

INSTITUTO FEDERAL DE MINAS GERAIS

CAMPUS SÃO JOÃO EVANGELISTA

CLEITON DA SILVA COSTA

ROSIANE PATRÍCIA FLOR DA SILVA

WÁRLLEY JÚNIO ANDRADE

**PROJETO DE GERENCIAMENTO E ECONOMIA NO TRÁFEGO DE
INTERNET NO CAMPUS SÃO JOÃO EVANGELISTA**

São João Evangelista

2013

CLEITON DA SILVA COSTA
ROSIANE PATRÍCIA FLOR DA SILVA
WÁRLLEY JÚNIO ANDRADE

PROJETO DE GERENCIAMENTO E ECONOMIA NO TRÁFEGO DE
***INTERNET* NO CAMPUS SÃO JOÃO EVANGELISTA**

Monografia apresentada ao Curso de Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus São João Evangelista, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Ricardo Bittencourt Pimentel
Coorientador: Fernando Henriques Mafra
Coorientador: Chirlando Weliton de Souza Rocha

São João Evangelista

2013

FICHA CATALOGRÁFICA

Elaborada pelo Serviço Técnico da Biblioteca do
Instituto Federal Minas Gerais – Campus São João Evangelista

C837p COSTA, Cleiton da Silva, 1992 -

Projeto de gerenciamento e economia no tráfego de internet no Campus São João Evangelista./ Cleiton da Silva Costa; Rosiane Patrícia Flor da Silva; Wárley Júnio Andrade. São João Evangelista, MG: IFMG – Campus São João Evangelista, 2013.
58 p.: il.

Trabalho de Conclusão de Curso - TCC (graduação) apresentado ao Instituto Federal Minas Gerais – Campus São João Evangelista – IFMG, Curso de Bacharelado em Sistemas de Informação, 2013.

Orientador: Prof. Esp. Ricardo Bittencourt Pimentel
Coorientadores: Prof. Esp. Fernando Henriques Mafra; Prof. Esp. Chirlando Weliton de Souza Rocha

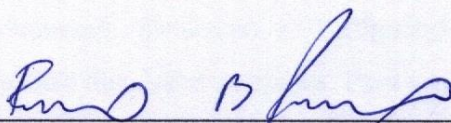
1. Rede de computadores. 2. Internet. 3. Protocolos. I. Instituto Federal Minas Gerais – Campus São João Evangelista. Curso de Bacharelado em Sistemas de Informação. II. Título.

CDD 004.6

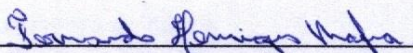
**CLEITON DA SILVA COSTA
ROSIANE PATRÍCIA FLOR DA SILVA
WÁRLLEY JÚNIO ANDRADE**

**PROJETO DE GERENCIAMENTO E ECONOMIA NO TRÁFEGO DE
INTERNET NO CAMPUS SÃO JOÃO EVANGELISTA**

Monografia apresentada ao Curso de Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus São João Evangelista, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.



Ricardo Bittencourt Pimentel (Orientador) – IFMG-SJE



Fernando Henrique Mafra (Coorientador) - IFMG-SJE



Chirlando Weliton de Souza Rocha (Coorientador) - IFMG-SJE

*A Deus por guiar todos os nossos passos
nessa jornada, também a todos que nos
deram apoio, principalmente nossas
familiares e amigos que contribuíram para
a conclusão de mais uma etapa importante
nas vidas.*

São João Evangelista

2013

*A Deus por guiar todos os nossos passos
nessa jornada, também a todos que nos
deram apoio, principalmente nossos
familiares e amigos que contribuíram para
a conclusão de mais uma etapa importante
em nossas vidas.*

RESUMO

A *Internet* vem se popularizando cada vez mais, com isso ocorre um grande aumento do volume de dados trafegados, que se dá em uma grande parte através da *World Wide Web*. Esse aumento trouxe consequências como a sobrecarga nos servidores, aumento do consumo de banda e tempo de latência. Partindo desse ponto, esse projeto tem como objetivo apresentar uma proposta de gerenciamento e economia no tráfego de *Internet* no Instituto Federal de Minas Gerais - Campus São João Evangelista. Para isso utilizou-se o *software Myauth Gateway 3* em conjunto com o *plug-in SuperCache*. O uso desse sistema de cache tem crescido bastante, colocando-o entre as principais ferramentas da área que tem como principal característica fazer *caching* de conteúdo de dados na *Web* com o propósito de gerar economia no tráfego de *Internet*. Para alcançar o objetivo proposto realizou-se o estudo dos materiais necessários ao desenvolvimento, instalação e configuração das ferramentas necessárias, realização de testes e a análise dos dados coletados. Para análise dos dados coletados adotou-se a métrica *Byte Hit Ratio*, pra medir a performance e eficiência do *web caching*. Os resultados foram satisfatórios onde demonstrou-se com a análise realizada que o *software* foi eficiente e proveu uma economia do *link* de *Internet* de 67,22%. Portanto é evidente a eficiência desse sistema de cache para gerar economia no *link* de *Internet* e um melhor aproveitamento na largura de banda.

Palavras-chave: Cache *web*. SuperCache. *Myauth Gateway 3*.

ABSTRACT

The Internet is increasingly becoming folk, it is a large increase in the bulk of data traffic, which occurs in a large part through the World Wide Web this increase brought consequences as overload the servers, increased bandwidth and latency time. From this point, this project aims to present a proposal for management and economics in internet traffic at the Federal Institute of Minas Gerais - Campus São João Evangelista. For this we used the software MyAuth Gateway 3 in conjunction with the plug-SuperCache, its use has grown considerably, placing it among the main tools of the area that aims to locally store objects of the Internet with the purpose of improving bandwidth usage. To achieve the proposed objective conducted the study materials required for the development, installation and configuration of the necessary tools, testing and data analysis. For data analysis we adopted the metric Byte Hit Ratio (BHR), to measure the performance and efficiency of web caching. The results were satisfactory where it was shown to the analysis that the software was efficient and the economy provided an Internet link of more than 67%. Therefore it is evident that the efficiency of proxy server to generate savings in internet link and provide an increase in capacity of data traffic on the network.

Keywords: Web Caching. SuperCache. MyAuth Gateway 3.

LISTA DE FIGURAS

Figura 1: Métrica BHR	28
Figura 2: Diagrama de rede do cenário A	31
Figura 3: Diagrama de rede do cenário B	36

LISTA DE TABELAS

TABELA 1: Configurações de rede no Windows Server 2008	30
TABELA 2: Configurações de rede no MyAuth Gateway 3	33

LISTA DE GRÁFICOS

GRÁFICO 1: Tráfego Total de Internet.....	32
GRÁFICO 2: Tráfego Total dos Dados	37
GRÁFICO 3: Estatísticas de Cache	37
GRÁFICO 4: Tráfego de rede do Myauth Gateway 3	38
GRÁFICO 5: Tráfego no link de Internet	40
GRÁFICO 6: Volume de tráfego de Entrada/Saída	41
GRÁFICO 7: Análise de Arquivos servidos	42
GRÁFICO 8: Estatísticas de Cache	43

LISTA DE SIGLAS

AUFS - Asynchronous Unix File System
CARP - Cache Array Routinng Protocol
CERN - Centro Europeu de Pesquisas Nucleares
COSS - Cyclic Object Storage System
DISKD - Daemon próprio para tarefas de entrada/saída
DNS - Domain Name System
ECO - Modelo Econômico
GB - Gigabyte
Ghz - Gigahertz
HEAP GDSF – Greed Dual Size Frequency
HEAP LFU-DA - Least Frequently Used With Dynamic Aging
HTCP - Hypertext Control Protocol
HTML - Hipertext Markup Language
HTTP - Hypertext Transfer Protocol
ICP - Internet Cache Protocol
IP - Internet Protocol
LAN - Local Area Network
LFU- Least Frequently Used
LRU - Least Recently Used
MAN - Metropolitan Area Networks
MB - Megabyte
RAM - Random Access Memory
RTP – Real-Time Transport Protocol
SO- Sistema Operacional
TI - Tecnologia da Informação
UFS - Unix File System
URL - Uniform Resource Locator
WCCP - Web Cache Control Protocol
WWW - World Wide Web
XML - Extensible Markup Language

LISTA DE ABREVIATURAS

GRAF. - Gráfico

IFMG-SJE - Instituto Federal de Minas Gerais Campus São João Evangelista

TAB. - Tabela

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Estrutura do Trabalho	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	World Wide Web	17
2.2	Interações Cliente/Servidor	17
2.3	Domain Name System	18
2.4	Hypertext Transfer Protocol	18
2.5	URLs Estáticas e Dinâmicas	18
2.6	Web Caching	19
2.7	Tipos de Proxy	19
2.7.1	Proxy Web	19
2.7.2	Proxy Transparente	20
2.7.3	Proxy Reverso	20
2.8	Servidor de Proxy	21
2.9	Ferramentas para Implantação de web Caching	21
2.9.1	Squid	21
2.9.2	MyAuth Gateway 3	22
2.9.3	SuperCache	22
2.9.3.1	Recursos do SuperCache	22
2.9.4	Varnish	23
2.9.5	Microsoft ISA Server	23
3	REVISÃO BIBLIOGRÁFICA	25
4	METODOLOGIA	27
5	PROCEDIMENTOS	29
5.1	Cenário A	29
5.2	Cenário B	32
6	ANÁLISE DOS RESULTADOS	39
6.1	Análise de tráfego de Internet	39
6.2	Análise do volume total de tráfego de Entrada/Saída	40
6.3	Análise de arquivos servidos	41
6.4	Análise de Estatísticas de Cache	42
7	CONSIDERAÇÕES FINAIS	44
	REFERÊNCIAS	46
	APÊNDICE A- CÓDIGO DO APLICATIVO JAVA	49

APÊNDICE B - TELAS DE INSTALAÇÃO DO *WINDOWS SERVER 2008 R2 ENTERPRISE* 52

APÊNDICE C – TELAS DE INSTALAÇÃO DO ASSISTENTE *NETTRAFIC* 56

1 INTRODUÇÃO

A *Internet* vem se popularizando cada vez mais, além disso, a facilidade de acesso a equipamentos móveis e o aumento do número de aplicações *web*, tem aumentado bastante o tráfego de dados na rede, que ocorre em grande parte através da WWW (*World Wide Web*). Esse aumento trouxe consequências como a sobrecarga nos servidores, aumento do consumo de banda e tempo de latência¹.

