

ABSTRACT

Some providers propose the charge for the data traffic, where the subscribers pay an extra tax when the quota limit for downloading and uploading is exceeded. In this research an experimental tool, called WebLimit, has been developed for the control of data traffic, based on proxy application, which block the access to Internet when the users exceed the quota limit. This work suggests the control and usage of the Internet by the experimental proxy application.

Key-Words: Control of Data Traffic, Internet, Networking, Proxy, *WebLimit*.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTAS DE SIGLAS

3G	Terceira Geração
ADSL	*Asymmetric Digital Subscriber Line*
ANATEL	Agência Nacional de Telecomunicações
ATM	*Asynchronous Transfer Mode*
CGI	*Common Gateway Interface*
CRLF	*Carriage-Return / Line-Feed*
DHCP	*Dynamic Host Configuration Protocol*
DNS	*Domain Name System*
ERB	Estação Rádio Base
EUA	Estados Unidos da América
FCC	*Federal Communications Comission*
FTP	*File Transfer Protocol*
Gb	*Gigabyte*
GPL	*General Public License*
GSM	*Global System for Mobile Communications*
HFC	*Hybrid Fiber-Coaxial Cable*
HTB	*Hierarchical Token Bucket*
HTML	*HyperText Markup Language*
HTTP	*HyperText Transfer Protocol*
IP	*Internet Protocol*
IPv4	*Internet Protocol version 4*
IPv6	*Internet Protocol version 6*
Mb	*Megabyte*

MSN	*Microsoft Network Messenger*
OSI	*Open Systems Interconnection*
P2P	*Peer-to-peer*
PDA	*Personal Digital Assistants*
PDF	*Portable Document Format*
QoS	*Quality of Services*
RNP	Rede Nacional de Pesquisa
SLA	*Service Level Agreement*
SLM	*Service Level Management*
SMTP	*Simple Mail Transfer Protocol*
SSL	*Secure Sockets Layer*
TCP	*Transmission Control Protocol*
TELNET	*Terminal Connector*
UDP	*User Datagram Protocol*
UML	*Unified Modeling Language*
URI	*Uniform Resource Identifier*
URL	*Uniform Resource Local*
USB	*Universal Serial Bus*
VDSL	*Very high-bit-rate Digital Subscriber Line*
WAP	*Wireless Access Protocol*
Wi-Fi	*Wireless-Fidelity*
Wi-Max	*World Wide Interoperability for Microwave Access*
WML	*WAP Markup Language*
WWW	*World Wide Web*

SUMÁRIO

1 INTRODUÇÃO

A evolução tecnológica e a conseqüente diminuição dos custos dos computadores, tornou cada vez mais popular a distribuição do poder computacional numa organização. A necessidade de compartilhamento de recursos de *hardware*, *software* e a troca de informações entre usuários, criaram um ambiente propício para o desenvolvimento das redes de computadores.

Com a popularidade da Internet e o aumento da oferta de tipos de serviços, os usuários tiveram a sua disposição conexões de alta velocidade que a cada dia adquire novos clientes. Freqüentemente são discutidas maneiras de otimizar a utilização dos serviços, suas formas de cobrança e também o desenvolvimento de novas tecnologias para suportar a demanda para os próximos anos.

As companhias de telefonia e provedores de acesso à Internet, possuem a mensalidade, como forma de cobrança mais usual, no entanto, a cobrança por tempo de uso e por quantidade de dados transmitidos também são alternativas que estão em discussão.

A cobrança sobre o tráfego de dados, proposto por algumas empresas provedoras de acesso à Internet, estão causando preocupação nos usuários, já que podem tornar-se inviáveis financeiramente. Porém, há divergência no mercado, pois enquanto algumas companhias telefônicas descartam a possibilidade deste tipo de cobrança, outras têm a cobrança por tráfego descrita em seus contratos de prestação de serviços, mas que ainda não foram postas em prática. A possibilidade deste tipo de cobrança, vir a se espalhar por diversos tipos de acesso à Internet também é preocupante.

Como trabalho proposto foi implementada uma aplicação experimental que limita a transferência de dados entre os computadores de uma rede local e a Internet.

Com isso, empresas poderão obter um controle maior sobre o uso de banda de Internet, podendo evitar gastos desnecessários.

1.1 OBJETIVO GERAL

Desenvolver uma aplicação experimental para controle do uso de banda em uma rede de computadores.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa consistem em:

a) conhecer os tipos de acesso à Internet;
b) compreender os principais protocolos do modelo TCP/IP;
c) compreender *Sockets e Proxies*;
d) conhecer algumas soluções disponíveis para controle de tráfego e
e) desenvolver aplicação experimental para controle do uso de banda em redes de computadores.

1.3 JUSTIFICATIVA

Entre os sistemas operacionais mais populares, estão o *Microsoft Windows* e as distribuições *Linux*, conforme (Info Exame, nº 229, p.79) de abril de 2005. No ambiente *Linux* há uma grande quantidade de *softwares* livres e gratuitos que dispõem

da função de controle de tráfego de dados, o que não ocorre na plataforma *Microsoft Windows*, que possui ferramentas proprietárias e de elevado custo, que podem inviabilizar o planejamento financeiro de uma empresa.

As empresas devem ter um maior controle sobre o tráfego de dados de sua rede, já que companhias telefônicas que fornecem acesso *Asymmetric Digital Subscriber Line* (ADSL), estão propondo meios de cobrarem junto à mensalidade, uma taxa adicional por *megabyte* que o usuário transmitir, caso ultrapasse uma cota limite contratada. Por isso, há a preocupação de que esta taxa possa se estender a outros tipos de acesso à Internet, como rádio, satélite, *cable modem* e surpreendam as empresas que não possuem nenhuma ferramenta para o controlar o tráfego de dados.

Esta pesquisa visa a implementação de uma aplicação gratuita que controle a quantidade de dados transmitidos em uma rede de computadores, de forma que seja definida uma cota limite para a transferência de dados, reduzindo investimento na infra-estrutura de comunicação.

1.4 ESTRUTURA DO TRABALHO

No primeiro capítulo estão descritos: o tema proposto, os objetivos e a justificativa da realização deste trabalho.

No segundo capítulo, são abordados os conceitos fundamentais de Redes de Computadores, Tipos de Acesso à Internet e conteúdo referente a protocolos da arquitetura TCP/IP. Na seqüência são demonstrados o modelo cliente/servidor e o conceito de *sockets* e portas.

No terceiro capítulo, é discutido o conceito do controle de tráfego de dados e a polêmica cobrança sobre esse serviço. Ainda são demonstradas quais as informações

que devem estar descritas nos contratos de prestação de serviços e por fim, a utilização de aplicações *proxies* e *browsers.*

No quarto capítulo, foram citados os trabalhos correlatos, bem como, algumas soluções disponíveis no mercado.

O sistema proposto foi detalhado no quinto capítulo, demonstrando suas funcionalidades, etapas do processo de modelagem, desenvolvimento e testes realizados.

A conclusão contém sugestões para trabalhos futuros, exposição das dificuldades encontradas no decorrer das atividades realizadas e os resultados obtidos.

2 REDES DE COMPUTADORES

Uma rede de computadores é um conjunto de *hosts*[1] interconectados, que possibilitam a troca de informações entre si, seja ela por fio de cobre, cabo coaxial, fibras ópticas, microondas ou ondas de rádio (WIRTH, 2002).

A definição do termo “Redes de Computadores”, pode estar sendo empregado de uma forma desatualizada, pois muitos equipamentos não tradicionais estão conectados na rede. Entre os dispositivos que invadiram o espaço dos computadores de mesa e os servidores estão: *Personal Digital Assistants* (PDAs)[2], TVs, computadores portáteis, telefones celulares, automóveis, equipamentos de sensoriamento ambiental, eletrodomésticos e aparelhos de segurança como câmeras digitais (KUROSE; ROSS, 2005).

Nos anos 80, as redes de computadores eram apenas usadas no mundo acadêmico, hoje encontram-se espalhadas pelo mundo inteiro, tanto para uso corporativo quanto pessoal. É utilizada para diversas tarefas, desde transações bancárias, comércio eletrônico, jogos e até compartilhamento de fotos, músicas, vídeos e outros.

Além disso, com a eficiência das redes possibilitou-se o compartilhamento de recursos como *softwares*, equipamentos e informações entre todos os computadores pertencentes à rede. Um exemplo são as redes P2P[3], que possibilitaram o compartilhamento de recursos entre os computadores dos usuários.

O que distingue uma rede de computadores dos outros tipos de redes, é sua generalidade, já que são montadas a partir de um *hardware* programável de uso geral e

---

[1] *Hosts*: Dispositivos de uma rede. Ex: Computadores, roteadores e outros;
[2] PDA: Computador em dimensões reduzidas, com poder computacional considerável. Ex: *PalmTop* e *Handheld*;
[3] P2P: Redes de compartilhamento entre usuários, *peer-to-peer* (ponto-a-ponto).

não são otimizadas para uma aplicação específica, como somente fazer ligações telefônicas ou enviar sinais de televisão, mas com capacidade de transportar tipos de dados diferentes (PETERSON; BRUCE, 2004).

São várias as vantagens, uma delas é que a rede aumenta a confiabilidade do sistema, pois todos os arquivos de um computador podem ser copiados em outros computadores, garantindo a disponibilidade e a segurança dos dados. A compatibilidade é outra vantagem, que conforme Soares, Lemos e Colcher (1995, p.16), "é a capacidade que a rede possui para se ligar a dispositivos de diversos fabricantes, seja em nível de *hardware* ou *software*".

A desvantagem de estar conectado a uma rede, é a exposição a ataques remotos e a vírus que podem prejudicar o desempenho do computador, ou ainda, pôr em risco a segurança das informações do usuário. Para tentar solucionar estes problemas, foram criados os anti-vírus e várias outras ferramentas que combatem *softwares* maléficos.

A realidade é que as redes se tornaram um meio de comunicação muito eficiente e econômico, pois possibilitaram empresas e pessoas que se localizam a longas distâncias, se comunicarem utilizando *softwares* de mensagens instantâneas, com transmissão de voz e vídeo em tempo real, reduzindo assim os gastos com telefonia.

A tecnologia utilizada na maioria das redes locais é a *Ethernet,*que transmite dados a velocidades que podem chegar a 10 Gbps, se utilizando a *Ethernet* comutada[4], que está em testes e em breve substituirá a *Ethernet* compartilhada, a mais utilizada no momento (KUROSE; ROSS, 2005).

[4] Comutada: Realiza Roteamento de pacotes e balanceamento de carga entre os nós da rede.

Conforme Pesquisa do Ibope/*NetRatings*, o crescimento de acesso a banda larga no Brasil em 2004, cresceu 80% em relação ao ano anterior. São aproximadamente 2,4 milhões de assinantes utilizando Internet de alta velocidade.

Tecnologias que também vêm crescendo muito nos últimos anos, são as redes sem-fio. Tecnologias com *Wireless Fidelity* (Wi-fi)[5], *World Wide Interoperability for Microwave Access* (Wi-Max)[6], *Bluetooth*[7] e outras destacam-se pela mobilidade e flexibilidade. Um problema é quanto a segurança dos dados, já que a transmissão dos mesmos são feitas por ondas de rádio e podem ser interceptadas por espiões.

Segundo Pinheiro (2004c), as tecnologias que sustentarão o cenário nos próximos anos serão: as redes celulares, as redes locais sem-fio, as redes pessoais e as redes corporativas de longa distância. O uso maciço de celulares, com transmissão de voz, dados e vídeo estarão interligados a computadores por meio de tecnologia *Wi-Fi* e *Bluetooth*.

Além disso, o grande investimento na infra-estrutura de telecomunicações possibilitará a utilização e proliferação de redes como *Asynchronous Transfer Mode* (ATM)[8], TV digital e proporcionará uma suposta convergência de tecnologias. Também estarão conectados em redes *Wi-Fi*, as geladeiras, fornos microondas e torradeiras que serão controlados via Internet (PINHEIRO, 2004c).

A inovação e o progresso estão presentes em várias áreas da computação, pois técnicas são otimizadas no desenvolvimento de aplicações, novas formas de distribuição de conteúdo na Internet são pesquisadas e investe-se no melhoramento de equipamentos para o aumento da velocidade da transmissão dos dados, bem como, a

[5] *Wi-Fi*: Protocolo de redes sem-fio para distâncias médias;
[6] *Wi-Max*: Protocolo de redes sem-fio para longas distâncias;
[7] *Bluetooth*: Protocolo de redes sem-fio para curtas distâncias;
[8] ATM: Protocolo de redes de computadores que realiza comutação de pacotes e encapsula os dados em pequenos pacotes de tamanho fixo.

diminuição de custos, importante porque incentiva a proliferação do acesso em banda larga[9] e o acesso sem-fio.

A segurança das redes, está a cada dia sendo levada mais a sério, pois as falhas causam enormes prejuízos, principalmente às organizações que não investem em *firewall*[10], anti-vírus e outras ferramentas de segurança.

Qualquer tipo de acesso à Internet corre o risco de apresentar falhas de segurança.

2.1 TIPOS DE ACESSO À INTERNET

São vários os tipos de acesso à Internet, entre eles *Asymmetric Digital Subscriber Line* (ADSL), *Hybrid Fiber-Coaxial Cable* (HFC), rádio e satélite que diferem principalmente, pelo meio de transmissão utilizado e pela velocidade na transferência dos dados.

Na conexão discada, os *modems* usados para a transmissão de dados alcançam até 56 Kbps sobre uma linha telefônica, utilizam um par metálico de fios de cobre, que devido ao alto custo, baixa velocidade e ocupação do canal telefônico, tem sido substituída por outras modalidades de conexão (TOLEDO, 2001).

Conforme o Comitê Gestor da Internet no Brasil (2005), a ADSL é um serviço de conexão à Internet de alta velocidade, presente em 57,95% das empresas brasileiras. Consiste em uma tecnologia de equipamentos que convertem os sinais da linha telefônica comum, em sinais digitais de alta velocidade. Utiliza a telefonia pública, transmite voz e dados por um par metálico de cobre, que pode atingir uma

[9] Banda Larga: Velocidade de acesso à Internet com largura de banda superior a 200Kbps (conforme FCC);
[10] *Firewall*: *Software* ou *hardware* de segurança que procuram impedir o tráfego de dados com vírus, ataques remotos e outros.

velocidade de até 8 Mbps. Tem canal de transmissão dedicado, ou seja, não permite o compartilhamento da largura de banda e possui conexão permanente durante 24 horas (PINHEIRO, 2004b).

O termo "Assimétrico" da ADSL, indica que a taxa de *download* é diferenciada do *upload*. A velocidade alcançada, depende da distância do modem do usuário em relação ao modem da central telefônica, quanto maior a distância, menor a taxa de transferência de dados, sendo que a distância máxima suportada atinge aproximadamente 6 Km de distância (PINHEIRO, 2004b).