Para tais consequências, existe uma solução que ameniza essa crescente de volume de dados trafegados na rede, a solução em questão seria o uso de armazenamento em cache também conhecido como *web caching*. Ele tem por finalidade armazenar temporariamente uma cópia de páginas HTML (*HyperText Markup Language*), imagens e vídeos, conhecidos como objetos da *Internet*. Objetos esses, que, posteriormente atenderão as solicitações dos mesmos, usando cópias que estarão armazenadas, localmente, não havendo necessidade de realizar novas requisições ao servidor de origem.

A primeira ideia de *web caching* surgiu com os servidores de *Proxy*, que armazenam as requisições processadas antes de enviá-las aos clientes. Uma vez armazenadas, essas cópias poderiam ser passadas sempre que solicitadas novamente, sem necessidade de requisição ao servidor de origem. Esta abordagem foi bem sucedida, pois, servidores de *Proxy*, normalmente, pertencem a redes de Organizações, cujos clientes possuem interesses comuns (STEINMACHER, 2004).

São evidentes os benefícios alcançados com o uso eficiente de uma implantação de sistema de cache. Entre os mais importantes incluem, maior satisfação dos usuários ao acessarem os recursos da *Internet*, devido ao baixo tempo de resposta, melhor aproveitamento da banda de *Internet* por diminuir o número de documentos solicitados e menor número de solicitações ao servidor de origem, minimizando a chance de uma sobrecarga no servidor remoto.

Sieminski (2005) destaca que, “um ponto crucial quanto à usabilidade da *Web* é a latência, que pode ser definida como o tempo entre a solicitação de uma página e sua total renderização no navegador e apresentação ao usuário”.

¹ Latência é o atraso de tempo entre o momento que um evento iniciou e o momento que os efeitos iniciam. A palavra deriva do fato que durante o período de latência, os efeitos do evento estão latentes, ou seja, potenciais ou não ainda observados. O significado de latência pode variar dependendo do domínio do problema (CHESHIRE, 2006).

O ser humano realiza cerca de seis operações por minuto, e uma demora de resposta acima de 10 segundos causaria desconforto e desinteresse. Outros estudos, focados no comportamento durante o uso da *Web*, indicam que a latência máxima tolerável é de 8 segundos (ZONA RESEARCH, 1999).

Partindo desse ponto, esse trabalho visa apresentar uma proposta de gerenciamento e economia no tráfego de *Internet* no Instituto Federal de Minas Gerais - campus São João evangelista (IFMG-SJE), onde um dos problemas enfrentados pelo meio acadêmico ao acessar e utilizar os recursos da rede é a latência. Isso é devido ao *link* de *Internet* não atender a quantidade de requisições de dados solicitados pelos usuários que utilizam a rede.

Atualmente, existem ferramentas *open source* e comerciais destinadas a implantação de *web caching*. Uma delas é o *plug-in* SuperCache que opera em conjunto com o sistema *MyAuth Gateway 3*, *softwares* comerciais exclusivos da empresa TMSOft. O SuperCache é utilizado para armazenamento de conteúdo dinâmico como: vídeos, atualizações de *softwares*, *downloads* de *softwares* essenciais, dentre outros.

Entretanto, em redes com grande volume de dados trafegados, existem a necessidade de armazenamento de arquivos em cache, sendo de grande valia o uso de uma poderosa ferramenta de cache que permite armazenar conteúdos HTTP com URLs (*Uniform Resource Locator*, Localizador Padrão de Recursos) estáticas e dinâmicas.

Dessa forma espera-se os melhores resultados ao longo do desenvolvimento desse projeto. Acredita-se que esse trabalho será de grande relevância e utilidade no Campus e em outras organizações, onde os benefícios esperados serão: proporcionar economia no tráfego de *Internet* obtendo melhor aproveitamento da banda de *Internet*, auxiliar os profissionais de TI na escolha de uma ferramenta para gerar economia no tráfego de *Internet* e conseqüentemente obter maior satisfação dos usuários ao estarem acessando os recursos da rede devido à redução da latência.

Esta pesquisa tem como objetivo geral apresentar uma proposta de gerenciamento e economia de tráfego de *Internet*, utilizando o *software* SuperCache, para realização e gerenciamento de *web caching* a fim de obter economia no tráfego de *Internet*, e contempla como objetivos específicos: a) estudar e compreender os mecanismos do funcionamento de cache, b) estudar as ferramentas *MyAuth Gateway 3* e SuperCache, c) efetuar a instalação dos *softwares* *MyAuth Gateway 3* e SuperCache, d) efetuar a configuração do SuperCache para funcionamento correto dos serviços de cache, e) efetuar a coleta do volume do tráfego do cenário sem utilização do sistema de cache, f) realizar a coleta do volume de tráfego após a implantação do SuperCache, g) analisar os dados coletados nos testes.

1.1 Estrutura do Trabalho

Este capítulo apresentou o contexto no qual esta pesquisa está inserido, a justificativa, propósito e motivação para o seu desenvolvimento e os objetivos a serem alcançados, o capítulo 2 introduz conceitos relacionados aos aspectos históricos e componentes básicos da *web* além de ferramentas para a implantação de *web caching*, o capítulo 3 apresenta-se trabalhos relacionados nas áreas de servidores de *Proxy* e *web caching*, o capítulo 4 apresenta-se o tipo de abordagem e as quatro fases da metodologia utilizada no trabalho, no capítulo 5 descreve-se os procedimentos utilizados para realização dos testes, os cenários construídos bem como as coletas dos dados, no capítulo 6 apresenta-se e discutem-se os resultados obtidos nos testes e as conclusões gerais sobre o trabalho de pesquisa são apresentadas no capítulo 7.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo trata da fundamentação teórica que serve como referencial para o trabalho. A primeira seção do capítulo trata-se da elucidação sobre *World Wide Web*. Na segunda realizou-se uma explicação sobre a interação cliente/servidor. Na terceira e quarta seção foram apresentados conceitos dos protocolos de *Internet* DNS e HTTP respectivamente. Na quinta seção descrevem-se a diferenciação entre URLs estáticas e dinâmicas. Na sexta seção apresentam-se conceitos de cache e *web caching*. Na sétima e oitava seção descreve-se uma explicação sobre os principais tipos de *Proxies* e Servidor *Proxy*. A última seção dedicou-se a descrever as principais características das ferramentas para implantação de *web caching*.

2.1 World Wide Web

A *World Wide Web*, também conhecido como *Web*, é um repositório de informação espalhado ao redor do mundo. É uma plataforma de comunicação que combina flexibilidade, portabilidade e características amigáveis que a tornam distinta de outros serviços encontrados na *Internet*. (TANENBAUM, 2003, p.59).

Segundo Forouzan (2006), atualmente a, *Web* é um serviço cliente-servidor distribuído, onde um cliente usando um *browser* pode acessar um serviço hospedado em um servidor. O serviço fornecido encontra-se distribuído em muitas localizações chamadas *websites*.

A *Web* é um dos maiores sistemas de informação distribuídos da *Internet*. Desta forma vem tomando proporções cada vez maiores, o que tem proporcionado crescimento tanto em número de usuário como de aplicações.

2.2 Interações Cliente/Servidor

Arquitetura Cliente/Servidor é uma arquitetura dividida em duas partes, uma pela distribuição e manutenção dos dados (Servidor) e a outra pela obtenção da informação (Cliente). Segundo Forouzan 2006 a navegação *Web* utiliza o paradigma cliente/servidor. Quando dada a *Uniform Resource Locator* (URL) de um documento, um navegador se torna um cliente, que contata ao servidor especificado pela URL solicitando o documento. O servidor por sua vez retorna o documento ao computador cliente.

2.3 Domain Name System

Segundo Tanenbaum (2003), o DNS (*Domain Name System*) é um protocolo de suporte que cuida da nomenclatura da *Internet* e permite que as aplicações funcionem. A essência do DNS é a criação de um esquema hierárquico de atribuição de nomes baseado no domínio e de um sistema de bancos de dados distribuídos para implementar esse esquema de nomenclatura. Ele é usado principalmente para mapear nomes de *hosts* e destinos de mensagens de correio eletrônico em endereços IP (*Internet Protocol*).

De uma forma mais simplificada o DNS é um serviço que traduz nomes de domínio em endereços IP. O seu funcionamento consiste da seguinte forma: ao acessar o endereço www.google.com, o cliente estaria enviando a requisição ao servidor DNS, através disso é feita uma busca no servidor DNS que verificará qual IP está associado ao domínio. Que por sua vez encaminha a resposta ao cliente (exemplo: www.google.com.br está associado a 74.125.234.223).

2.4 Hypertext Transfer Protocol

Segundo Tanenbaum (2003) HTTP (*Hypertext Transfer Protocol*) é um protocolo de comunicação entre sistemas de informações que permite a *web servers* e *browser* enviarem e receberem dados através da *web*.

Segundo Forouzan (2006), o funcionamento do HTTP se consiste da seguinte forma: um cliente envia um pedido, na forma de uma mensagem, ao servidor. O servidor envia a resposta, também na forma de uma mensagem, ao cliente. As mensagens pedidas e resposta transportam dados na forma de documento.

2.5 URLs Estáticas e Dinâmicas

Uma URL (*Uniform Resource Locator*) pode ser classificada em duas modalidades sendo elas estáticas ou dinâmicas.

Segundo Ivo (2012) as URLs estáticas se caracterizam por possuírem sempre o mesmo endereço e utilizarem diretórios para separar os arquivos que a compõem de modo que a navegação fique mais organizada. Tais diretórios indicam o conteúdo da página *web*, isto pode ser observado no exemplo a seguir: <http://dominio.com.br/produto/geladeira/>.

As URLs dinâmicas por sua vez são caracterizadas por mudarem frequentemente

devido as atualizações constantes. Assim sendo a página é gerada de acordo com o tipo de pedido feito pelo usuário ou depois que executa algum tipo de ação. Um exemplo seria a execução de uma pesquisa utilizando o sistema de busca de um *site*.

Este tipo de URL é exibido de forma criptografada não indicando nenhuma característica das informações buscadas no site. Geralmente as URLs dinâmicas são constituídas da seguinte forma: `http://dominio.com.br/boleto.php?cid=7&pid=9`, onde os parâmetros `cid` e `pid` presentes na URL podem ser obtidos através da base de dados do *site*.

2.6 Web Caching

O conceito de cache surgiu na computação no sentido de melhorar o desempenho através de cópias de informações em locais que facilitassem seu acesso.

Segundo Steinmacher (2004) é comprovada na prática a eficiência do *caching* em memória e sistemas de arquivo. Com o grande crescimento do tráfego na *web*, percebeu-se que a estratégia de *caching* seria uma técnica que possivelmente reduziria a latência da rede. Daí surgiu o termo *Web Caching*, que nada mais é que o armazenamento de cópias de informações de provável acesso em um futuro próximo em locais onde usuários as acessem de forma rápida e fácil, da mesma forma que caches de memória e disco, que mantêm os dados mais acessados em uma área específica para posterior recuperação, o *web caching* armazena os objetos acessados na *Internet*.

2.7 Tipos de Proxy

Proxy é o termo utilizado para definir os intermediários entre o usuário e seu servidor. E por isso desempenha a função de conexão do computador (local) à rede externa (*Internet*). De acordo com Barwinski “Como os endereços locais do computador não são válidos para acessos externos, cabe ao *Proxy* enviar a solicitação do endereço local para o servidor, traduzindo e repassando-a para o seu computador.” (BARWINSKI 2008).

2.7.1 Proxy Web

O *Proxy* funciona como um cache de páginas e arquivos, armazenando informações já acessadas. Segundo Morimoto quando alguém acessa uma página que já foi carregada, o *Proxy* envia os dados que guardou no cache, sem precisar acessar a mesma página

repetidamente. Isso acaba economizando bastante banda, tornando o acesso mais rápido (MORIMOTO 2008).

“A aplicação mais comum é o *Proxy web*. Isto é, determina o cacheamento de páginas *web* e arquivos disponíveis em servidores *web* remotos, permitindo o acesso mais rápido e confiável de clientes da rede local.” (PONTES, HIRARA, HONÓRIO, 2008).

2.7.2 *Proxy Transparente*

Um *Proxy* transparente é uma implementação de *Proxy* sem que a maioria dos usuários comuns saiba da existência de tal ferramenta. Nesse caso, o *Proxy* funciona mesmo que os usuários não o configurem em cada máquina, ou seja, independente da configuração local, o que evita que os usuários avançados o burlem (ANDRADE; RICARDO, 2009).

Através da utilização do *Proxy* transparente é dispensada a configuração manual de *Proxy* para o navegador de *Internet*, uma vez que esta é feita automaticamente. Onde quando o cliente faz uma solicitação ao servidor para conectar-se a um *site*, essa é transferida para o servidor *Proxy* que a captura e a desvia para a porta do *Proxy* que por sua vez se faz passar pelo *site* remoto.

2.7.3 *Proxy Reverso*

Proxy Reverso surgiu da necessidade de aproximar os proxies dos servidores para reduzir a carga sobre eles.

Estes proxies são também chamados de aceleradores, pois o sistema de armazenamento em cache fica à frente de um ou mais servidores *web*, interceptando solicitações e agindo como um *Proxy*. Os documentos armazenados em cache são fornecidos a uma maior velocidade, enquanto os que não estiverem em cache (conteúdo dinâmico/personalizado) são solicitados aos servidores *web* de origem quando necessário (STEINMACHER, 2004).

Um *Proxy* reverso é um *Proxy* instalado nas imediações de um ou mais servidores. Tipicamente, os proxies reversos são utilizados na frente de um *Webserver*, ou seja, colocado como o primeiro servidor que a *Internet* irá enxergar. Todas as conexões vindas da *internet* são endereçadas para um ou mais *Webservers* que são roteados pelo servidor *Proxy* (Teixeira, 2006).

O *Proxy* Reverso pode ser entendido também como *Cluster web*, ou seja, uma máquina principal recebe a solicitação e essa máquina tem por objetivo solicitar o serviço de outra máquina. É uma troca de processamento, sendo que o processamento estará sendo

dividido, ou seja, uma estará postando as informações e a outra estará fornecendo o serviço (RICARDO, 2004).

2.8 Servidor de Proxy

Um servidor de *Proxy* é um computador que mantém um histórico das respostas aos pedidos solicitados em uma rede. Na presença de um servidor de *Proxy*, primeiramente, o cliente HTTP envia o pedido ao servidor de *Proxy*. O servidor de *Proxy* verifica no cache local² a procura de respostas anteriores à solicitação atual. Se não houver respostas no cache correspondente ao pedido atual, o servidor de *Proxy* retransmite o pedido ao servidor correspondente na *web*. Todas as respostas são recebidas, processadas e armazenadas para uso futuro de outros clientes, caso tenha sido aprovadas, pelo servidor de *Proxy*.

Um servidor de *Proxy* reduz a carga do servidor original, diminui o tráfego e melhora a latência da rede. Entretanto, os clientes devem estar configurados para acessar um servidor de *Proxy* ao invés do servidor alvo (FOROUZAN, 2006).

2.9 Ferramentas para Implantação de web Caching

Atualmente existem diferentes ferramentas *open source*³ e comerciais destinadas a implantação de *web caching*. Entre os vários existentes destaca-se o *Squid*, *Varnish*, *Microsoft ISA Server* e *SuperCache*.

2.9.1 Squid

Foi originado de um projeto denominado *Harvest* entre o governo americano e a Universidade de Colorado.

Atualmente ele é o *Proxy open source* mais popular e mais usado como controle de conteúdo e implantação de cache *web*. É disponibilizado por quase todas as distribuições *Linux* e fornece todas as funcionalidades de um *Proxy* comum (REIS, RAIMUNDO, CARMO, 2006).

Segundo Morimoto (2008), o *Squid* é um servidor *Proxy*, que serve como um

² Está no mesmo servidor.

³ Programa de código aberto.

intermediário entre a rede local e a *Internet*. Dependendo da configuração, ele pode se limitar a aperfeiçoar a conexão, utilizando um sistema de cache para reduzir o volume de dados transferidos, pode ser usado para logar os acessos, permitindo que você tenha acesso a um relatório detalhado do que foi acessado, ou pode ser usado para efetivamente bloquear *sites* indesejados, bloquear determinados tipos de arquivos, restringir o acesso baseado no horário e assim por diante.

2.9.2 MyAuth Gateway 3

MyAuth Gateway 3 é um *software* comercial desenvolvido pela empresa TMSoft destinado à administração de usuários para redes de provedores avançados, permitindo gerenciamento completo de clientes, usuários de acesso, grupos de controle de banda, acesso administrativo com níveis de acesso e registros de ações dos técnicos, controle completo de suporte e *ticket*, central de estatísticas e sistema de cobrança via boleto bancário dos principais bancos nacionais. (COMO..., 2013).

Segundo o manual da TMSoft o sistema *MyAuth Gateway 3* permite total integração com concentradores de acesso para grandes provedores: Cisco, *Mikrotik* e outros compatíveis com autenticação e contabilidade RADIUS. O *Software* funciona baseado no conceito de *gateway* de captura. Nessa técnica, um *host* localizado em uma rede atrás do *MyAuth Gateway 3* necessita se autenticar para obter acesso. (TMSOFT SOLUÇÕES, 2013).

Para o funcionamento e estabilidade o *MyAuth Gateway 3* tem como pré-requisito instalação em uma máquina com a plataforma *Linux Slackware* 12.0 ou 12.2, devido a quantidade de recursos anexos ao *kernel* e pacotes do sistema.

2.9.3 SuperCache

O SuperCache é um *software* com uma nova tecnologia desenvolvido com técnicas para realização e gerenciamento de cache exclusivo da TMSoft, utilizado para armazenamento de conteúdo com URL dinâmica como: vídeos, atualizações de *softwares*, *downloads* de *softwares* essenciais, dentre outros. (TMSOFT SOLUÇÕES, 2013)

2.9.3.1 Recursos do SuperCache

O *plug-in* SuperCache possui alguns recursos relevantes quanto ao seu uso que são:

- a) Permite cache de URLs dinâmicas (*Windows Update*, atualizações de *softwares* e antivírus).
- b) Possibilita Cache-full de arquivos em cache, ou seja, quando um arquivo constar no cache local, ele é enviado para o cliente sem controle de banda, dando ao mesmo uma impressão de altíssima velocidade (este recurso pode ser desativado ou controlado).
- c) Admite o compartilhamento de fluxo: se um mesmo arquivo for solicitado ao mesmo tempo por dois usuários, apenas um fluxo de *Internet* é consumido, em vez de dois, economizando consideravelmente o consumo de banda. Suporta armazenamento de milhões de arquivos sem acrescentar lentidão ao sistema, pois permitem montar outras partições de HD dedicado para armazenar separadamente outros tipos de objetos.
- d) Possibilita a visualização de gráficos que mostram estatísticas de tráfego e economia de banda.