De acordo com Toledo (2001), existem três canais de informação na ADSL:

a) um canal *dowstream* de alta velocidade, na faixa de 50 KHz a 1MHz;

b) um canal duplex de média velocidade, na faixa de 4 KHz a 50 KHz;

c) um canal telefônico normal, na faixa de 0 a 4 KHz.

Outro meio de transmissão de dados é a fibra óptica, que tem a desvantagem do alto custo e dificuldade de implantação, porém possui um alto grau de confiabilidade e segurança, pois é imune a interferências, possui velocidade elevada e baixa atenuação. É aplicada em *backbones*[11], para interligação de Estações de Rádio Base (ERB) na telefonia móvel, em cabos submarinos para transmissão de dados intercontinentais e até mesmo em empresas ou residências para acesso à Internet. Devido a necessidade de alto desempenho nas redes mundiais de comunicação, é inevitável a troca a longo prazo, das redes de cobre pelas ópticas.

Outra alternativa é a transmissão via ondas de rádio, que pode ser instalada em condomínios e ser compartilhada pelos diversos moradores. Necessita de um modem e uma antena de recepção, não requer a contratação de um provedor de conteúdo,

[11] *Backbones:*Espinha dorsal nas redes de telecomunicações, equipamentos de alto desempenho, esquema de ligações centrais de um sistema amplo.

diferente do que ocorre em outros tipos de acesso e a transmissão dos dados pode variar de 300 Kbps a 2 Mbps de velocidade.

A conexão por satélite abrange praticamente todo o território nacional e o sinal é recebido através de uma antena receptora e um transceptor. Como a ADSL, possui diferença nas taxas de *download* e *upload*. Além do alto custo, outro ponto fraco é a vulnerabilidade a tempo ruim, pois a chuva forte pode ocasionar a perda de sinal por alguns instantes. Sua taxa de *download* varia de 200 Kbps a 1Mbps (TOLEDO, 2001).

A tecnologia de *Hybrid Fiber-coaxial Cable* (HFC) ou *Cable Modem,* é transmitida por meio de cabos coaxiais, os mesmos usados nas TVs a cabo, que chegam até um modem que converte os sinais analógicos em digitais, que podem ser conectados no receptor da TV, na entrada *Universal Serial Bus* (USB) do micro ou diretamente na placa-de-rede. A largura de banda é compartilhada, com conexão permanente de 24 horas e o meio de acesso à Internet mais utilizado nos EUA (KUROSE; ROSS, 2005).

A tecnologia sem-fio *Wi-Fi,* mais comum para uso corporativo, funciona com um roteador que emite o sinal através de um *Hot-Spot*[12]. É comum encontrar acesso *Wi-Fi* em aeroportos, hotéis, centros de convenções, redes de cafés e restaurantes. O acesso é permitido por meio de um cartão de recepção, que pode ser acoplado ao *notebook* ou a um *handheld*[13]. Câmeras fotográficas, impressoras e *hubs,* têm saído de fábrica equipadas com essa tecnologia. As barreiras de sinais são problemas enfrentados pelo *Wi-Fi.* Paredes grossas, placas de metais ou qualquer grande bloqueador podem causar falhas na transmissão (WIRTH, 2002).

Existe também a conexão *wireless World Wide Interoperability for Microwave Access* (WiMax*),* criada para cobrir cidades inteiras, com alcance de 50 Km e velocidade de até 70 Mbps.

---

[12] *Hot-Spot:* Dispositivo equipado com antena que transmite ondas de rádio para redes *Wi-fi;*
[13] *Handheld*: Computador reduzido com um pequeno teclado para entrada de dados;

O Brasil, EUA e Europa pesquisam juntos, a transmissão de dados via cabos de energia elétrica, que utiliza o cabeamento existente. Em testes realizados, alcançaram a taxa de 75 Mbps e para os próximos anos, pesquisadores anunciaram que poderá alcançar até 200 Mbps. Bastará ter um computador, um modem e uma tomada especial para o usuário acessar a Internet pela fiação elétrica.

As redes móveis, possibilitaram o acesso à Internet por meio de telefones portáteis, que utilizam tecnologia *Wireless Access Protocol* (WAP), Terceira Geração (3G) ou *i-mode*[14]. A tecnologia WAP[15] é muito utilizada na Europa, é compatível com GSM[16] e trabalha com a arquitetura TCP/IP. Investimentos na tecnologia 3G possibilitaram o alcance de áreas de grande cobertura, com velocidade que podem alcançar 384 Kbps, permitindo a utilização sem problemas de vídeo interativo e voz com qualidade superior ao do telefone fixo (KUROSE; ROSS, 2005).

Para a comunicação ser bem-sucedida, os equipamentos precisam se entender, para isso foram desenhadas várias arquiteturas de protocolos, entre elas se destaca a TCP/IP que é largamente utilizada na Internet.

2.2 ARQUITETURA TCP/IP

A maior parte da comunicação entre computadores da Internet, é baseada na pilha de protocolos TCP/IP, que é caracterizada pela interoperabilidade entre vários sistemas operacionais. É considerada um padrão mundial de comunicação, já que é abrangente a ponto de unir empresas, instituições, escolas, universidades e indivíduos do mundo inteiro.

[14] i-mode: Tecnologia de acesso à Internet por meio de telefone portátil muito utilizada no Japão;
[15] WAP: *Wireless Access Protocol*, tipo de protocolo para acesso a redes sem-fio;
[16] GSM: Sistema de telefonia móvel mais utilizada no mundo.

A arquitetura TCP/IP é um conjunto de protocolos, que consistem em regras que visam dar a base e a estrutura de uma rede. Estão localizados no sistema operacional e podem ser aplicados tanto para comunicação em redes remotas quanto em redes locais (COMER; STEVENS, 1999).

O modelo TCP/IP pode ser comparado ao modelo *Open Systems Interconnection* (OSI), que possui sete camadas (física, enlace, rede, transporte, sessão, apresentação e aplicação). O modelo TCP/IP é composto por 4 camadas (rede, inter-rede, transporte e aplicação).

Cada camada do modelo TCP/IP possui uma função: a comunicação entre os processos ocorre na camada de aplicação; a transferência de dados ocorre na camada de transporte, que tem o *Transmission Control Protocol* (TCP) e o *User Datagram Protocol* (UDP) como protocolos principais; já a camada inter-redes, também conhecida como camada de rede, possui o protocolo *Internet Protocol* (IP), caracterizado por não ser confiável, não possuir controle de fluxo ou recuperação de erros. Por fim, a camada de interface-de-rede, também conhecida como camada de *link*, que faz a interação direta com o *hardware*.

Os protocolos estão estruturados de forma hierárquica e em camadas. A disposição dos protocolos é demonstrada na figura 1.

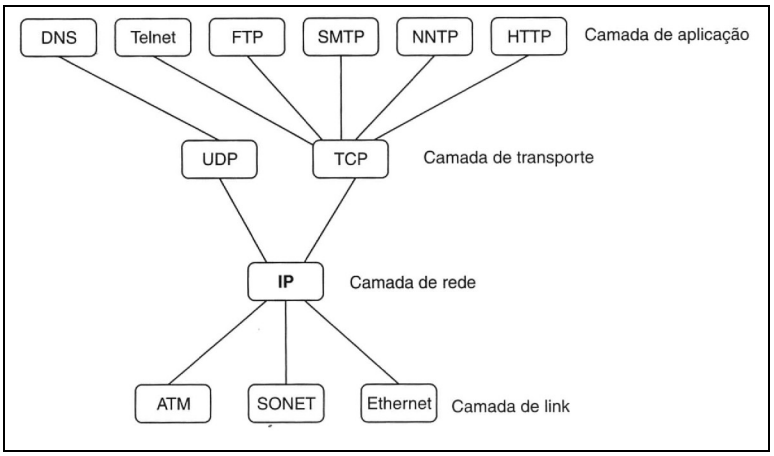

Figura 1. Disposição dos protocolos na Arquitetura TCP/IP.
Fonte: KRISHNAMURTHY, B.; REXFORD, J. (2001, p.126)

Um dos principais protocolos da arquitetura TCP/IP é o IP, que movimenta os datagramas de uma máquina para outra.

2.2.1 *Internet Protocol* (IP)

A Internet é formada por várias redes interligadas com diferentes topologias, com computadores e roteadores formando os *backbones* principais, que por sua vez são conectados aos *backbones* nacionais e estes aos regionais. O que mantém estas redes unidas é o IP, que está localizado na camada de rede. Tanenbaum (2003), descreve que o IP tem a tarefa de fornecer a melhor forma de transportar datagramas da origem para o destino.

Um datagrama IP é formado por cabeçalho e texto, possui uma parte fixa de 20 *bytes* e uma parte opcional de tamanho variável. São vários os campos do cabeçalho de um datagrama. Um tem função para controlar a versão do protocolo, outro o tamanho dos dados, tipo de rede, tempo de vida do pacote, endereço de origem, destino e muitas outras informações. A representação de um cabeçalho de datagrama IP pode ser vista na figura 2.

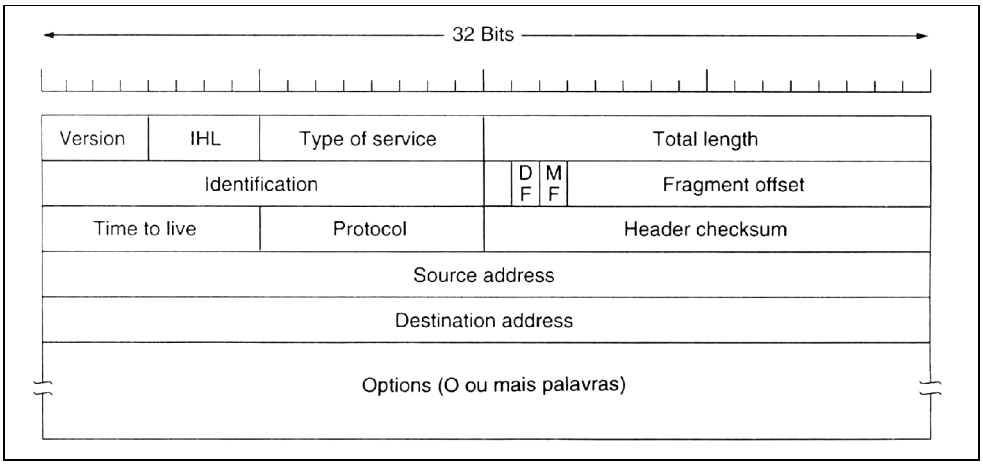

Figura 2. O Cabeçalho *Internet Protocol* (IP)
Fonte: TANENBAUM, A. (2003, p. 461)

Na Internet cada computador possui um endereço IP, que representa seu número de rede e número de identificação de *host.* Duas máquinas nunca têm o mesmo endereço IP.

Os endereços de redes, são divididos nas classes A, B, C, D e E que são representados por 32 *bits*, escritos em notação decimal entre pontos, além de suportar valores de 0 a 255 para cada um dos 4 *bytes* do endereço.

Com o crescimento do número de *hosts* logados na grande rede, surgiram problemas como a insuficiência de endereços IP. Uma solução adotada, foi a utilização de sub-redes com endereços inválidos, ou seja, invisíveis fora da rede local.

Mesmo com o desenvolvimento de técnicas para solucionar os problemas de insuficiência de endereços IP, o *Internet Protocol* versão 4 (IPv4)[17], atualmente o mais utilizado na Internet, está com os dias contados, pois é necessário aperfeiçoar o IP para que se torne mais flexível e eficiente.

Por isso a implantação do IPv6[18], segundo Tanenbaum (2003), propõe:

a) aceitar bilhões de *hosts*, mesmo com alocação de espaço de endereço ineficiente;
b) reduzir o tamanho das tabelas de roteamento;
c) simplificar o protocolo de modo a permitir que os roteadores processem os pacotes com mais rapidez;
d) oferecer mais segurança;
e) dar mais importância ao tipo de serviço, particularmente para os dados em tempo real;
f) permitir *multicast*, possibilitando a especificação de escopos;
g) permitir que um *host* mude de lugar sem precisar mudar o endereço;

[17] IPv4: Protocolo IP versão 4, o mais utilizado atualmente;
[18] IPv6: Protocolo IP versão 6, veio para substituir o IPv4 e já está em testes em muitas redes.

h) permitir que o protocolo evolua no futuro;

i) permitir a coexistência entre o novo e o antigo protocolo durante anos.

Enfim, trabalha-se muito para que a transição e a compatibilidade entre os protocolos IPv4 e IPv6, seja bem-sucedida para que não haja nenhum tipo de perda ou prejuízo na comunicação de dados.

No transporte de mensagens entre aplicações, se destaca o TCP, que é um protocolo confiável por utilizar a confirmação na entrega de pacotes.

2.2.2 *Transmission Control Protocol* (TCP)

Na arquitetura TCP/IP, há dois protocolos principais na camada de transporte, o UDP que é basicamente o IP com um pequeno cabeçalho e o TCP que é um protocolo orientado para conexão e com controle de fluxo. O TCP foi projetado especificamente para oferecer um serviço de transporte confiável, com fluxo de *bytes* fim-a-fim numa inter-rede não-confiável. Segundo Comer e Stevens (1999), como a maioria dos protocolos de transporte confiáveis, o TCP utiliza *timeout* com retransmissão para alcançar confiabilidade.

A camada IP não oferece controle de fluxo, por isso os datagramas podem sofrer atrasos, serem duplicados, perdidos, entregues fora de ordem, danificados ou truncados.

> O propósito inicial do TCP é fornecer um circuito lógico ou serviço de conexão confiável entre pares de processos. Como o TCP não conta com confiabilidade dos protocolos de níveis inferiores como o IP, ele deve garantir isto por si mesmo. (MURHAMMER, 2000, p.76).

O TCP pode estar localizado num processo do usuário ou numa parte do *kernel* do sistema operacional, que gerencia fluxos e interfaces TCP para a camada IP.

Vale ressaltar que uma entidade TCP é diferente de um protocolo TCP, pois o primeiro é um *software* que processa os datagramas, enquanto o segundo é um conjunto de regras (TANENBAUM, 2003).

Para uma conexão ser estabelecida é necessário que haja um *socket*[19], ou seja, a comunicação entre terminais receptor e transmissor. Após estabelecida a conexão, poderão ser feitas as trocas de informações do tráfego TCP, que é realizado entre dois terminais numa transmissão simultânea para ambas as direções, ou seja, ponto-a-ponto e *full-duplex* respectivamente.

A maioria dos protocolos de aplicativos, como o *Terminal Connector* (TELNET) e o *File Transfer Protocol* (FTP)[20], usam o TCP como meio de transporte. Além disso, o TCP é compatível com os processos de *multicast*, isto é, distribuem sinais somente para alguns *hosts* e difusão para todos, também conhecida como *broadcast*.