2.9.4 Varnish

O Varnish é um servidor *daemon*⁴ rico em funcionalidades e que foi desenvolvido pelo arquiteto Poul Henning Kamp, com o objetivo de ser um poderoso proxy/cache reverso. A primeira versão estável foi liberada em 2006 e nomeada de 1.0. (SILVA, 2009).

Varnish é um software *open source* distribuído sobre a licença BSD (*Berkeley Software Distribution*), funciona como um acelerador de requisições HTTP, deixando o *site* muito mais rápido, pois é um serviço que fica entre o usuário e o servidor *web*, fazendo uma análise dos conteúdos mais acessados, armazenando um cache e efetuando a gerência das páginas acessadas. (VARNISH, 2003).

2.9.5 Microsoft ISA Server

A *Microsoft Internet Security and Acceleration (ISA) Server* é o *gateway* integrado de segurança desenvolvido pela empresa *Microsoft* que ajuda a proteger o ambiente de Tecnologia da Informação de organizações contra as ameaças vindas da *Internet*, ao mesmo

⁴ Programa que fica rodando continuamente em uma máquina, esperando uma requisição para executar alguma tarefa, serviço.

tempo em que fornece aos seus usuários um acesso remoto seguro e mais rápido aos dados e aplicativos. (VISÃO, 2006).

Uma das funções presentes no *ISA Server* é o *cache web* de alto desempenho. A configuração de cache do usuário é realizada através da configuração do serviço *web*. Dentre suas características permite a configuração do tamanho da memória cache por disco rígido e do agendamento de tarefas de armazenamento em cache. (SERWIN, 2013).

3 REVISÃO BIBLIOGRÁFICA

O objetivo deste capítulo é a apresentação de trabalhos relacionados às áreas estudadas, visando descrever o estado da arte tanto as áreas de servidores de *Proxy* quanto de *web caching*.

Silva (2009) realizou um estudo comparativo em termos de performance entre o *Squid* e o *Varnish* como aceleradores HTTP. O trabalho apresentado avaliou a desempenho desses dois servidores *Proxy* de alto desempenho, utilizando o mecanismo de *caching* baseado em expiração para que poucas requisições fossem passadas ao servidor *web*. A análise teve como métricas aplicadas aos aceleradores HTTP, o número de respostas HTTP por segundo e o tempo médio de conclusão de respostas. Silva (2009) deixa explícito que, no critério de desempenho, ambas as soluções tem resultado similar.

Pontes, Hirata e Honório (2008) escreveram um artigo sobre segurança e aceleração de *Internet* através da utilização de *Proxy servers* para manutenção de *web caches*. O artigo aborda os protocolos ICP (*Internet Cache Protocol*), HTCP (*Hypertext Control Protocol*), WCCP (*web Cache Control Protocol*) e CARP (*Cache Array Routinng Protocol*), que são protocolos proprietários que, em implementações conjuntas a servidores *Proxy-cache*, constituem uma solução muito interessante para *Proxy* transparente. O artigo mostra que o uso de servidores *Proxy-cache* contribui muito para a economia da banda, disponibilizando o serviço solicitado pelos usuários mais rapidamente.

Reddy (2005) em seu seminário relata sobre a importância de se utilizar *caching web*, onde descreve propriedades desejáveis de cache *web*. Reddy deixa evidente que um dos principais desafios de cache *web* é realizar o cache de dados dinâmicos e que *Caching web* é reconhecido por ser uma das técnicas mais eficazes para aliviar o gargalo dos servidores e reduzir tráfego de rede, assim, minimizar a latência de acesso do usuário.

Em Dodonov (2004) é apresentado e discutido um mecanismo integrado de cache e *prefetching* para sistemas de entrada e saída de alto desempenho. Onde foram avaliados os algoritmos implementados e a influência da escolha das políticas adequadas no funcionamento de aplicações do usuário. Essa avaliação resultou em uma significativa diminuição da latência nas operações de leitura. Além disso, foram realizados estudos de casos e testes de desempenho que possibilitaram determinar os fatores que limitam a utilização dos mecanismos de cache e *prefetching*, tais como o tempo decorrido entre requisições de entrada e saída e padrões de acesso utilizados pelas aplicações.

Em Barbosa e Martins (2002) é proposto um método para análise de desempenho de

estruturas hierárquicas de servidores *web caching* seguindo um modelo de experimentação em laboratório, onde também é apresentada a validação do método proposto aplicando-o a um estudo de caso real. O método proposto em estudo de caso real teve sucesso, porém a análise de desempenho em laboratório tendeu a ser demorada e complexa.

Em Reis, Raimundo e Carmo (2006) é apresentado um estudo a respeito dos servidores *Proxy*, e do desempenho do *Squid* que se torna crítico quando possui uma grande quantidade de usuários conectados ao *Proxy*. Frente a isso foram feitos estudos dos servidores *Proxy* e dadas algumas soluções a fim de aperfeiçoar o desempenho do *Squid*, em que a solução principal foi a aplicação com o banco de dados *PostgreSQL*⁵, que em conjunto com algumas ferramentas como *Front-end*⁶, *HDparm*⁷ e a compilação de *Kernel*⁸, demonstrou que a sua implementação com o *Squid* pode ser uma boa alternativa, tanto em desempenho quanto na economia de *hardware*.

Portanto é possível observar que, apesar dos trabalhos anteriores, o termo *web caching* é bastante abrangente e existem alguns trabalhos relacionados. Não é encontrada na literatura estudos sobre ferramentas comerciais destinadas a implantação de *web caching* para gerenciamento e economia no tráfego de rede em um ambiente corporativo. Nosso projeto também se difere por abordar um *software* muito utilizado atualmente, o SuperCache em conjunto com *MyAuth Gateway 3*, que se distingue dos demais *softwares* comuns no mercado por se tratar de um *software* que armazena em cache conteúdos com URLs dinâmicas.

⁵ Um sistema *open source* gerenciador de banco de dados objeto relacional.

⁶ Aplicação em que os usuários interagem diretamente com o aplicativo.

⁷ Utilitário de linha de comando do *Linux* e do *Windows* para definir e exibir unidade de disco rígido.

⁸ Tradução do código fonte do núcleo do sistema.

4 METODOLOGIA

Neste capítulo, descreve-se a natureza da pesquisa; o contexto em que a pesquisa ocorreu, os instrumentos utilizados para a realização da coleta de dados e os procedimentos para a análise dos dados.

Segundo Richardson (1989), a abordagem quantitativa corresponde ao método que faz o emprego da quantificação, tanto nas modalidades de coleta de informações, quanto no tratamento dessas através de técnicas estatísticas, desde as mais simples até as mais complexas.

A pesquisa neste trabalho caracteriza-se como uma pesquisa de caráter quantitativa, pois se utilizou da quantificação, tanto na coleta quanto no tratamento das informações, utilizando técnicas estatísticas que permitiram verificar o percentual de economia gerada no tráfego de *Internet*.

Para alcançar os objetivos da pesquisa realizaram-se as seguintes etapas: a) estudo dos materiais necessários ao desenvolvimento do projeto, b) instalação e configuração das ferramentas necessárias, c) realização de testes, d) Análise de dados coletados nos testes.

Para realizar o estudo dos materiais necessários, desenvolveu-se as atividades de estudo de conceitos relacionados, análise de trabalhos relacionados, estudo sobre os mecanismos de funcionamento de cache, estudo do Sistema Operacional *Linux Slackware*, estudo das ferramentas *MyAuth Gateway 3* e *SuperCache*.

Na fase de instalação e configuração das ferramentas, criaram-se dois cenários de infraestrutura de rede onde se realizou a instalação e configuração dos *softwares* para que fosse possível a coleta de dados referentes a pesquisa. Os *softwares* utilizados foram: o Sistema Operacional *Linux Slackware 12.2*, *MyAuth Gateway 3*, *Plug-in SuperCache*, *Windows Server 2008 R2 Enterprise* e o monitor de tráfego *Nettraffic*.

Na etapa de realização dos testes coletaram-se dados referentes ao tráfego de rede de 10 computadores conectados a rede, em um laboratório de informática, ambiente cedido pelo IFMG-SJE. Em ambos os cenários para teste utilizou-se de um aplicativo *desktop* desenvolvido em linguagem de programação Java, para simular os tráfegos de rede de usuários acessando páginas e efetuando o *download* de dez domínios aleatórios, que constam na lista de domínios suportados para armazenamento em cache do *SuperCache*.

A análise de dados coletados contempla o comparativo do volume de tráfego, usando métricas que tiveram como resultado um percentual que estimou e quantificou a economia do tráfego de rede gerado através do *software SuperCache*.

Para melhor análise dos dados coletados utilizou-se a métrica *Byte Hit Ratio* (BHR), uma das maneiras mais adotadas pra medir o desempenho de *web caching*.

“*Byte Hit Ratio* é o percentual de bytes requisitados pelo cliente que foram enviados pelo cache, sem solicitação ao servidor de origem. Sendo h_1, h_2, \dots, h_i o tamanho de cada um dos i objetos enviados após um hit no cache, m_1, m_2, \dots, m_j o tamanho de cada um dos j objetos recuperados do servidor (miss) e $i + j$ o número total de requisições ao cache” (STEINMACHE, 2004)

Tem-se a equação:

Figura 1: Métrica BHR

$$BHR = \frac{\sum h_i}{\sum h_i + \sum m_j}$$

Fonte: Elaborada por STEINMACHER (2004)

Optou-se por apresentar os dados coletados nesta pesquisa através de gráficos, com o propósito de realizar uma discussão sobre os mesmos.

5 PROCEDIMENTOS

Neste capítulo descreveram-se os testes realizados, assim como o ambiente construído para realização dos mesmos e as ferramentas utilizadas para auxiliá-los.

5.1 Cenário A

Nesse cenário houve a instalação e configuração dos *softwares* necessários para a realização dos testes referente à coleta do volume de tráfego da rede deste cenário.

Para analisar a eficiência do SuperCache é preciso analisar o tráfego de *internet* com e sem a utilização do mesmo. E também utilizando a métrica BHR como critério de avaliação. Pois de acordo com SILVA (2012) “a utilização de cache *web* diminui o volume de tráfego de *Internet*”.

Para coletar o volume de tráfego de *Internet* por um determinado período optou-se pelo *software Nettraffic*, por ser um *software* estável, amplamente utilizado por administradores de rede e profissionais de TI para monitoramento do tráfego de rede.

O *Nettraffic* é um *software open source* desenvolvido pela VENEIA. NET que monitora a velocidade atual de *download* e *upload* realiza estatísticas de uso como o total de arquivos baixados, tempo que o seu computador ficou conectado a rede, entre outros métodos. Os dados gerados pelo *Nettraffic* podem ser exportados em formato XML (*Extensible Markup Language*) e posteriormente importados para o *software* de forma que se possa fazer uma revisão de dados. Esse *software* opera sobre a plataforma *Windows*, devido a isso ouve a necessidade de se instalar tal plataforma para a instalação do mesmo. (AMOROSO, 2012).

A plataforma instalada foi o *Windows Server 2008 R2 Enterprise* que está disposta no apêndice A, devido ao desempenho e a quantidade de recursos disponíveis. A instalação do *Nettraffic* foi efetuada logo após a instalação do *Windows Server 2008 R2* e se encontra no apêndice B. O *Nettraffic* foi configurado para monitorar a interface de rede *Wan* do *Windows Server 2008 R2*.

O *Windows Server 2008 R2* é um servidor multifuncional desenvolvido pela Microsoft Corporation para aumentar a confiabilidade e a flexibilidade de infraestruturas de rede. Este sistema operacional é bastante utilizado em organizações provendo uma plataforma mais produtiva para suportar as aplicações, redes e serviços da *web*. Torna-se bastante útil sua utilização em servidores de domínio onde é necessário um controle dos usuários e máquinas na rede. (MICROSOFT, 2008).

A seguir apresentam-se as informações básicas do *Windows Server 2008 R2 Enterprise*:

- Processador: Intel(R) Xeon(R) CPU E5320 1.86 GHz (2 processadores)
- Memória instalada (RAM): 4,00 GB
- Tipo de Sistema Operacional: 64 Bits

Após a instalação dos *softwares* mencionados acima realizou-se a configuração dos mesmos. Utilizaram-se as configurações de rede no *Windows Server 2008* mostradas na (TAB.01), porém, as mesmas são arbitrárias, e devem ser adaptadas para cada tipo de ambiente de rede:

Tabela 1: Configurações de rede no *Windows Server 2008*

Ethernet Lan	Ethernet Wan
Endereço IP: 10.10.0.1	Endereço IP: 172.16.0.148
Máscara de subrede: 255.255.255.0	Máscara de subrede: 255.255.248.0
Servidor DNS preferencial: 10.10.0.1	Gateway Padrão: 172.16.0.6
Servidor DNS alternativo: 8.8.8.8	Servidor DNS preferencial: 172.16.0.1
	Servidor DNS alternativo: 8.8.8.8

Fonte: Elaborado pelos autores

O *link* de *internet* fornecido a rede do *Windows Server 2008* foi de 1Mbps. Devido ao tamanho do *link* de *internet* fornecido foram utilizados 10 computadores de um laboratório de informática disponibilizado pelo IFMG, pois um número maior causaria um maior tempo de latência, podendo até mesmo prejudicar os resultados do teste.

A seguir apresentam-se as informações básicas dos dez computadores utilizados para o teste:

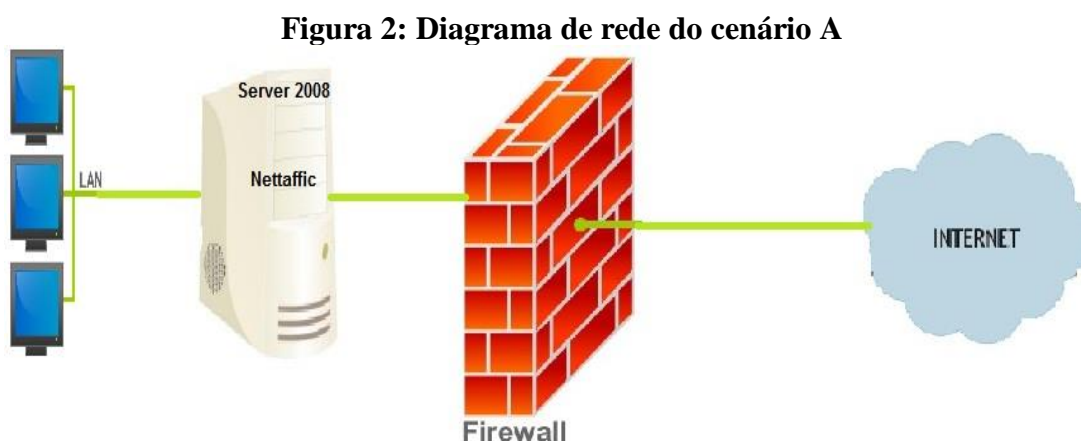
- Processador: Intel Core (TM) i5 CPU 760 2.80GHz
- Memória instalada (RAM): 8,00 GB
- Sistema Operacional: *Windows 7 Professional*
- Tipo de Sistema: 64 Bits

Para coletar o tráfego de *internet* gerado pelos dez computadores através do *Nettraffic*, as interfaces de rede dos mesmos foram configuradas para que o acesso à *Internet* fosse fornecido pelo *Windows Server 2008 R2*. Utilizaram-se as seguintes configurações de rede nos dez computadores para teste:

- Endereço IP: 10.10.0.(range de 10 a 20)
- Máscara de sub-rede: 255.255.255.0
- *Gateway* Padrão: 10.10.0.1

- Servidor DNS preferencial: 8.8.8.8
- Servidor DNS alternativo: 8.8.4.4

Na (FIG.02) é apresentado o ambiente de rede do cenário A de teste. A rede local é formada pelos computadores do laboratório conectados ao *Windows Server 2008*, onde a filtragem de pacotes provenientes da *internet* é feita pelo *Firewall*.



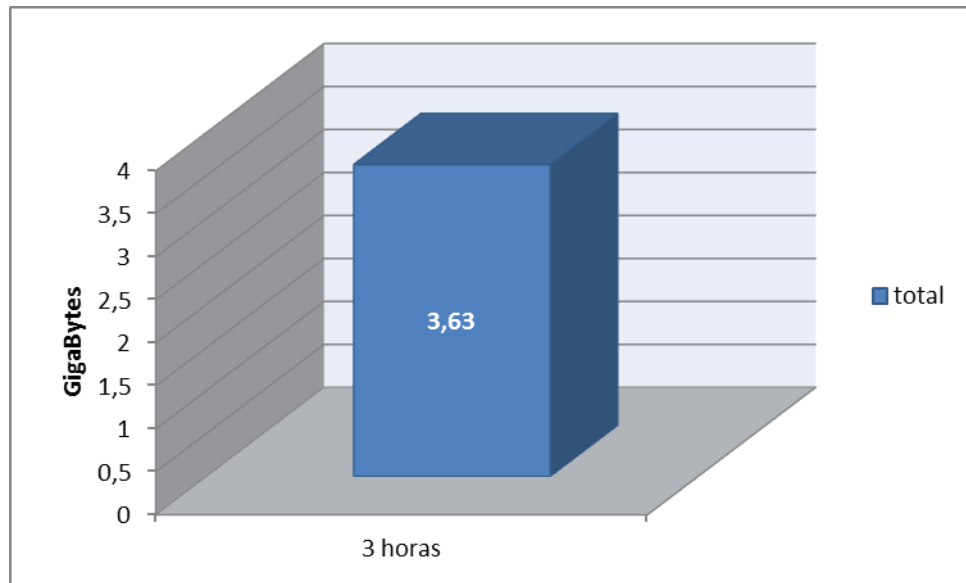
Fonte: Elaborada pelos autores

Para a realização dos testes utilizou-se um aplicativo desktop executável desenvolvido em linguagem de programação Java, que simulou o acesso a páginas *web* e *download* de conteúdos multimídia de dez domínios escolhidos aleatoriamente. Os *sites* utilizados se encontram na lista de domínios para a realização de *cache web* do *plug-in* SuperCache. O código fonte encontra-se no Apêndice A. O único pré-requisito que o aplicativo possui é ter o *Java Runtime Environment (JRE)* instalado no computador. A instalação do JRE não foi necessária nos dez computadores utilizados, pois eles já possuíam tal pacote instalado.

O aplicativo *java.exe* foi inserido e executado nos dez computadores disponibilizados para o teste, onde simulou por durante três horas, o tráfego de *Internet* de usuários acessando somente as páginas listadas no aplicativo. Ao término das três horas coletaram-se através do *Nettraffic* os dados referentes ao tráfego de rede da placa *Ethernet do Windows Server 2008* onde estavam conectados os dez computadores utilizados no teste.

Encontra-se no (GRAF.01) volume de tráfego gerado pelo período de três horas, dos dez computadores conectados ao *Windows Server 2008 R2*:

Gráfico 1: Tráfego Total de Internet



Fonte: Dados da Pesquisa

5.2 Cenário B

Nesse cenário realizou-se a instalação e configuração para a coleta de dados referentes ao teste de cache *web*.