O protocolo que fornece serviços diferenciados do TCP, é o UDP, que possui maior velocidade no envio de pacotes, porém não é confiável, pois em redes de longa distância possibilita a perda de pacotes.

2.2.3 *User Datagram Protocol* (UDP)

O UDP é um protocolo de transporte simples, que tem a função de estender o serviço de entrega *host*-a-*host* numa comunicação entre processos. Oferece uma forma dos programas enviarem datagramas, sem estabelecer uma conexão (PETERSON; BRUCE, 2004).

O UDP é um protocolo não-confiável, que se localiza logo acima do protocolo IP e está no mesmo nível do TCP na camada de protocolos. Caracterizam-se por não usar confirmação no recebimento de pacotes, nem ordenar as mensagens ou

[19] *Socket:* Abstração utilizada na comunicação entre processos, união de IP e Porta;
[20] FTP: Protocolo utilizado para transferência de dados.

fornecer informação para controle da velocidade com que os segmentos fluem (COMER, 1998).

Algumas características no protocolo de transporte UDP, são importantes para o sucesso das aplicações de telefonia por Internet e videoconferência, pois estas toleram uma pequena quantidade de perdas de pacotes. *Buffers*[21] de envio e recebimento de dados, parâmetros de congestionamento e outros, não possuem estado de conexão, no entanto, possuem uma pequena sobrecarga, pois são apenas 8 *bytes* de cabeçalho nos pacotes, deixando assim o envio imediato e mais rápido (KUROSE; ROSS, 2005).

Na tabela 1, são demonstradas algumas aplicações e seus respectivos protocolos de transporte.

Tabela 1. Aplicações populares na Internet e protocolos de transporte subjacentes

Aplicação	Protocolo de camada de aplicação	Protocolo de transporte subjacente
Correio eletrônico	SMTP	TCP
Acesso a Terminal Remoto	Telnet	TCP
Web	HTTP	TCP
Transferência de arquivo	FTP	TCP
Recepção de multimídia	Tipicamente proprietária	Tipicamente UDP
Telefonia por Internet	Tipicamente proprietária	Tipicamente UDP
Gerenciamento de rede	SNMP	Tipicamente UDP
Tradução de nomes	DNS	Tipicamente UDP

Fonte: Adaptado de KUROSE, J. F.; ROSS K. W. (2005, p.155)

O IP é o protocolo responsável por transportar as mensagens UDP. Para diferenciar os programas que estão sendo executados, o UDP utiliza portas[22] para controle da comunicação entre processos. No datagrama, juntamente com o cabeçalho do pacote, estão descritas as portas de origem e destino dos pacotes.

O formato das mensagens UDP consiste em duas partes:

a) cabeçalho UDP;

b) área de dados UDP.

[21] *Buffers:* Filas de armazenamento;
[22] Portas: Identificação de processos num sistema operacional.

O cabeçalho é dividido em 4 campos de 16 bits, que especificam a porta de origem e destino, o comprimento da mensagem e a soma de verificação. A figura 3 representa um datagrama UDP.

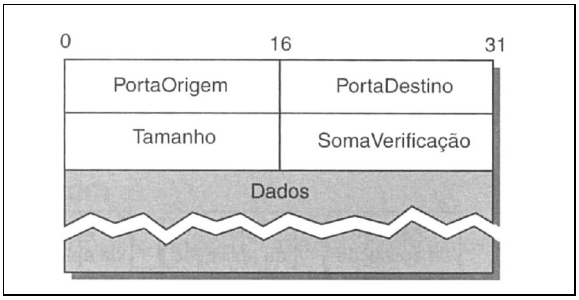

Figura 3. Formato do *User Datagram Protocol* (UDP)
Fonte: PETERSON L.; BRUCE D. (2004, p.273)

Quando a mensagem é transportada entre as camadas, são incluídos vários valores aos cabeçalhos, que são retirados na passagem das mensagens pelas camadas da aplicação destino.

O UDP trabalha com filas para o acúmulo de processos, pois quando as mensagens chegam, são imediatamente enviadas para o final da fila. Se a fila ficar cheia, os novos processos são descartados e caso a fila esteja vazia, o processo de leitura UDP é bloqueado, até uma nova mensagem solicitá-la novamente.

Programas que utilizam o protocolo UDP devem responsabilizar-se pela confiabilidade dos dados transmitidos, já que o protocolo não faz este tipo de serviço. O UDP geralmente não é utilizado no transporte de dados importantes, pois não verifica se o pacote de dados chegou a seu destino.

Por fim, o UDP se torna viável quando utiliza redes locais confiáveis, pois nestas quase não ocorrem perdas de pacotes, entretanto, em redes maiores a taxa de perda é grande e por isso não se deve utilizá-lo para a comunicação entre aplicações. Na prática é utilizado em protocolos como o DNS e o *Dynamic Host Configuration Protocol* (DHCP) (TORRES, 2001).

Alguns protocolos em nível de aplicação, como o *HyperText Transfer Protocol* (HTTP), não utilizam o UDP por necessitarem de serviços confiáveis.

2.2.4 *HyperText Transfer Protocol* (HTTP)

O HTTP é um protocolo de pedido e resposta, que oferece suporte a *World Wide Web*, se localiza na camada de aplicação da arquitetura TCP/IP e é conhecido como porta 80 nos sistemas operacionais. Foi projetado para permitir a transferência de documentos HTML. E é um protocolo sem estado, pois não guarda informações sobre as conexões (KRISHNAMURTHY; REXFORD, 2001).

Por não guardar informações de conexões, surgiram alguns problemas. No *e-commerce*[23] por exemplo, uma transação pode ser comprometida por não haver dados da confirmação de uma venda. Para solucionar estes problemas surgiram os *cookies*[24].

Para carregar uma página com duas figuras, um navegador irá abrir três conexões TCP, uma para a página e duas para as figuras. A maioria dos navegadores são capazes de lidar com várias dessas conexões simultaneamente.

A comunicação usando o protocolo HTTP é bidirecional, permitindo que dois recursos se comuniquem ao mesmo tempo, o que acaba diminuindo o tráfego da rede e aumentando o desempenho da mesma.(TORRES, 2001).

Conforme Murhammer et al (2000, p.424), o HTTP:

> Se baseia em uma atividade de requisição-resposta. Um cliente, executando um navegador estabelece uma conexão com um servidor na forma de um método de requisição. O servidor responde com uma linha de status, incluindo a versão do protocolo da mensagem e um código de sucesso ou erros seguidos por uma mensagem contendo informações sobre a entidade e um possível conteúdo.

---

[23] *E-commerce:* Comércio Eletrônico;
[24] *Cookies:* Informações do usuário que são trocadas entre o servidor e o *browser* em uma conexão HTTP.

Quando o usuário entra com uma *Uniform Resource Locator* (URL) no navegador, este faz uma conexão com o servidor remoto, que é denominado por tal URL. Após estabelecer a conexão, geralmente abre uma página com texto, imagens e sons, que são rodeados de hipertextos. Estes hipertextos, são *links* ou atalhos para novas páginas ou para outras máquinas remotas localizadas na rede mundial de computadores.

Conforme Peterson e Bruce (2004), o formato de uma mensagem HTTP apanhada pelo *browser* é representada na figura 4.

```
Linha_Inicial <CRLF>
Cabeçalho_Mensagem <CRLF>
<CRLF>
Corpo_mensagem <CRLF>
```

Figura 4. Formato de uma mensagem *HyperText Transfer Protocol* (HTTP)
Fonte: PETERSON L.; BRUCE D. (2004, p.475)

A primeira linha identifica se a mensagem é uma solicitação ou uma resposta, logo após é descrito o *carriage-return/line-feed* (CRLF), ou seja, retorno de carro/ linha nova. Ainda na mesma figura, logo abaixo, vêm as opções e os parâmetros utilizados na mensagem (PETERSON; BRUCE, 2004).

O sucesso ou falha de uma mensagem de resposta consiste num código de resposta numérico. Nas mensagens de solicitação, diversas operações são utilizadas, algumas delas são demonstradas na Tabela 2.

Tabela 2. Operações de solicitação *HyperText Transfer Protocol* (HTTP)

Operação	Descrição
Options	Solicita informações sobre opções disponíveis
Get	Apanha documento identificado no URL
Head	Apanha metainformações sobre documento identificado no URL
Post	Dá informações (por exemplo, anotação) ao servidor
Put	Armazena documento sob URL especificado
Delete	Exclui URL especificado
Trace	Mensagem de solicitação de *loopback*
Connect	Usada por *proxies*

Fonte: PETERSON L.; BRUCE D. (2004, P. 477).

Como exemplo, pode-se demonstrar: *“ GET HTTP:\\www.unesc.net/index . html HTTP / 1.1”*. Indica ao servidor *“www.unesc.net”*, que mostre a página *index.html*, utilizando a versão 1.1 do protocolo HTTP.

Já nas mensagens de resposta, o servidor envia códigos de requisições, que podem ou não ser bem-sucedidas. Por exemplo, *“HTTP / 1.1 404 Not Found”*, significa que a solicitação contém sintaxe errada ou não pode ser atendida.

Alguns tipos de mensagens podem ser vistas na tabela 3.

Tabela 3. Cinco tipos de códigos de resultado HTTP

Código	Tipo	Exemplos de motivos
1xx	Informação	Solicitação recebida, continuando a processar
2xx	Sucesso	Ação recebida com sucesso, entendida e aceita
3xx	Redirecionamento	Outra ação precisa ser tomada para completar a solicitação
4xx	Erro do cliente	Solicitação contém sintaxe errada ou não pode ser atendida
5xx	Erro do servidor	Servidor falhou ao realizar uma solicitação aparentemente válida

Fonte: PETERSON L.; BRUCE D.(2004, p. 478).

As operações de pedido, alterações, criação ou exclusão de recursos, são feitas por meio de métodos de pedidos contidos no protocolo.

O protocolo HTTP, depende muito da estrutura de *Uniform Resource Identifier* (URI), que é um mecanismo que localiza os endereços dos mais diferentes recursos da Internet, sejam eles objetos de dados ou serviços da rede.

O HTTP também pode ser considerado um esquema, que designa qual protocolo deve ser usado para acessar um determinado recurso, por meio dos mecanismos de nomeação e de sua própria sintaxe (KRISHNAMURTHY; REXFORD, 2001).

Existem recursos chamados metadados, que estão inclusos em grande parte dos pedidos e respostas HTTP. Eles trazem informações relacionadas a cada recurso, como o tamanho do conteúdo de um pacote, a hora que foi realizada a última

modificação, ou ainda, fornece dados sobre o formato de codificação da entidade para ajudar no seu processamento.

O *Common Gateway Interface* (CGI), é outro recurso importante do HTTP, pois os programas são executados no servidor, diferente dos *applets Java*[25] que executam no cliente. (TORRES, 2001).

Na comunicação, os aplicativos obedecem a estruturas chamadas modelo cliente/servidor, na qual uma aplicação fornece os dados, enquanto outras os utilizam.

2.3 MODELO CLIENTE/SERVIDOR

O padrão principal de interação entre os aplicativos, é conhecido como cliente/servidor.

O servidor é um programa que recebe uma solicitação, executa e envia os resultados em uma resposta. Um servidor pode geralmente tratar de várias solicitações ao mesmo tempo.

Também são exemplos de aplicações servidoras, um servidor de hora do dia, que simplesmente retorna a hora corrente sempre que um cliente enviar um pacote solicitando a informação. Ou ainda, um servidor de arquivos, que recebe solicitações para executar operações que armazenam ou recuperam dados de um arquivo.

Conforme Murhammer (2000, p.141) :

> Um servidor é um aplicativo que fornece um serviço para os usuários da Internet; um cliente é um solicitador de um serviço. Um aplicativo consiste de ambos, servidor e cliente, que podem ser executados no mesmo sistema ou em sistemas diferentes.

Geralmente os usuários solicitam a parte cliente do aplicativo, a qual constrói uma solicitação para um serviço em particular e o envia ao servidor do

[25] *Applet Java*: Recurso da linguagem *Java* que executa a aplicação na máquina cliente.

aplicativo, usando TCP/IP como veículo de transporte. Os *browsers* são exemplos bem populares de aplicativos clientes para a *web*, que requisitam pedidos aos servidores.

No servidor, é obrigatório a identificação de portas bem-conhecidas[26], já no cliente, qualquer porta disponível pode ser reservada arbitrariamente, com mecanismos de mapeamento de porta e outras técnicas.

Uma máquina pode rodar várias aplicações servidoras, podendo juntamente com estas, compartilhar informações por meio da comunicação TCP/IP. Em alguns casos, pode-se colocar as aplicações servidoras em máquinas independentes, para aumentar a confiabilidade e melhorar o desempenho dos programas, já que assim, as chances de ocorrerem falhas diminuem.

Dentre as partes de uma aplicação servidora estão: um programa mestre simples, responsável por aceitar novas solicitações e um conjunto de escravos para tratamento de solicitações individuais ou isoladas.

O mestre tem a função de abrir e esperar na porta, onde chegam as solicitações. Também pode escalar um escravo quando há processamento de solicitações paralelas. O escravo simplesmente executa tal solicitação e a conclui. Como o processamento ocorre paralelamente, a solicitação que requerer menos tempo, terminará mais cedo, independentemente da ordem de chegada.(COMER, 1998).

Para controlar essas várias solicitações, são utilizadas regras e políticas de proteção, para evitar a sobrecarga no sistema e até mesmo os travamentos. O sistema operacional põe uma prioridade mais alta para os aplicativos servidores, já que estes precisam ler arquivos do sistema, manter registros e acessar dados protegidos.

Em alguns casos, as aplicações servidoras podem se comportar como um cliente, que busca informações em outros aplicativos ou no próprio sistema operacional.

[26] Portas bem-conhecidas: Números de portas entre 0 e 1023.

Para evitar congestionamento, pode-se fazer uma cópia prévia dessas supostas informações e disponibilizá-las em *cache* para haver ganho de velocidade.

Na comunicação entre os processos se destacam os *sockets,* que são abstrações utilizadas na troca de mensagens.

2.4 *SOCKETS*

Sockets são abstrações utilizadas na comunicação entre computadores de uma rede, composta por servidor e cliente, onde cada máquina gera seu *socket,* sendo o servidor o responsável por "escutar" ou "esperar" por uma conexão do cliente, enquanto este dispara um *socket* requisitando algum recurso. (TANENBAUM, 2003).

Para um *socket* ser gerado, o servidor deve fornecer um serviço, aguardar a requisição dos clientes, que por sua vez precisam do endereço IP e da porta do servidor.

Um *socket* pode ser utilizado por várias conexões ao mesmo tempo. Em outras palavras, duas ou mais conexões podem terminar no mesmo *socket*. Para identificar cada *socket,* usa-se a denominação *socket1, socket2* e assim sucessivamente.

Como o protocolo TCP é orientado para conexão, exige que ambos os pontos terminais concordem em trocar informações. Para isso, o sistema operacional de origem, atribui uma porta à sua extremidade destino como *open passivo*[27] e quando o pacote for enviado, o aplicativo da outra extremidade solicitará ao sistema operacional, que estabeleça um *open ativo*[28] para efetuar a conexão.

Como o *Socket* é formado por um endereço IP e uma porta, é necessário o conhecimento de portas num sistema operacional.

---

[27] Open passivo: ocorre quando o *socket* espera por uma conexão;
[28] Open ativo: ocorre quando é disparado um *socket* para conexão.

2.4.1 Portas

O modelo TCP/IP utiliza portas para identificar o destino final de cada pacote. Conforme Comer e Stevens (1999) "...a abstração porta proporciona um ponto de encontro por meio da qual processos podem passar dados. Imaginamos uma porta como uma fila de mensagens finita acompanhada por dois semáforos que controlam o acesso". A tabela 4 apresenta alguns protocolos e suas respectivas portas.

Tabela 4. Alguns exemplos de portas bem-conhecidas

Identificação	Palavra-chave	Descrição
21	FTP	*File Transfer Protocol*
23	TELNET	*Terminal Connector*
25	SMTP	*Simple Mail Transfer Protocol*
53	DNS	*Domain Name System*
69	TFTP	*Trivial File Transfer Protocol*
80	HTTP	*HyperText Transfer Protocol*
123	NTP	*Network Time Protocol*

Fonte: Adaptado de COMER, D. (1998, p.245).

Em geral as portas possuem *buffers*, para que os dados sejam armazenados até que um processo esteja pronto para aceitá-los.

A atribuição das portas, pode ocorrer de forma dinâmica, sendo o *software* o responsável por questionar o sistema operacional, qual porta utilizar, ou então, adotar uma abordagem híbrida, com algumas portas definidas e outras escolhidas dinamicamente. Na prática, antes de enviar um datagrama, cada programa negocia com o sistema operacional, uma porta de protocolo.

Cada servidor possui uma lista com valores de portas fixas: 53 para *Domain Name System* (DNS); 25 para correio eletrônico e assim por diante. Pode acontecer de o servidor e o cliente usarem portas bem-conhecidas, depois mudarem para portas não tão

comum e assim liberar as bem-conhecidas para outros clientes (PETERSON; BRUCE, 2004).

Outra alternativa, seria o mapeamento de porta, na qual o cliente envia uma mensagem com o objetivo de perguntar qual porta deverá ser utilizada para tal serviço e o servidor retorna o número da porta a ser utilizada (PETERSON; BRUCE, 2004).

O conceito de portas utilizado no TCP, é muito mais complexo que o UDP, pois trabalha com conexões e não com portas isoladas. No TCP, a conexão ocorre através de um par de pontos terminais, que são definidos por um endereço IP de 32 *bits* e uma porta de 16 *bits*, para identificação em cada *host*. Juntos, o endereço IP e a porta TCP, especificam uma conexão de Internet ou um *socket* (COMER,1998).

Assim, pode-se representar (128.10.0.36, 1069) e (128.10.2.3, 25), como uma conexão entre uma máquina com endereço IP igual a (128.10.0.36) e porta 1069, está recebendo informações da máquina-origem de endereço IP (128.10.2.3) e porta 25.

As conexões podem ocorrer de forma exclusiva entre dois ou mais computadores. Não há nenhum tipo de conflito nesses casos, pois o controle é feito pela associação das mensagens com número da conexão.

Portanto, o programador que conhece o funcionamento de portas, pode desenvolver um sistema que forneça serviços simultâneos, sem se preocupar em definir portas exclusivas para cada conexão.

Todo o tráfego de dados ocorre entre as camadas da arquitetura TCP/IP, por meio dos datagramas e demais procedimentos. O controle desse tráfego, pode ser efetuado por meio de aplicações desenvolvidas especialmente para esse fim.

3 CONTROLE DE TRÁFEGO DE DADOS

A medição do tráfego em uma rede de transmissão de dados, possibilita a obtenção de informações sobre a largura de banda, análise do tipo de dados, filtragem de pacotes e até mesmo estatísticas dos recursos mais utilizados na transferência de dados.

Pode-se controlar a quantidade de dados trafegados em determinado período de tempo, medindo a taxa de dados transmitidos, com o objetivo de analisar e monitorar os *downloads* e *uploads* da rede. Nestes casos, é feita análise dos tipos de serviços, velocidade de transmissão de arquivos, horários de maior utilização da rede e outros.

As medições de transferências de arquivos na *Web* têm sido importantes, para identificação das principais características do tráfego na rede e avaliação de novas técnicas para a melhoria do desempenho da *Web* (KRISHNAMURTHY; REXFORD, 2001).

Em *sites* de *e-commerce*, seria interessante saber qual o *link* mais acessado pelos usuários. Ou ainda, quando um *site* de uma empresa leva muito tempo para ser carregado, é necessário melhorar seu desempenho, colocando-se menos imagens e animações para diminuir o tempo de carregamento do *site*. Para empresas de hospedagem de *sites*, poderia-se diagnosticar quais os recursos mais utilizados, para desta forma disponibilizar somente os que são necessários, diminuindo o excesso de recursos e a sobrecarga.

De acordo com Krishnamurthy e Rexford (2001, p.351), “A medição da latência[29] percebida pelo usuário também pode ser útil no julgamento do desempenho do servidor da *Web* que hospeda o *site*”.

[29] Latência: Intervalo de tempo entre uma solicitação e uma resposta;

As empresas que possuem grande tráfego de dados, podem instalar um servidor *proxy*[30] com *caching* para economizar largura de banda, isso porque grande parte dos dados ficam armazenados no servidor *Proxy.*

No caso de provedores, o controle de tráfego pode verificar a quantidade de dados transferidos pelos usuários, para assim determinar a largura de banda necessária e conseqüentemente, a capacidade de usuários que a rede suporta, sem afetar o desempenho do *link*[31] que o provedor possui.

As medições de tráfego auxiliam na realização de testes em equipamentos recém-lançados pelos fabricantes, ou até mesmo na definição de melhores rotas para servidores. Em pesquisas, a medição de tráfego tem ajudado no melhoramento de protocolos, como o HTTP e em avaliações de técnicas, desempenhando um papel importante para os pesquisadores, que estudam a dinâmica do tráfego da *Web*.

Dentre algumas das técnicas utilizadas para a análise do tráfego estão os *logs*, gerados nos pedidos de servidores, *proxies* e *browsers*. Além disso, os rastros do tráfego da *Web* podem ser coletados pela monitoração passiva dos *links* e roteadores na rede. Os servidores normalmente geram *logs*, que registram informações sobre o cliente solicitante, o horário de pedido e as mensagens de requisição e resposta. Devido a sobrecarga não estão inseridas mais informações nestes *log*.

As companhias telefônicas, já discutem as cobranças baseadas na quantidade de dados transmitidos pelo usuário, por meio de aplicações que monitoram o tráfego de dados na rede.

---

[30] *Proxy: Software* intermediário entre o cliente e o servidor, utilizado para compartilhamento de conexão e também como servidor de *caching;*

[31] *Link:* Canal de transmissão de dados.

3.1 COBRANÇA SOBRE TRÁFEGO DE DADOS

A criação de uma cota para transmissão de dados está em discussão há alguns anos, mas nenhuma grande operadora de banda larga colocou em prática esta forma de cobrança. Alguns serviços, como o Ajato[32] , não têm cotas nem na teoria, enquanto outros, como o *speedy*[33] e o *virtua*[34], afirmam que irão cobrar por excesso de dados trafegados, mas eventualmente adiam a cobrança. A empresa Telefônica disponibilizou uma tabela com os valores das cotas para cada plano, que pode ser analisada no anexo A. Alguns provedores possuem cotas especificadas nos contratos de *Service Level Agreement* (SLA)[35] e poderão a qualquer momento efetuar a cobrança, porém a barreira provavelmente é a concorrência, pois a companhia que fizer isso primeiro certamente perderá clientes, ou então, problemas de ineficiência na infra-estrutura de comunicação para controlar esse tipo de serviço.

A *Brasiltelecom* é uma empresa de telecomunicações que fornece serviços para grande parte do Brasil, sua área de cobertura é ainda maior no sul do país, onde fornece serviços de telefonia fixa, acesso à Internet em banda larga e telefonia móvel.

Turbo[36] é o serviço de Internet banda larga da *Brasiltelecom*. Como a ADSL é caracterizada por transmitir dados de forma assíncrona, suas taxas na *Brasiltelecom*, variam de 56 Kbps a 512 Kbps para *upload* e 150 Kbps a 2Mbps para *download*.

No contrato da empresa, não está especificada em nenhuma cláusula a garantia de alcance da taxa nominal contratada pelo usuário. Além disso, na categoria *Turbo Lite*, há a limitação de 50 horas de acesso, na qual o usuário que contratar o

[32] *Ajato*: Provedor de Internet banda larga da TVA. (www.ajato.com.br);
[33] *Speedy*: Provedor de Internet banda larga da Telefônica (www.speedy.com.br);
[34] *Vírtua*: Provedor de Internet banda larga da Net.(www.virtua.com.br);
[35] SLA: Contrato que descreve o acordo entre o usuário e o prestador de serviços;
[36] Turbo: Provedor de Internet banda larga da *Brasiltelecom*.(www.brasiltelecom.com.br).

serviço de acesso à Internet e ultrapassar o tempo de uso pré-determinado, pagará um valor adicional pelo tempo excedido. Nas demais categorias não há limitação por tempo, isto é, o usuário pode usar o tempo que achar necessário que não haverá acréscimos no valor da mensalidade. Em nenhum momento, foi citado no contrato que a empresa cobrará por tempo em todos os planos, mas isso poderá acontecer, haja vista que é uma forma da empresa aumentar sua arrecadação e diminuir o intenso tráfego de dados na rede.

O *vírtua* possui a cobrança por tráfego de dados descrito em seus contratos de prestação de serviços. Insinua que funciona de forma similar a cobrança por tempo de uso, já que será cobrado um valor adicional, pelos *bytes* que excederem a cota definida pela prestadora dos serviços.

As companhias telefônicas e os usuários devem cumprir deveres e obrigações quando um acordo entre ambos é firmado. O documento onde se encontram essas cláusulas é chamado de contrato SLA.

3.2 ACORDO EM NÍVEL DE SERVIÇO (SERVICE LEVEL AGREEMENT - SLA)

A quantidade de serviços cresceu muito nas redes de comunicação, devido a grande evolução tecnológica ocorrida nos últimos anos. Conseqüentemente empresas terceirizaram serviços por não poderem custear o desenvolvimento de suas próprias ferramentas de comunicação. A terceirização exige contratos que descrevem os deveres e obrigações do cliente e do fornecedor.

O conceito aplicado nestes contratos, é chamado de Modelo para Gerenciamento de Qualidade de Serviço - *Service Level Management* (SLM). O acordo

que estabelece os deveres e obrigações tanto do prestador de serviços quanto do usuário é chamado de *Service Level Agreement* (SLA).

Dentre as obrigações do prestador de serviços, está o compromisso da performance dos serviços, a infra-estrutura, confidencialidade e a segurança dos dados.

De acordo com Pinheiro (2004a), o SLA exige a utilização de indicadores de desempenho e outros fatores:

a) um objetivo;

b) a descrição do serviço;

c) o horário e a freqüência;

d) valores mínimos que terão que ser alcançados pela prestadora de serviços, em cada período de medição;

e) penalidades e

f) as responsabilidades (Prestadora de serviços, cliente e terceiras partes).

Esses itens se referem a cada tipo de serviço prestado. A prestadora de serviços ainda deve oferecer permissão para o monitoramento da performance dos serviços durante a duração do contrato.

Conforme Pinheiro (2004a), os critérios de desempenho propostos no SLA são:

a) disponibilidade;

b) confiabilidade;

c) pontualidade e

d) eficiência dos serviços.

Para o controle destes indicadores de desempenho, o usuário dispõe de *softwares* que gerenciam em tempo real a qualidade dos serviços (QoS)[37].

[37] QoS: A qualidade de serviço em redes são utilizadas para garantir e assegurar a qualidade priorizando o tráfego de dados de certas aplicações como voz sobre IP, videoconferência e telemedicina;

Dentre os problemas que mais ocorrem com os provedores são:

a) desempenho inferior ao prometido;

b) falhas momentâneas e

c) tempo gasto com reparos.

Um caso prático é a prestadora de serviços *Speedy*, que garante 10% da taxa nominal, enquanto a *Velox* e a *Giro*[38], não garantem a velocidade contratada devido ao congestionamento influenciado pelo tráfego nas linhas-tronco. (LOPES, 2005).

Para a provedora ser bem aceita no mercado, precisa cumprir as cláusulas do contrato. Já o usuário deve utilizar os serviços com fins lícitos, obedecendo aos deveres e obrigações descritas no contrato.

Dentre as aplicações utilizadas para o acesso à Internet e o compartilhamento da conexão estão as aplicações *proxies*, que são permitidas nos contratos SLA, desde que pertençam ao mesmo assinante.

3.3 APLICAÇÕES *PROXIES*

As aplicações *proxies* têm a função de intermediar a comunicação entre um *host* cliente e um servidor, com o objetivo de centralizar as conexões, permitindo o compartilhamento do acesso à Internet dentro de uma rede local, com recursos de FTP, HTTP e outros.

Proxies são capazes de monitorar, filtrar e bloquear qualquer tipo de informação antes de enviá-los ao destino.

Também podem aumentar a velocidade das requisições e respostas, utilizando *caching*. Portanto, pode-se dizer que o *proxy* deixa os servidores mais

[38] Provedor de Internet Banda Larga da *Embratel*. (www.giro.com.br)

próximos dos clientes e reduz o congestionamento na Internet, pois menos *bytes* são transferidos entre o *proxy* com *caching* e o servidor de origem. O sistema de *caching* aumenta a performance de acesso às páginas *Web*, porque armazena os documentos carregados e os disponibiliza num servidor local, aumentando assim a velocidade de acesso.

Uma desvantagem é que o conteúdo armazenado no *proxy* pode estar desatualizado, pois em alguns casos o mesmo realiza a atualização em tempos pré-definidos, por isso informações que são alteradas no servidor de origem podem estar diferentes no *proxy*.

É muito comum casos em que até chegar no servidor de origem, existam vários *proxies*, formando uma espécie de hierarquia de intermediários.

> Essas coleções hierárquicas de *proxies* são muito comuns e muito úteis nos países onde os custos de comunicação para acesso à Internet são altos. Restringindo uma parte razoável da comunicação e seus custos. (KRISHNAMURTHY; REXFORD, 2001, p.81).

Dentre as diversas funções dos *proxies*, também pode-se citar o anonimato, pois os servidores de origem conhecem o endereço IP do *proxy*, mas não sabem os endereços IP das máquinas que compõem a rede onde está instalado o *proxy*.