O Sistema operacional utilizado para a instalação do *Software Myauth Gateway 3* foi o *Linux Slackware 12.2*, por ser um sistema operacional que se mantém simples, estável e por fazer parte do pré-requisito de instalação do *Myauth Gateway 3*.

Segundo o manual do fabricante, o *Myauth Gateway 3* não irá executar em nenhuma outra distribuição *Linux* além do *Slackware 12.0* ou *12.2*, devido a quantidade de recursos anexos ao *Kernel*, pacotes do sistema entre outros. (TMSOft, 2013)

A seguir estão as informações básicas do *Linux Slackware 12.2*:

- Processador: Intel(R) Xeon(R) CPU E5320 1.86 GHz (2 processadores)
- Memória instalada (RAM): 4,00 GB
- Tipo de Sistema Operacional: 64 Bits
- HD 1: 80 GB
- HD 2: 500 GB

No cenário de teste B assim como no cenário A, também foi utilizado o *software Nettraffic* para medir o tráfego de rede, portanto a interface de rede do *Linux Slackware* foi configurada para que o acesso à *internet* fosse transmitido pelo *Server 2008*.

Após a instalação e configuração do *Linux Slackware* realizou-se a instalação do

software Myauth Gateway 3. Para dar início a sua instalação, precisou-se efetuar o *download* do mesmo. A seguir é apresentada a seguinte linha de comando que foi utilizada para efetuar o *download* a partir do diretório “/”:

```
root@servidor:~# cd /
```

```
root@servidor:~/# wget http://download.tmssoft.com.br/myauth3.tar.gz
```

Após o *download* do *software* foi necessário descompactá-lo, através da seguinte linha de comando:

```
root@servidor:~/# tar xvzf myauth3.tar.gz
```

Posteriormente realizou-se sua instalação dentro do diretório /admin, através das seguintes linhas de comando:

```
root@servidor:~/# cd /admin
```

```
root@servidor:~/# sh install.sh
```

Após a execução dos comandos listados acima na ordem correta, foi necessário esperar por alguns minutos até que a instalação fosse finalizada.

A licença de uso do *Myauth Gateway 3* foi fornecida pela empresa TMSOft por um curto período de tempo. A licença fornecida inclui também o *plug-in* SuperCache. Para a ativação do *plug-in* foi necessário efetuar a seguinte linha de comando:

```
root@servidor:~#myauth plugin
```

Para o *plug-in* entrar em funcionamento foi necessário reiniciar o *Myauth Gateway 3* através das seguintes linhas de comando:

```
root@servidor:~#myauth stop
```

```
root@servidor:~#myauth start
```

Depois de finalizada sua instalação realizou-se a configuração de rede do mesmo. As configurações utilizadas estão dispostas na (TAB.02).

Tabela 2: Configurações de rede no *Myauth Gateway*

Ethernet Lan	Ethernet Wan
Endereço IP: 192.168.0.20	Endereço IP: 10.10.0.2
Máscara de sub-rede: 255.255.255.0	Máscara de sub-rede: 255.255.255.0
Servidor DNS preferencial: 172.16.0.1	Gateway Padrão: 10.10.0.1
Servidor DNS alternativo: 8.8.4.4	Servidor DNS preferencial: 172.16.0.1
	Servidor DNS alternativo: 8.8.4.4

Fonte: Elaborado pelos autores

Em primeiro instante configurou-se no *plug-in* SuperCache a porcentagem de uso máximo do disco para os arquivos mantidos pelo SuperCache. A porcentagem foi

especificada em 95 %, ou seja, a partir desse valor o *software* inicia a remoção de arquivos em cache. Os algoritmos providos pelo *software* para a limpeza de arquivos em cache são: *Least Recently Used* (LRU), *Least Frequently Used* (LFU) e Modelo Econômico (ECO).

Segundo Oliveira (2004), o LRU remove os arquivos antigos sem acessos recentes. Pois os arquivos que estão frequentemente sendo solicitados tem grande chance de continuarem a serem usados, e devem permanecer em cache.

De acordo com Oliveira (2004), o LFU remove os arquivos em cache com o menor número de acessos. Dessa forma os arquivos com menor número de acessos tem menor chance de serem solicitados por usuários futuramente.

O ECO remove arquivos com a menor economia relativa. Sendo assim arquivos com maior economia se tornam mais favoráveis de serem mantidos em cache, por economizarem mais tráfego de *internet* ao serem requisitados.

O algoritmo de limpeza de cache escolhido para o teste foi o LRU, pois parte do pressuposto que arquivos que não estão sendo acessados devem ser removidos do cache por falta de uso. O limite de tempo sem uso foi especificado em 30 dias, após esse período sempre que necessário haverá a limpeza de disco e os arquivos serão removidos.

Uma das configurações mais importantes em relação ao desempenho do cache e otimização do tráfego de *Internet* é a configuração dos caches, que é onde o SuperCache armazena os objetos, paginas e arquivos já acessados de forma a fornecê-los mais rapidamente quando solicitado novamente.

Em geral, assim como em outros sistemas de cache como o Varnish e Squid entre outros o SuperCache trabalha com dois tipos de cache: O primeiro utiliza parte da memória RAM e o outro utilizando o HD.

O cache realizado utilizando a memória RAM é considerado rápido, porém como o volume de memória RAM no servidor é sempre menor que espaço em disco, é ideal armazenar arquivos pequenos, como páginas HTML e imagens, que quando solicitados e encontrados no cache são entregues para o cliente quase que instantaneamente.

Por outro lado o cache feito através do HD é um pouco mais lento, porém com maior capacidade. Este tipo de cache é amplamente utilizado para armazenar arquivos maiores, como *Downloads* grandes e arquivos do *Windows Update* e atualização de *software* que demoram a serem baixados consumindo bastante banda de *Internet*.

Partindo desse ponto visando uma maior otimização e eficiência do SuperCache realizou-se algumas configurações, dentre as mais importantes destacam-se a política de substituição de objetos em memória e política de substituição de objetos em disco.

As políticas de substituição de objetos no disco que o SuperCache detém são responsáveis por elegerem quais objetos devem ser retirados do cache, para que sejam inseridos novos objetos no disco. Os algoritmos fornecidos pelo SuperCache são: LRU, *Greed Dual Size Frequency* (HEAP GDSF), *Least Frequently Used with Dynamic Aging* (HEAP LFU-DA) e *LRU policy implemented using a heap* (HEAP LRU).

O HEAP GDSF consiste em manter objetos menores e referenciados mais vezes em cache onde possibilita maiores chances de acontecer um *Hit Ratio*.

O HEAP LFU-DA mantém os arquivos mais acessados no cache, não importando o seu tamanho. Este consegue um alto *Byte Hit Ratio* devido a chave do objeto ser calculada levando em conta a frequência da requisição, ou seja quanto mais um objeto é requisitado, maior é a sua chance de permanecer em cache, economizando tráfego de *Internet*.

O HEAP LRU consiste em manter em cache os arquivos abertos recentemente e utiliza a estrutura de dados organizada como árvore binária.

A política de substituição adotada na configuração do SuperCache foi a HEAP LFU-DA devido a recomendação do próprio mantenedor do *software* e por apresentar um maior *Byte Hit Ratio* em relação a frequência da requisição.

Na política de reciclagem em memória que continham os mesmos algoritmos disponibilizados na política de substituição em disco foi utilizada a política HEAP GDSF. Essa política foi adotada devido ao fato de que a memória RAM é uma memória acesso rápido, para isso o melhor é manter arquivos menores e frequentemente utilizados para se obter o máximo de aproveitamento da memória e maximizar a taxa de acerto de *Hit Ratio*.

Para uma melhor eficiência e gerenciabilidade, foram criados dois diretórios, o primeiro chamado “/Cache” e o segundo “/SuperCache”. Eles foram criados na partição hda2 do *Linux Slackware*. Logo após, o SuperCache foi configurado para que os arquivos a serem armazenados fossem direcionados aos dois diretórios criados. A criação do diretório “/Cache” em uma partição separada da partição onde se instalou o *Linux Slackware* nos permitiu ativar o modo turbo no SuperCache. O modo turbo permite executar múltiplos processos simultâneos para aumentar a capacidade de requisições por segundo.

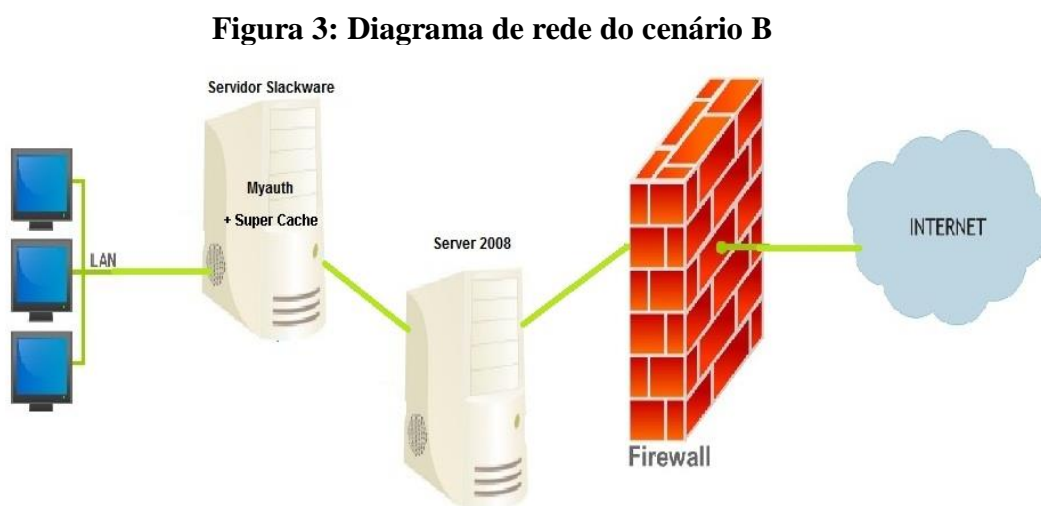
Para a realização dos testes, utilizou-se novamente o aplicativo simulador nos dez computadores onde as interfaces de rede dos computadores foram configuradas para que o acesso a *Internet* fosse realizado através do *software Myauth Gateway*. Para que fosse

liberado o acesso a *Internet* dos dez computadores no *Myauth Gateway*, criou-se um *login* e senha na seção de clientes do *Myauth Gateway 3* para cada um dos computadores.