Além disso, o *proxy* realiza a conversão de mensagens de diferentes protocolos, por exemplo, um cliente da *Web* envia um pedido a um servidor FTP, o pedido em URL vem como HTTP e precisa ser convertido para se comunicar com o servidor FTP. Assim como um *gateway*, o *proxy* faz a configuração de conexão, autorização, envio da seqüência de comandos e fechamento de conexão.

O usuário que necessitar diminuir o tempo de carregamento de uma página *Web*, poderá utilizar um *proxy* que filtre pedidos de um *site*, bloqueando o carregamento

de imagens, sons e vídeos. Ou ainda, pode auxiliar uma empresa que deseja restringir o acesso a determinados *sites*.

A figura 5, representa um processo sendo iniciado por uma entrada URL num *browser*, que faz a consulta ao servidor DNS, passa pelo *proxy* e logo chega ao servidor de origem. Após esse caminho ser percorrido uma vez, os dados do *site* consultado ficam armazenados no *proxy* para uma futura consulta.

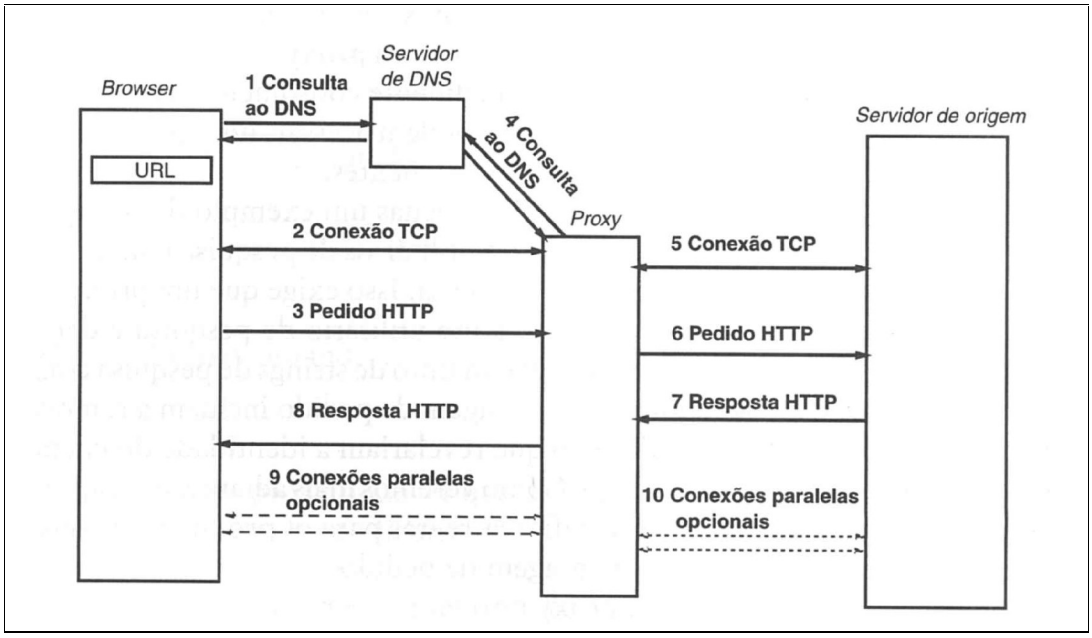

Figura 5. Utilização de *browser* com *proxy* no caminho.
Fonte: KRISHNAMURTHY B.; REXFORD J. (2001, p. 74)

São discutidos novos conceitos com relação a *proxies*, mas que ainda não foram padronizados. Por exemplo, o *proxy* reverso, que se localiza na frente de um servidor muito solicitado, tem função de evitar a sobrecarga ou até ataques de negação de serviço. Outro, é o *proxy* de interceptação, que examina o conteúdo do tráfego da rede.

Muitas aplicações possuem suporte a *proxy*, e entre os mais comuns estão os navegadores *Web*.

3.4 *BROWSERS*

Na pesquisa realizada, os *browsers* são utilizados para o acesso à Internet, de forma que se conectam a aplicação desenvolvida.

Exemplos de *browsers* são o *Internet Explorer* e o *Mozilla Firefox*, que utilizam configurações semelhantes para conectarem-se a um *proxy*. Os *browsers*, são um dos principais responsáveis pela popularidade da *Web*, já que sua interface com hipertexto facilitou a navegação para o usuário.

O *browser* é uma aplicação cliente, que segundo Krishnamurthy e Rexford (2001, p. 29), "constrói e envia um pedido HTTP, depois recebe, analisa e apresenta a resposta". Para o *browser* acessar um servidor, precisa da URL, que é traduzida num servidor de DNS, para depois estabelecer uma conexão com o mesmo e enviar o pedido HTTP.

Quanto aos aspectos de segurança, se destacam as aplicações executáveis que rodam localmente e podem oferecer riscos ao usuário, como a capacidade da *JavaScript*[39] de executar comandos sem conhecimento do usuário. Outra preocupação, é em relação aos *cookies*, que mantêm informações do usuário em servidores sem autorização, causando assim, muita discussão quanto à privacidade dos usuários.

No *browser*, há configurações de segurança que possibilitam a utilização ou a restrição de *cookies*. Além disso, é possível configurar propriedades que variam de cores e idioma até o tipo de conexão, via *proxy* ou diretamente no servidor.

Além de acessar a *Web*, os *browsers* podem acessar *sites* de *e-mail*, entrar em sessões de bate-papo e outros. Os *browsers* são feitos para carregarem arquivos *HyperText Markup Language* (HTML), que muitas vezes não são suficientes para

[39] *JavaScript*: Linguagem de programação para *web*, executada localmente.

satisfazer todos os recursos, como abrir um documento *Portable Document Format* (PDF), mas, existem as chamadas aplicações auxiliadoras, que o *browser* pode disparar e então abrir esses documentos.

Um outro recurso que os *browsers* apresentam, é o armazenamento local de mensagens chamado de *caching*, que pode ser utilizado para diminuir o tempo de carregamento de um *site*.

Várias pesquisas são realizadas na área de tráfego de dados. Alguns *softwares* são pagos e geralmente desenvolvidos para plataforma *Microsoft Windows*, enquanto outros, são gratuitos e distribuídos para a plataforma *Linux*.

4 TRABALHOS CORRELATOS

Várias pesquisas compreendem a área de controle e monitoramento de dados em redes de computadores, com a intenção de aperfeiçoar e melhorar a qualidade dos serviços.

Em 2000, foi escrito um artigo por Santos (1999), que se refere a um controlador de banda e está disponível no *site* da Rede Nacional de Ensino e Pesquisa (RNP). O artigo descreve soluções para se ter um maior controle dos recursos disponíveis, dentro de domínios que oferecem Qualidade de Serviços (QoS).

O trabalho foi realizado com base na arquitetura *DiffServ*[40], que é uma forma de diferenciar aplicações definindo a quantidade de banda utilizada, atrasos e latência na transmissão dos dados. Além disso, gerencia a alocação de recursos conforme as políticas de tráfego de dados especificadas e explica como ocorre a comunicação entre as aplicações. O trabalho propõe o desenvolvimento de uma aplicação chamada *Bandwidth Broker* (BB), que podem trabalhar em hierarquia.

Outro trabalho, desenvolvido pelos acadêmicos Rocha, Veloso, Meira e Almeida (2002), da Universidade Federal de Minas Gerais, propõem o desenvolvimento de uma aplicação *proxy* que utiliza a técnica de *pushing*[41], chamada de *Proxy A*tivo que utiliza a mineração de dados para o armazenamento de informações em *cache*. O trabalho foi desenvolvido, para amenizar o problema de lentidão em conexões de Internet com pequena largura de banda, como as utilizadas em linhas discadas.

Na Universidade do Extremo Sul Catarinense, a acadêmica Tiscoski (2005), apresentou um trabalho que demonstrou uma avaliação de desempenho em redes de Linha Digital Assimétrica para Assinante (ADSL), na região de Criciúma, com ênfase

[40] *DiffServ*: Rede com serviços diferenciados;
[41] *Pushing:* Técnica em que o *proxy*, ao invés da aplicação cliente, toma a iniciativa de armazenar as informações em *cache.*

em aspectos que influenciam a largura de banda, como os navegadores, provedores de acesso e atualização de *hardware.*

Dentre algumas soluções disponíveis no mercado, têm-se o *Wingate e o Winconnect,* que são *softwares* que permitem o compartilhamento de Internet, possuem recursos de *caching*, ajuste de banda, bloqueio e monitoramento de *sites*. Além disso, apresentam recursos de *log*. Permitem visualizar o que cada máquina está fazendo, controle de quais usuários têm permissão para acessar a Internet e possibilita configuração de horários pré-estabelecidos para liberação de acesso. Ambos são compatíveis com o sistema operacional *Microsoft Windows,* possuem código-fonte fechado e são *softwares* proprietários.

O *Squid* é um *proxy* alternativo para a plataforma *Unix,* que funciona como um intermediário entre a conexão do cliente e servidor. Neste meio caminho armazena os objetos que foram solicitados e permite que as próximas requisições para os mesmos objetos possam ser respondidas localmente, deixando assim a conexão mais rápida. Suporta protocolos FTP, HTTP e outros. É caracterizado por permitir flexibilidade no controle de acesso e configurações, além de garantir segurança com *Secure Sockets Layer* (SSL)[42] e criação de *logs* avançados. *O Squid* é um *software* livre, que pode ser modificado e distribuído conforme a licença GPL[43](*General Public License*) e está disponível para *download* no *site www.squid-cache.org.*

Também há o *Hierarchical Token Bucket* - HTB, um recurso gratuito disponível em versões recentes do *Linux* que realiza o controle de banda, por meio de *scripts* que monitoram a passagem de dados em portas específicas.

Para complementar esta pesquisa, foi desenvolvida uma aplicação experimental, que demonstra na prática alguns conceitos abordados.

---

[42] SSL: Protocolo utilizado para promover segurança na transmissão de dados;
[43] GPL: Licença que garante a liberdade de compartilhar e alterar *softwares* livres.

5 *O WEBLIMIT*

Na aplicação experimental desenvolvida, chamada *WebLimit*, foram inseridos alguns conteúdos pesquisados e discutidos neste Trabalho de Conclusão de Curso, haja vista que propõe o Controle do Uso de Banda em Redes de Computadores, baseado em uma aplicação *proxy* responsável por prover o acesso à Internet numa rede local.

O método de captação de informações foi aplicado durante o acesso dos computadores clientes ao servidor *proxy*, onde é realizada a interceptação da conexão e nesse momento obtida a quantidade de *bytes* transmitidos. Logo depois, é armazenado no banco de dados os valores referentes a utilização da rede por usuário, tendo o endereço IP como chave de identificação.

O Administrador do sistema define permissões, visualiza a quantidade de tráfego de cada usuário, especifica a cota limite de *bytes* transmitidos e também o bloqueio dos usuários em tempo real. O usuário cliente acessa a Internet por meio de *browser*, configurado para conectar no endereço IP e porta 8080 do servidor *proxy*.

Devido a limitação dos componentes utilizados, a aplicação desenvolvida oferece ao *browser* somente recursos provenientes do protocolo HTTP. Sendo assim, o usuário dispõe de navegação em páginas *Web*, bem como, *download* e *upload* utilizando HTTP.

Nos testes realizados foram utilizados três computadores, sendo um servidor e dois clientes. Um cliente e o servidor, tinham o sistema operacional *Microsoft Windows XP* instalados, no outro, o *Microsoft Windows 98.* O meio de acesso à Internet utilizado foi ADSL com *link* de 600 Kbps da *Brasiltelecom*.

5.1 METODOLOGIA

O desenvolvimento do *WebLimit* fundamentou-se metodologicamente pelas seguintes etapas: revisão bibliográfica, estudo sobre arquitetura TCP/IP, pesquisa sobre tipos de acesso à Internet, estudo sobre programação e componentes *Delphi*, modelagem e implementação da aplicação servidora.

5.1.1 Revisão Bibliográfica

Consistiu no levantamento bibliográfico dos temas envolvidos na pesquisa, com o intuito de compreender e descrever os conceitos fundamentais para o desenvolvimento do projeto.

Nesta etapa aprofundou-se os estudos sobre redes de computadores e controle de tráfego de dados, visto que se constituem na base da fundamentação teórica. Para o desenvolvimento da aplicação foram pesquisados os conceitos e o funcionamento da programação em *Delphi*.

5.1.2 Modelagem do *WebLimit*

A modelagem da aplicação foi elaborada em *Unified Modeling Language* (UML), por meio de diagramas que representam a interação entre usuário e aplicação.

As ações que o administrador e os usuários desempenham são descritas no diagrama de casos de uso. O administrador do sistema tem a responsabilidade de definir permissões para os usuários, como: o limite total de *bytes* transmitidos durante o mês, dar permissão aos usuários de acessarem a Internet mesmo tendo seu limite de

tráfego atingido, também bloquear o usuário em tempo real e visualizar a quantidade de *bytes* sendo transmitidos durante uma conexão.

O usuário comum efetua a entrada no sistema por meio de navegador *Web* e acessa a Internet caso não esteja bloqueado. Na figura 6, é demonstrado o Diagrama de Casos de Uso.

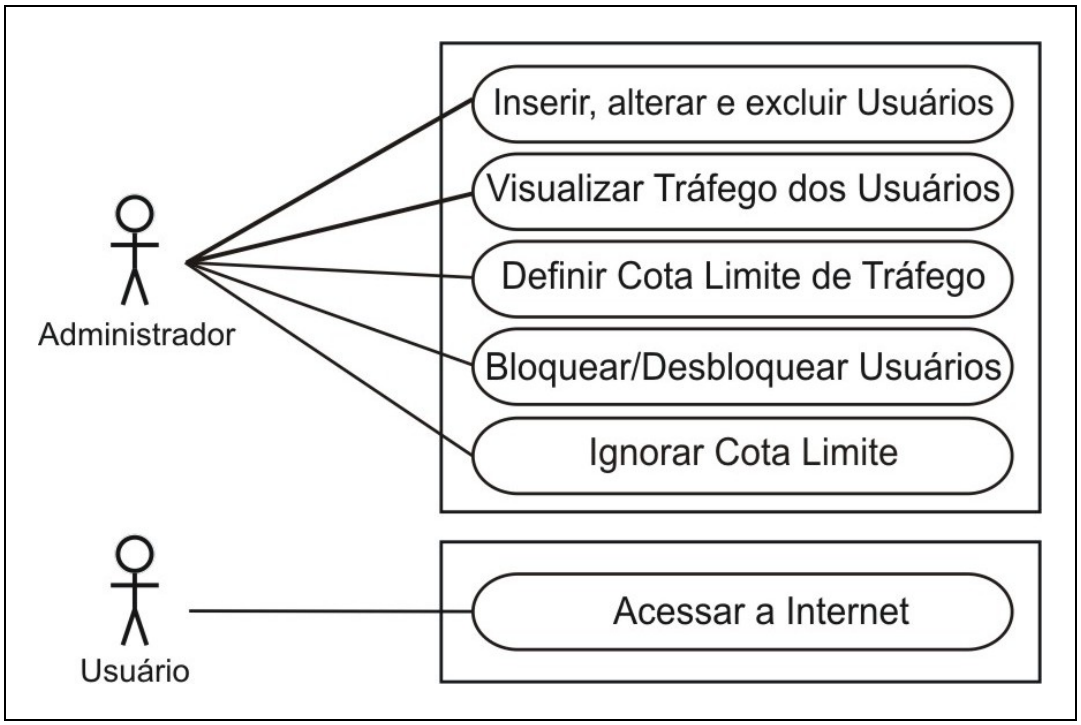

Figura 6. Diagrama de Casos de Uso

A seguir será demonstrado o Diagrama de Classes, que é composto por quatro entidades.

As classes Administrador, TráfegoDia, TráfegoTotal estão associadas a classe principal Usuário.

Na figura 7 é demonstrado o Diagrama de Classes.

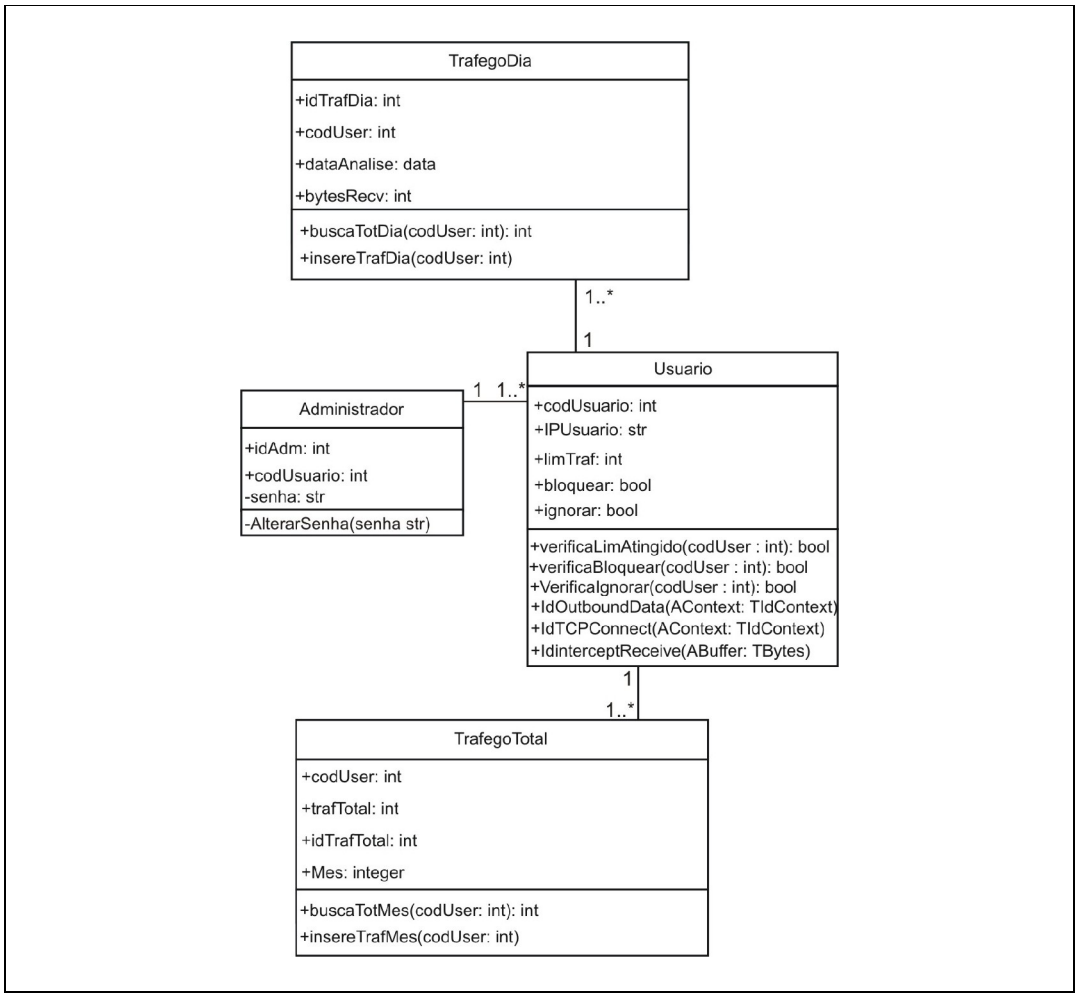

Figura 7. Diagrama de Classes

A descrição da composição do Diagrama de Classes é feita nas tabelas 5, 6, 7 e 8.

Tabela 5. Classe Usuário

Atributo	Tipo	Descrição
codUsuario	Integer	Código do usuário
IPUsuario	VarChar	IP do Usuário
Bloquear	Boolean	Bloqueio em Tempo Real
limTrafego	Integer	Cota limite de Tráfego de dados
Ignorar	Boolean	Permissão Acesso com Limite Atingido

Tabela 6. Classe Administrador

Atributo	Tipo	Descrição
idAdm	Integer	Identificador
codUser	Integer	Código do Administrador
Senha	VarChar	Senha do Administrador

Tabela 7. Classe TráfegoDia

Atributo	Tipo	Descrição
idTrafDia	Integer	Identificador
bytesRecv	Integer	Bytes Recebidos
dataAnálise	Date	Data dos Bytes Recebidos
codUser	Integer	Código do Usuário

Tabela 8. Classe TráfegoMês

Atributo	Tipo	Descrição
idTrafMês	Integer	Identificador
trafTotal	Integer	Bytes Recebidos
Mês	Integer	Data dos Bytes Recebidos
codUser	Integer	Código do Usuário

No Diagrama de Atividades são descritas as etapas de funcionamento do *WebLimit*. A aplicação aguarda por requisições, na qual recebe o endereço IP da máquina cliente e verifica sua existência no sistema. Caso o IP seja desconhecido, ou seja, não esteja cadastrado, sua conexão é cancelada e o mesmo deverá requisitar ao administrador que o cadastre no sistema.

A partir do endereço IP, é possível localizar as demais informações, como: verificar se o usuário está no estado bloqueado, comparar o valor do limite de tráfego com o já efetuado e caso este seja igual ou maior, o usuário tem seu tráfego impedido. Com base nas condições anteriores, o sistema definirá se a permissão de acesso será concedida.

Ainda assim, o usuário poderá acessar a Internet caso tenha seu limite atingido, pois o atributo “ignorar limite”, possibilita a navegação mesmo tendo seu limite de tráfego atingido.

Na figura 8 é demonstrado o Diagrama de Atividades.

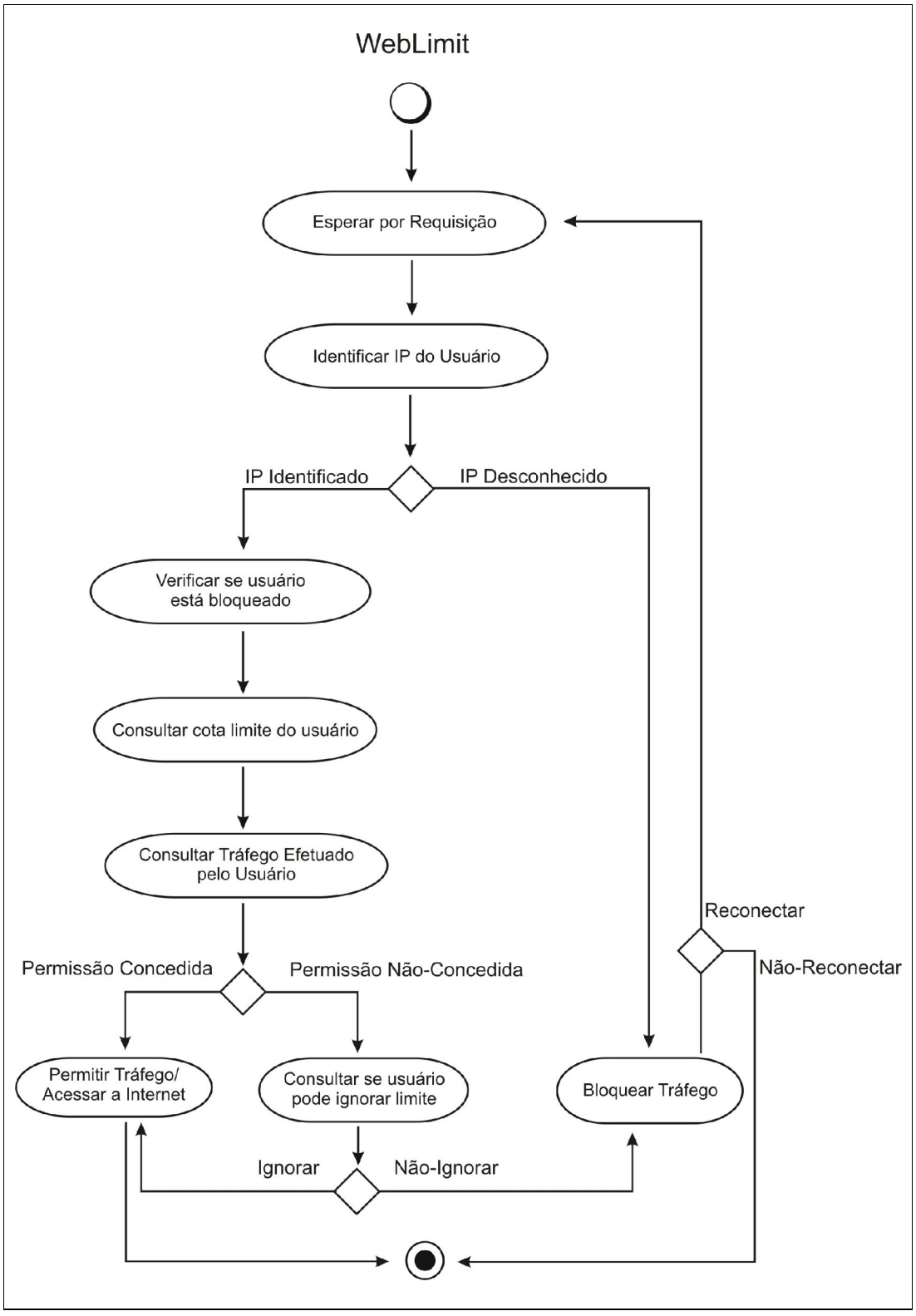

Figura 8. Diagrama de Atividades

5.1.3 Desenvolvimento do *WebLimit*

A aplicação desenvolvida foi projetada para suportar uma rede local com até 64 computadores.

O código-fonte foi escrito na linguagem de programação *Object Pascal*, utilizando o ambiente *Delphi* 7, com o auxílio de componentes *Indy* e *Zeos*. Ambos são gratuitos e disponíveis nos *sites www.indyproject.org* e *www.sourceforge.net/projects/zeoslib* respectivamente.

Os principais componentes utilizados na aplicação foram o *IdHTTPProxyServer*, *IdConnectionIntercept* e o *IdMappedPortTCP*, todos pertencentes a versão *Indy* 10.

No *WebLimit*, o componente *IdHTTPProxyServer*, que é identificado pela porta 8081, troca informações com a rede externa através de uma porta aleatória.

O *IdMappedPortTCP*, que é um componente que faz mapeamento de portas, constrói uma ponte para redirecionar o tráfego da porta 8081 para a porta 8080. Neste momento entra em ação o componente *IdConnectionIntercept*, que intercepta a conexão e obtém o valor dos *bytes* transmitidos.

Para navegar na *Web*, os computadores clientes precisam simplesmente de um *browser* configurado para conectar-se ao IP do *proxy* servidor na porta 8080.

Na figura 9 são demonstrados os componentes utilizados.

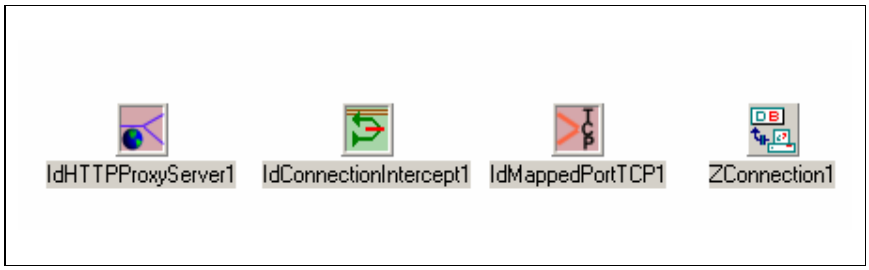

Figura 9. Principais componentes utilizados no *WebLimit*

O componente *Zeos*, é o responsável por fazer a conexão com o banco de dados *MySQL*. Nele estão disponíveis componentes para se conectar a tabelas do banco, fazer consultas utilizando SQL e outros.

A Figura 10 descreve o método e os componentes utilizados para a interceptação da conexão de cada usuário.

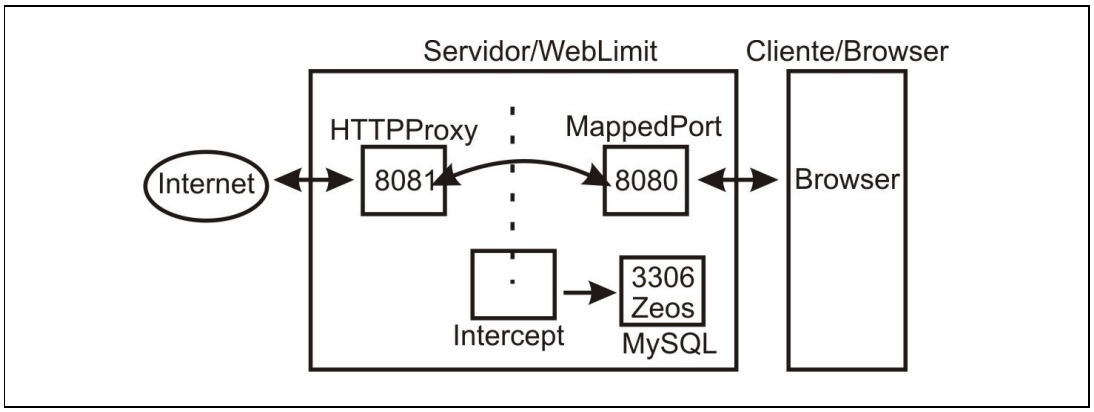

Figura 10. Interceptando a Conexão

Devido à limitação do componente utilizado, a aplicação tem suporte somente ao protocolo HTTP. Dessa maneira os usuários podem acessar qualquer página que processe este tipo de protocolo, bem como, fazer *downloads* e *uploads*.

A aplicação monitora os *bytes* transmitidos durante a conexão, possibilitando também o bloqueio e desbloqueio dos usuários e visualização do endereço IP de cada máquina que se conecta ao *proxy*.