Utilizaram-se as seguintes configurações de rede nos dez computadores para teste:

- Endereço Ip: 192.168.2.(range de 10 a 20)
- Máscara de sub-rede: 255.255.252.0
- *Gateway* Padrão: 192.168.0.20
- Servidor DNS preferencial: 8.8.8.8
- Servidor DNS alternativo: 8.8.4.4

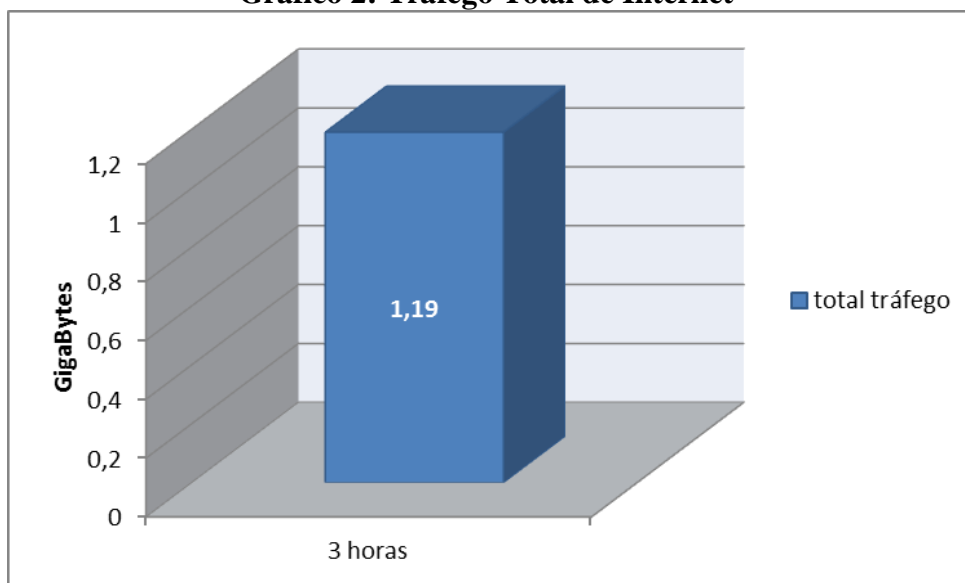
Na (FIG.03) é apresentado o diagrama de rede desse cenário após a instalação do sistema de cache. Onde a rede local é formada pelos computadores do laboratório conectados ao *Myauth Gateway 3* que por sua vez direciona as requisições ao *Windows Server 2008* e a filtragem de pacotes oriundos da *Internet* é feita pelo *Firewall*.



Fonte: Elaborado pelos autores

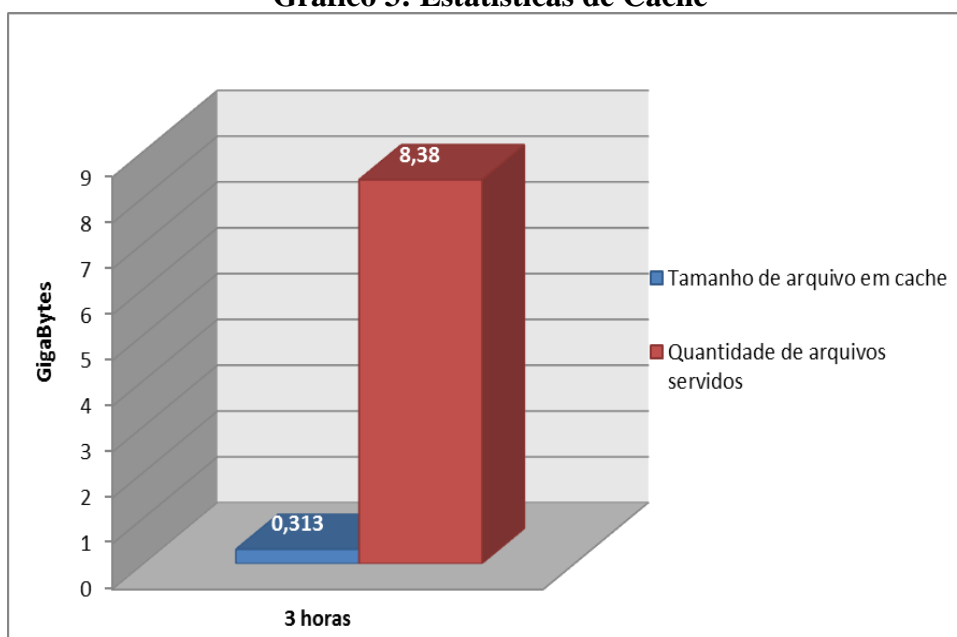
O aplicativo executável foi utilizado nos dez computadores disponibilizados para o teste, onde simulou por durante três horas, o tráfego de *Internet* de usuários acessando somente as páginas listadas no aplicativo. Ao término das três horas foram coletados através do *Software MyAuth Gateway 3* e *Nettraffic* os dados referentes ao tráfego de rede gerado no próprio *Proxy MyAuth Gateway 3* e na placa *Ethernet* do *Windows Server 2008 R2*.

No (GRAF.02) encontra-se o volume de tráfego gerado pela placa *Ethernet* do *Server 2008* monitorada pelo *Nettraffic* durante um período de três horas:

Gráfico 2: Tráfego Total de Internet

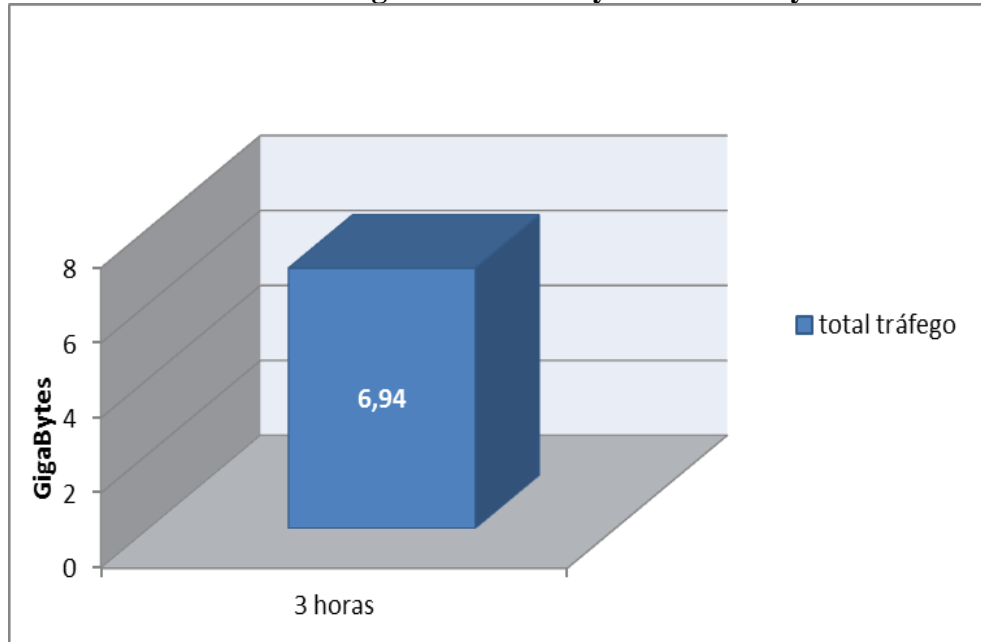
Fonte: Dados da Pesquisa

O (GRAF.03) mostra estatísticas do SuperCache referentes a quantidade de arquivos em cache e de arquivos servidos pelo cache durante um período de três horas.

Gráfico 3: Estatísticas de Cache

Fonte: Dados da Pesquisa

A seguir se encontra no (GRAF.04) o volume de tráfego coletado pelo *MyAuth Gateway 3* referente ao volume de tráfego dos dez computadores durante um período de três horas:

Gráfico 4: Tráfego de rede do MyAuth Gateway 3

Fonte: Dados da Pesquisa

6 ANÁLISE DOS RESULTADOS

Neste capítulo apresenta-se e discutem-se os resultados obtidos nos testes conduzidos de acordo com o ambiente descrito previamente no capítulo anterior, com o objetivo de avaliar a eficiência do Sistema de cache. Para isso na seção 6.1 apresenta-se a análise do tráfego de *Internet*, na seção 6.2 observa-se a análise do volume total do tráfego de entrada e saída. A análise de arquivos servidos é exposta na seção 6.3, e por fim na seção 6.4 mostra a análise de estatísticas de cache.

Antes de relatar a análise dos resultados obtidos neste trabalho, é importante ressaltar que antes de realizar os testes dos cenários A e B, efetuou-se testes prévios para verificar o funcionamento dos domínios fornecidos pelo sistema SuperCache para armazenamento em cache. Após a verificação constatou-se que o cache de domínios como Globo, Uol, entre outros não estavam sendo efetuados. Posterior à efetuação dos testes dos cenários A e B, testou-se novamente os domínios disponíveis para cache e verificou-se que o SuperCache não estava realizando cache de vídeos do Youtube.