O administrador da aplicação pode definir a quantidade máxima permitida que cada usuário pode transmitir, também possibilita que o *proxy* ignore o limite definido e libere a conexão para determinados usuários.

Na figura 11, é demonstrado um trecho do código-fonte, mais especificamente a *procedure IdMappedPortTCP1Connect*, que é acionada a cada requisição dos computadores clientes da rede local.

O endereço de IP é identificado nesta *procedure*, onde se localizam as propriedades de cada conexão. São realizadas comparações que determinam as condições possíveis para o que usuário tenha seu tráfego de acesso à Internet liberado.

```
Procedure TForm1.IdMappedPortTCP1Connect(AContext: TIdContext);
var
  codigoUser,  i: integer;
begin
  codigoUser := 0;
  Memo1.Lines.Add (AContext.Connection.Socket.Binding.PeerIP); //exibe endereço IP
  ipremoto := AContext.Connection.Socket.Binding.PeerIP;
  for i := 0 to 255 do
    if (AnsiSameStr(ipLiberado[i],ipremoto)) then //compara com a lista dos IPs liberados
    begin
      codigoUser := i;
      break;
    end;
   codUsuario := codigoUser; //codUsuario global
   if (codUsuario=0) or
   (bloquear[codUsuario]=TRUE)or
   ((limiteAtingido[codUsuario]=TRUE)and(ignorar[codUsuario]=FALSE)) then
   begin
     AContext.Connection.Disconnect; //impede a conexão
   end;
   if ((IPCodSelecionado=codUsuario)and(bloqueioTReal[IPCodSelecionado]=TRUE)) then
   begin
     AContext.Connection.Disconnect;
   end;
end;
```

Figura 11. Código-fonte da aplicação: *Procedure IdMappedPortTCP1Connect*

Na figura 12, é demonstrada a *procedure IdConnectionIntercept1Receive*, que tem o valor da quantidade de *bytes* transmitidos em cada conexão. As estruturas de dados utilizadas para armazenamento temporário dos *bytes* transmitidos, foram três vetores, um para armazenamento dos *bytes* transmitidos na conexão em tempo real, um para valor diário e outro para o mensal. A cada dez segundos é feito o armazenamento no banco de dados.

```
Procedure TForm1.IdConnectionIntercept1Receive(
  ASender: TIdConnectionIntercept; var ABuffer: TBytes);
var
   AuxDia, AuxMes : integer;
begin
   Real[codUsuario] := Real[codUsuario]+Length(ABuffer);
   Dia[codUsuario] := Dia[codUsuario]+Length(ABuffer);
   Mes[codUsuario] := Mes[codUsuario]+Length(ABuffer);
   AuxDia := Dia[IPCodSelecionado] div 1024; //transformando em KBytes
   AuxMes := Mes[IPCodSelecionado] div 1024000; //em MBytes
   lbtotTReal.caption := intToStr(Real[IPCodSelecionado])+' bytes';
   lbTotDia.caption := intToStr(AuxDia)+' KBytes';
   lbTotMes.caption := intToStr(AuxMes)+' MBytes';
end;
```

Figura 12. Código-fonte da aplicação: *Procedure InterceptReceive*

5.1.4 Interfaces da Aplicação

Na figura 13, é demonstrada a interface principal da aplicação, onde são exibidos os valores dos *bytes* transmitidos, para cada usuário selecionado na caixa de seleção. Caso o administrador necessite saber o valor da quantidade de *bytes* necessários para carregar uma certa página, o mesmo pode zerar o valor atual antes de carregá-la e observar a nova quantidade exibida no tráfego atual.

Ainda são representadas as opções de bloquear e desbloquear o acesso de usuários à Internet em tempo real. Na mesma interface possui o botão “configurações”, que dá acesso a interface de configuração dos usuários.

Por meio da barra de menus é possível acessar o ambiente de configuração de usuários, zerar o tráfego da conexão atual para efeitos de visualização, mas sem reflexos no banco de dados e por fim a opção sair da aplicação. Ainda nesta barra está localizada a opção de ajuda que auxilia o administrador a interagir com o *WebLimit.*

Figura 13. Interface principal da aplicação

A figura 14 demonstra outra interface da aplicação, onde são definidos: o limite de tráfego dos usuários e a permissão para acessar a Internet mesmo tendo seu limite de tráfego atingido. Somente terá acesso à Internet os IPs que estiverem cadastrados no sistema.

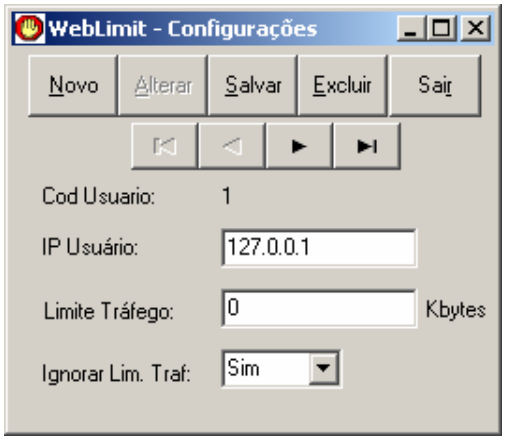

Figura 14. Interface de configurações

5.1.5 Banco de dados do *Weblimit*

No *WebLimit* foi utilizado o *MySQL*, que é um sistema de gerenciamento de banco de dados relacional. Permite armazenar, pesquisar, classificar e recuperar dados eficientemente. Tem suporte a linguagem SQL para realização de consultas, inserções, exclusões e atualizações.

Na aplicação desenvolvida o componente *Zeos,* faz a conexão com o banco de dados na porta 3306 com permissões para leitura e escrita.

Os atributos e tabelas estão descritos no diagrama de Entidade-Relacionamento do Apêndice A deste trabalho.

Durante a execução da aplicação é feita a inserção do tráfego de cada usuário a cada dez segundos, tendo como identificador o número de IP do usuário. As informações são armazenadas em *bytes* e demonstradas em *bytes*, *Kilobytes e Megabytes* respectivamente para os valores tráfego atual, tráfego do dia e mês na tela principal da aplicação.

Para melhor descrever as funcionalidades do *WebLimit*, foram realizados alguns testes.

5.1.6 Testes com o *WebLimit*

Na realização dos testes foram utilizados três computadores, um servidor e dois clientes. Todos ligados num *hub* de 8 portas conectado à Internet, por meio de um modem ADSL 500G da empresa *D-Link*. Inicialmente o servidor aguarda a requisição do cliente, logo após, troca informações e inicia a contabilização dos *bytes* transmitidos.

Um dos testes realizados foi a medição da velocidade da conexão. Para medir o acesso foi acessado o *site www.meuip.com.br*.

Sem *proxy* a velocidade foi de aproximadamente 109 Kbps, já com a utilização de *proxy* o valor demonstrado foi 52.240 Kbps. Essa diferença foi causada pelo *cache* do *proxy*.

Em um dos computadores clientes, foi utilizado o *Microsoft Windows* 98, que acessou normalmente o servidor *proxy* com sistema operacional *Microsoft Windows XP*.

Durante determinado período foram definidos 10 MBytes para um usuário específico. Dentre as atividades, foram feitas navegação, *downloads* e *uploads* utilizando o protocolo HTTP através do *browser Firefox 1.5*. Após atingir a cota definida pelo administrador, o *browser* não teve mais acesso a *web*. O usuário pôde acessar a Internet somente depois do administrador conceder permissão.

Sites que disponibilizam vídeos via HTTP, como a *www.globo.com*, transmitem os quadros com nitidez e sincronia, possibilitando a transmissão de voz e vídeo em tempo real, mesmo com a utilização de *proxy*.

Durante a realização de *downloads*, foi percebido que o método utilizado pelo componente para baixar arquivos da Internet acaba deixando o usuário confuso, pois primeiramente copia todo o arquivo sem demonstrar barra de progresso durante a transferência de arquivos. Após copiar todo o arquivo, questiona o usuário sobre o local de armazenamento e quando o usuário indica o local o armazenamento do arquivo no *hard disk* é instantâneo.

Dentre algumas fragilidades de segurança, foram testados no *WebLimit* a utilização do IP de outros usuários em computadores diferentes. Essa trapaça pode ser praticada através da técnica de [44]*IP Spoofing*.

Na utilização do *browser* para obter o acesso a Internet o usuário deverá utilizar o IP do servidor na porta 8080, mas deve-se ter um cuidado para adicionar estas informações em todos os protocolos, caso contrário alguns não terão acesso à Internet. Apesar da aplicação ser limitada ao protocolo HTTP, ela pode realizar conversões entre alguns protocolos.

O servidor utilizado foi um microcomputador com plataforma *Microsoft Windows XP*, 256 Mb de memória e processador Duron de 1.2 MHz. Como clientes foram utilizados dois microcomputadores. O cliente A, com plataforma *Microsoft Windows XP*, 1Gb de memória e processador Intel Celeron de 2 GHz. O cliente B, com plataforma *Microsoft Windows 98*, 32 Mb e processador Intel de 233 MHz.

O teste foi realizado numa conexão ADSL da Brasiltelecom, com velocidade de 600 Kbps durante horário comercial. A figura 15 descreve o ambiente de testes.

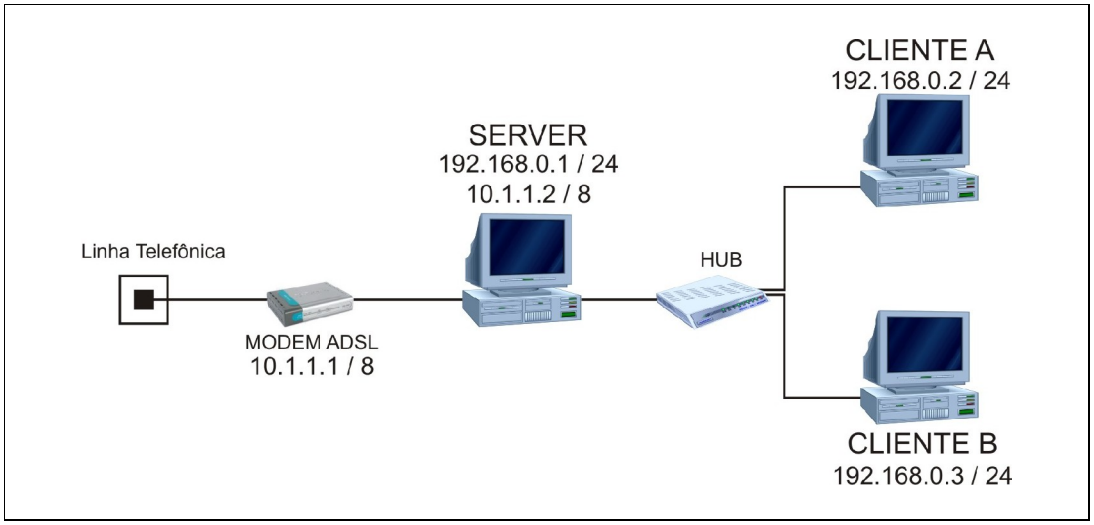

Figura 15. Descrição do Ambiente de testes

[44] IP *Spoofing*: consiste na troca do IP original por um outro, podendo assim se passar por um outro *host*.

CONCLUSÃO

A pesquisa realizada abordou a cobrança sobre tráfego de dados que alguns provedores de acesso à Internet estão propondo, onde o assinante deverá pagar uma taxa adicional por *megabyte* transmitido caso ultrapasse uma cota mensal. Alguns provedores já possuem a descrição dessa forma de cobrança em seus contratos de prestação de serviços, porém ainda não colocaram em prática, provavelmente pela possibilidade de perda de clientes, haja vista que a concorrência é também acirrada e também pela falta de infra-estrutura para realização deste tipo de controle.

O *WebLimit* foi a aplicação desenvolvida neste trabalho e se destaca por possuir código-fonte aberto para plataforma *Windows*, além de possibilitar a obtenção de *bytes* transmitidos durante as conexões, o que permite o controle do uso de banda em uma rede de computadores.

Durante os testes realizados pôde-se observar que administrar o uso de banda muitas vezes diminui os gastos com Internet, já que a aplicação impede que os usuários ultrapassem a cota limite de tráfego de dados.

Foram encontradas dificuldades principalmente no desenvolvimento da aplicação, já que foi necessário pesquisar uma função que retornasse os *bytes* transmitidos de cada computador da rede. Inicialmente pensou-se em obter os dados por meio da placa-de-rede, mas a complexidade e a carência de documentação referente a programação com APIs, obrigaram a mudança para métodos mais flexíveis como a programação com componentes *Indy* no *Delphi*.

Como trabalhos futuros sugerem-se:

a) aprimorar a aplicação com o desenvolvimento de novos serviços, como o filtro de conteúdo, que analisa o conteúdo do tráfego da rede, possibilitando

o bloqueio de informações vindas de *softwares* específicos como o MSN[45], *Kazaa*[46], *Orkut*[47] e *sites*, quando estes não atendem aos interesses da empresa;

b) a abrangência do trabalho está limitada ao protocolo HTTP e poderá ser expandida para outros protocolos como: FTP, UDP e outros;

c) capacitar a aplicação para geração de *logs*, obtendo assim estatísticas do uso de banda dos usuários e erros do sistema;

d) desenvolver módulo com o histórico de tráfego por usuário representando-o por meio de gráficos, bem como utilizar a aplicação para coleta de dados em pesquisas que auxiliam na definição do perfil de usuários quanto a utilização de banda em uma rede de computadores.

e) corrigir falhas de segurança do sistema, evitando a intrusão de outros usuários com a técnica de *IP Spoofing* e outros. Dentre algumas soluções está a autenticação com senha para usuários;

f) alteração do código-fonte dos componentes utilizados, para correção do método de *download* e *upload* da aplicação;

g) possibilitar aos usuários clientes acessarem diretamente o *WebLimit*, como um *gateway* na porta 80, sem necessidade de configuração do *browser* no IP e porta 8080 do servidor;

h) desenvolver uma aplicação que seja capaz de ajustar a capacidade da largura de banda, definindo um valor limite de velocidade por usuário.

[45] MSN: *Software de comunicação onde são trocadas mensagens instantâneas.*
[46] *Kazaa: Software* P2P utilizado para *download e upload* de arquivos*;*
[47] *Orkut: Site* de relacionamento entre pessoas na *Web*.

REFERÊNCIAS

COMER, Douglas E. **Interligação em Rede com TCP/IP:** Princípios, protocolos e arquitetura. Campus: Rio de Janeiro. 1998.

COMER, Douglas E; STEVENS, David L. **Interligação em rede com TCP/IP**: Projeto, implementação e detalhes internos. Rio de Janeiro: Campus, 1999.

COMITÊ Gestor da Internet.**CGI.** Disponível em: <www.cgi.br> Acesso em: Abril 2005.

IBOPE/*NetRatings*. **O Crescimento de Banda Larga no Brasil**. Disponível em: <http:/www.ibope.com.br> Acesso em: 08 Nov 2005.

INFO Exame**. As melhores marcas de tecnologia no país**. São Paulo, nº 229, Abril 2005.

KRISHNAMURTHY, Balachander; REXFORD, Jennifer. **Redes para a Web:** HTTP/1.1, Protocolos de Rede, *Caching* e Medição de Tráfego. Rio de Janeiro: Campus, 2001.

KUROSE, James F.; ROSS, Leith W. **Redes de Computadores e a Internet: Uma abordagem Top-Down**. São Paulo: Pearson, 2005.

LOPES, Airton. Banda Larga. **Info Exame**. São Paulo, nº 227, Fev. 2005.

MURHAMMER et al. **TCP/IP: Tutorial e Técnico**. Tradução: Jussara Licinia S. Gaertner, Lavio Pareschi; revisão técnica: Álvaro Antunes. São Paulo: Makron books, 2000.

PETERSON, Larry L; BRUCE Davie S.**Redes de Computadores:** Uma abordagem de sistemas Larry L. Peterson e Bruce S. Davie; Rio de Janeiro:Elsevier, 2004.

PINHEIRO, José Maurício S. **Gerenciamento em Níveis de Serviços.** Rio de Janeiro, 2004a. Disponível em: <www.projetoderedes.com.br/artigos>. Acesso em: 11 Nov. 2005.

PINHEIRO, José Maurício S. **Introdução ao ADSL.** Rio de Janeiro, 2004b. Disponível em: <www.projetoderedes.com.br/artigos>. Acesso em: 12 Nov. 2005

PINHEIRO, José Maurício S. **A Inevitável Convergência das Tecnologias.** Rio de Janeiro, 2004c. Disponível em: <www.projetoderedes.com.br/artigos>. Acesso em: 13 Nov. 2005

ROCHA, Bruno G.; VELOSO, Adriano A.; MEIRA Jr., Wagner; ALMEIDA, Virgilio A. F. **Proxy Ativo:** Diminuindo a Latência através de *Pushing*. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 20º : 2002 : Búzios, RJ. Anais...Rio de Janeiro: UFRJ, 2002. p. 151-162.

SANTOS, Ana Paula S. **O Controlador de Banda**. Disponível em: http://www.rnp.br. Acesso em: 20/11/05.

SOARES, Luiz Fernando G.; LEMOS, Guido; COLCHER Sérgio. **Redes de Computadores: Das LANs, MANs e WANs às Redes ATM**. Rio de Janeiro: Campus, 1995.

TANENBAUM**,** Andrew S. **Redes de Computadores**. Rio de Janeiro: Campus, 2003.

TISCOSKI, Gabriela Dal Toé. **Avaliação de Desempenho em Redes de Linha Digital Assimétrica para Assinante na região de Criciúma, utilizando Métrica**. Criciúma: Unesc, 2005.

TOLEDO, Adalton P. **Redes de Acesso a Telecomunicações**. São Paulo: Makron Books, 2001.

TORRES, Gabriel. **Redes de Computadores: Curso Completo**. Rio de Janeiro: Axcel Books, 2001.

WIRTH, Almir. **Tecnologias de Rede & Comunicação de Dados.** Rio de Janeiro: Alta Books, 2002.

BIBLIOGRAFIA RECOMENDADA

CANTÚ, Marco; TORTELLO.**Dominando o *Delphi* 6** : a bíblia. Tradução: João Eduardo Nóbrega. São Paulo: Makron Books, 2002. 934 p.

CONTROLE de Tráfego de Dados. **Associação Brasileira dos Usuários de Acesso Rápido (ABUSAR).** Janeiro, 2005. Disponível em: <http://www.abusar.org> Acesso em: 13 Nov 2005.

FERREIRA, Marco dos Santos. ***Delphi* 5.0: Tópicos Avançados**. São Paulo: Érica, 2000.

INDY, Project**. Componentes Indy para *Delphi***. Disponível em: <http:/www.indyproject.org > Acesso em: 10 Mar 2006.

MACHADO, Carlos. **Microsoft Windows XP**. Rio de Janeiro: Campus, 2001.

PENDER, Tom. **UML, a Bíblia**. Rio de Janeiro: Elsevier, 2004.

SONNINO, Bruno. **Desenvolvendo Aplicações com *Delphi* 6**. São Paulo: Makron Books, 2001.

WELLING, Luke; THOMSON, Laura. **PHP e MySQL**: desenvolvimento Web. Rio de Janeiro: Campus, 2003.

APÊNDICE A – DIAGRAMA DE ENTIDADE-RELACIONAMENTO

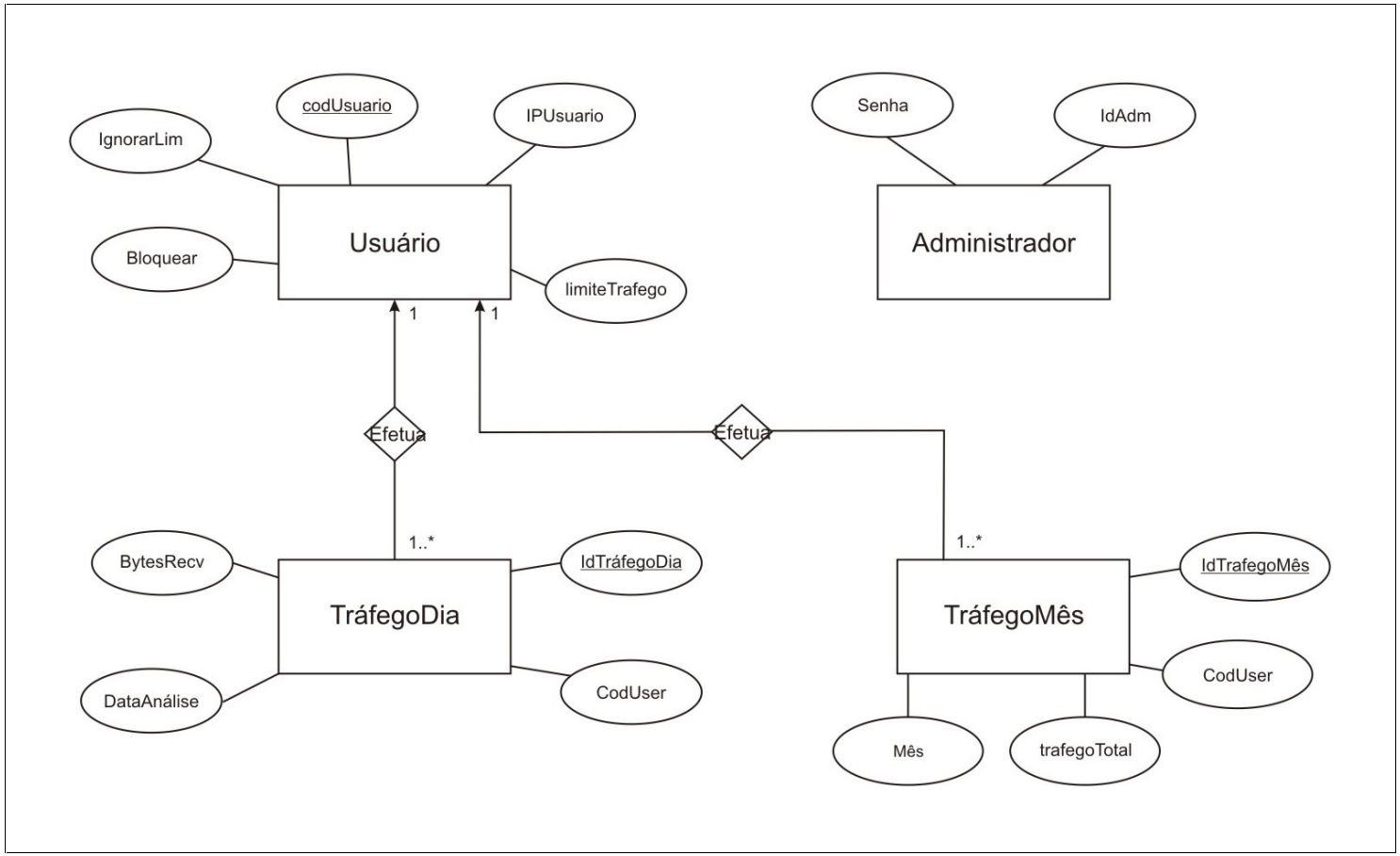

Figura 16. Diagrama de Entidade-Relacionamento

APÊNDICE B – INSTALAÇÃO DOS COMPONENTES NO DELPHI 7

A versão do *MySQL* utilizada foi a (mysql-4.1.3b-beta-win), disponível no *site* www.mysql.com.

O diretório para instalação deverá ser C:\mysql. Como o *MySQL* não cria nenhuma aba no menu Iniciar, logo é necessário executá-lo via *prompt*. Para abrir deve-se clicar em Iniciar, Executar e digitar o comando “cmd” no console.

Para executar os comandos sem precisar entrar em c:\mysql\bin é necessário criar as variáveis de ambientes, localizada na aba Avançado do menu Meu Computador. Na variável *PATH* deve-se adicionar logo após o ‘;’ o comando c:\mysql\bin. Agora já pode-se executar os comandos de qualquer lugar do *prompt*.

Alguns comandos do *MySQL* podem ser vistos na tabela 9.

Comando	Descrição
NET START mysql	Iniciar o mysql
NET STOP mysql	Parar o mysql
Mysqld-nt –install	Instala o mysql como serviço
Mysqlshow	Mostra os bancos existentes

Tabela 9. Alguns comandos do *MySQL*

Uma outra opção para ser utilizada, caso o banco de dados não tenha sido instalado com sucesso, é a instalação do *EasyPHP* 1.8, que possui o *MySQL 4.1*, o *Apache 1.3 e o PHP 4.3*. Possuem fácil instalação e permitem executar o banco de dados por meio do menu “Iniciar”. Estão disponíveis gratuitamente no *site www.easyphp.org*.

Para instalar o *Zeos*, componente que faz a conexão com o banco de dados, é necessário descompactar o arquivo (zeosdbo-6.5.1-alpha) para a pasta c:\Zeos. Após descompactar, é necessário copiar a dll, que está na pasta *Zeos* e que coincida com a versão do banco de dados instalado no micro, por exemplo *MySQL 4.1* e adicioná-la na pasta system32 do *Windows*. Depois Abrir o projeto ZeosDbo.bpg no *Delphi*, referente

a versão do que se está utilizando, nesse caso a versão sete. Caso não apareça uma janela contendo os pacotes, vá em *View, Project manager*. Um a um, obedecendo a ordem, compilar e depois instalar.

O componente utilizado para conectar ao banco de dados foi encontrado no *site* http://sourceforge.net/projects/zeoslib/ , mais especificamente no *link View older releases from the Zeos Database Objects package*, que consiste em versões anteriores e compatíveis com o *Delphi* 7. Foram testadas versões mais recentes e não foi possível realizar a instalação com sucesso.

Ao compilar as aplicações pode ocorrer do *Delphi* não encontrar algum arquivo. Sendo assim pode-se resolver isso, clicando na opção *Tools, Enviroments Options* e clicar na aba *Library*. Na janela que será aberta, clicar no botão de reticências da opção *Library Path*. Dessa vez você deverá selecionar o diretório onde estão os pacotes. Dentre eles temos C:\Zeos\packages\delphi7\build e C:\Zeos\src\component. Assim o *Delphi* pesquisará exatamente nestes locais os arquivos necessários.

Para instalação de componentes da família *Indy* versão 10 no *Delphi*, primeiramente deve-se desinstalar a versão nativa dos componentes *Indy* que vêm instalados no *Delphi* 7. Isso pode ser feito colocando o cd de instalação do *Delphi* e selecionando a opção para desinstalar os componentes *Indy*. A versão utilizada no *WebLimit* foi a *indy10*.1.5_d7, que pode ser adquirida gratuitamente no *site www.indyproject.org.*

Para a instalação e distribuição do sistema já pronto para outras máquinas, foi utilizado o *software* livre *InnoSetup*, que possui assistente para criação de projetos de Instalação e que é configurável por meio de *script.* Possui várias funções, dentre elas a escolha do diretório onde serão armazenados os arquivos, bem como a criação de ícones e outros.

ANEXO A – PLANO DA TELEFÔNICA COM COTA DE CONSUMO

Plano de Consumo - 2005-143 - 3/11/2005

Segue a cota de consumo e o valor de Megas excedentes dos produtos Speedy

SPEEDY HOME

Produto	Velocidade Downlaod	Velocidade Upload	COTA de Consumo	Preço do MByte Excedente	Período de Isenção
Speedy FIT	250 Kbps	128 Kbps	200MBytes	R$ 0,50	**até 31/dez/2.005**
Speedy LIGHT	250 Kbps	128 Kbps	4.000MBytes	R$ 0,10	**até 31/dez/2.005**
Speedy FLEX	500 Kbps	128 Kbps	5.000MBytes	R$ 0,10	até 31/dez/2.006
Speedy POWER	500 Kbps	128 Kbps	10.000MBytes	R$ 0,10	até 31/dez/2.006
Speedy TURBO	1 Mbps	128 Kbps	20.000MBytes	R$ 0,10	até 31/dez/2.006
Speedy NITRO	8 Mbps	512Kbps	60.000MBytes	R$ 0,10	até 31/dez/2.006

SPEEDY CONDOMÍNIO

Produto	Velocidade Downlaod	Velocidade Upload	COTA de Consumo	Preço do MByte Excedente	Período de Isenção
Speedy Condomínio LIGHT	250 Kbps	128 Kbps	4.000MBytes	R$ 0,10	**até 31/dez/2.005**
Speedy Condomínio POWER	500 Kbps	128 Kbps	10.000MBytes	R$ 0,10	até 31/dez/2.006
Speedy Condomínio 750	750 Kbps	128 Kbps	15.000MBytes	R$ 0,10	até 31/dez/2.006
Speedy Condomínio TURBO	1 Mbps	128 Kbps	20.000MBytes	R$ 0,10	até 31/dez/2.006

SPEEDY BUSINESS

Produto	Velocidade Downlaod	Velocidade Upload	COTA de Consumo	Preço do MByte Excedente	Período de Isenção
Business 350K	350 Kbps	128 Kbps	20.000MB	R$ 0,10	até 31/dez/2.006
Business 550K	550 Kbps	128 Kbps	30.000MB	R$ 0,10	até 31/dez/2.006
Business 800K	800 Kbps	128 Kbps	40.000MB	R$ 0,10	até 31/dez/2.006
Business 1.2M	1.2 Mbps	128 Kbps	60.000MB	R$ 0,10	até 31/dez/2.006
Business 2M	2 Mbps	300 Kbps	80.000MB	R$ 0,10	até 31/dez/2.006
Business Horas*	350 Kbps	128 Kbps	Não Há	Não Há	Não Há

***O Produto não possui cota, pois é cobrado por hora**

Fonte: www.abusar.org