Os problemas encontrados foram relatados ao mantenedor do produto, a fim de identificar se as falhas ocorriam devido a alguma deficiência do sistema de *caching* ou era apenas um erro de configuração do mesmo. Após a análise das situações problemáticas, o mantenedor informou que pode haver mudanças na forma de implementação dos domínios periodicamente devido a alguma atualização, o que inviabiliza o armazenamento em cache dos arquivos.

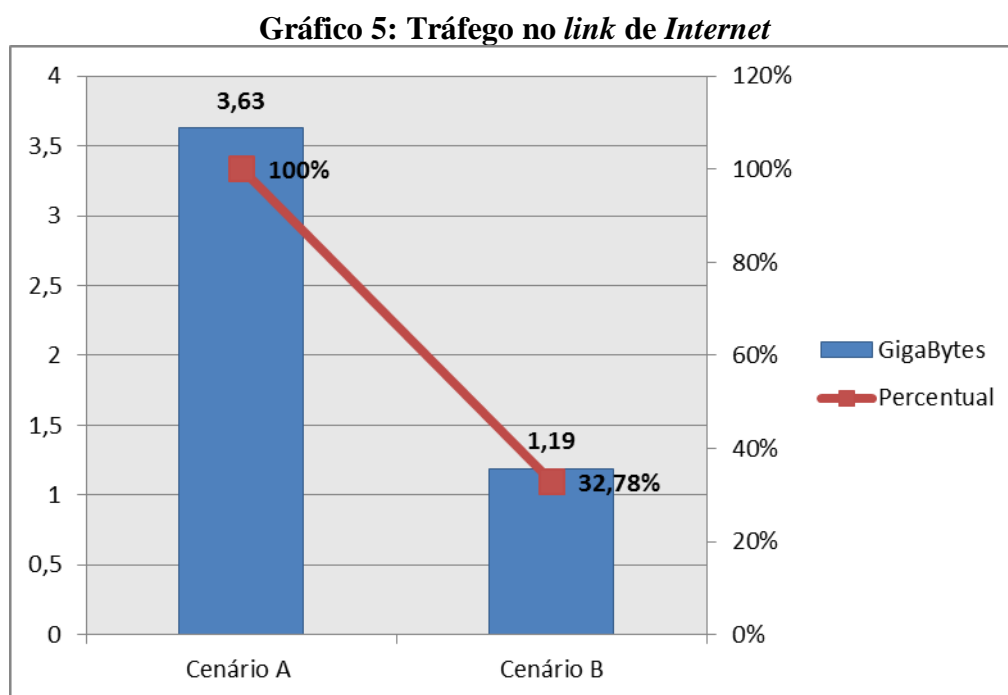
6.1 Análise de tráfego de Internet

Nesta análise apresenta-se o comparativo entre a coleta de dados do tráfego de *Internet* referente ao *software Nettraffic* dos cenários A e B correspondentes às seções 4.1 e 4.2 respectivamente.

Observa-se no (GRAF.05) que o tráfego total de entrada e saída, foi de 3,63 *Gigabytes* no cenário A, sem a utilização do Servidor de cache, e de 1,19 *GigaBytes* no cenário B, ou seja obteve-se uma economia no tráfego de *Internet* de 67,22% em relação ao cenário A, utilizando-se do Sistema de cache armazenando localmente objetos da *Internet*. Com isso constatou-se que existe uma diferença considerável entre os dois cenários, sendo que no cenário A foi trafegado 2,44 *GigaBytes* a mais que no B. A partir da exposição dos dados pressupõe-se que houve uma grande economia no *link* de *Internet* fornecido, pois o cenário B

além de conter um tráfego total de *bytes* enviados/recebidos menor, o acesso às páginas *web* e *download* de conteúdos multimídia foi mais eficaz durante o teste.

Segundo Oliveira (2004), citando Jonack e Murta (2002) “o uso de sistemas de cache na *web* trás como benefício economia no consumo de largura de banda devido à diminuição do tráfego na rede”.



Fonte: Dados da Pesquisa

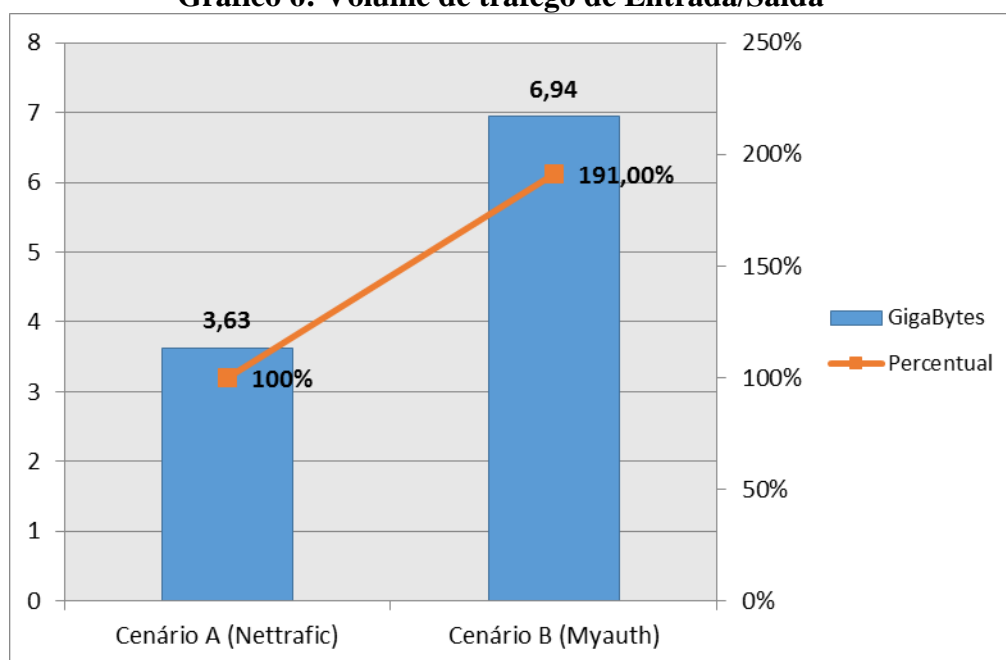
6.2 Análise do volume total de tráfego de Entrada/Saída

Nesta análise apresenta-se o comparativo entre a coleta de dados do tráfego total de *download/upload* de ambos os cenários, referente ao *software Nettraffic* do cenários A e do *Myauth Gateway 3* do cenário B utilizando-se do *plug-in SuperCache*.

Como mostra no (GRAF.06) o tráfego total de entrada e saída, foi de 3,63 *GigaBytes* no cenário A, sem a utilização do Servidor de cache, e de 6,94 *GigaBytes* no cenário B, utilizando-se servidor de cache para realizar cache em disco de objetos da *Internet*.

A partir do que foi exposto destaca-se que com a utilização do *SuperCache* ouve uma crescente no tráfego local de *download/upload* de 3,31 *GigaBytes*, ou seja, 91% em relação ao cenário A. É possível constatar que o *SuperCache* mostrou-se eficiente na entrega de solicitações de dados, disponibilizando mais dados em um mesmo período de tempo em relação ao cenário sem a sua utilização.

Gráfico 6: Volume de tráfego de Entrada/Saída



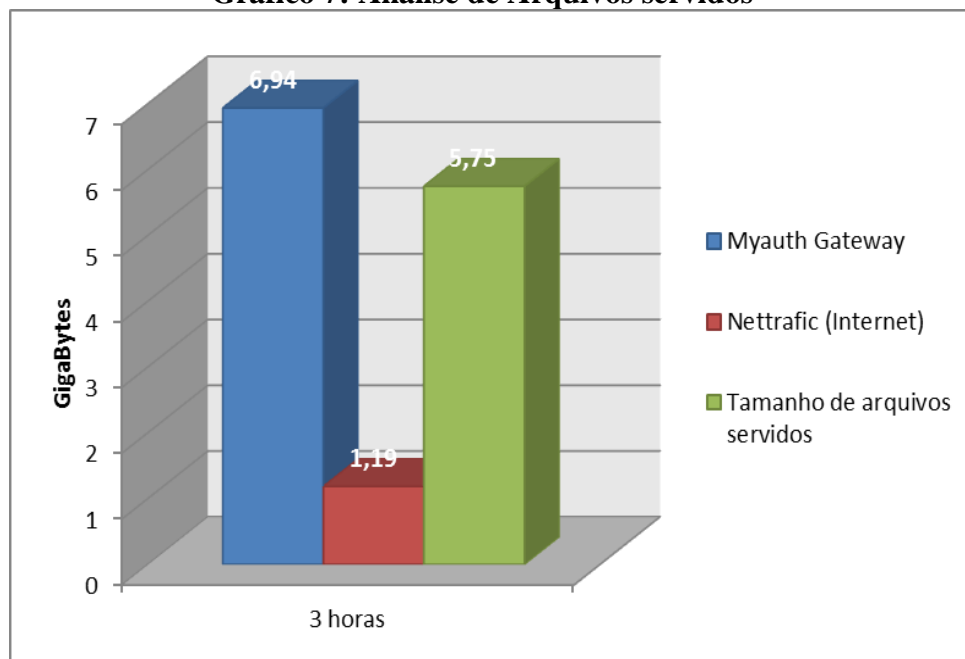
Fonte: Dados da Pesquisa

Segundo OLIVEIRA (2004), o uso de sistemas de cache *web* acelera significativamente as operações na *Internet*, visto que grande parte dos objetos acaba trafegando apenas localmente.

6.3 Análise de arquivos servidos

Esta análise refere-se ao comparativo entre a coleta de dados do tráfego total de *download/upload* coletado pelo *Myauth Gateway 3* e *Nettraffic* referente ao cenário B com o sistema de cache ativo.

A partir do que está exposto no (GRAF.07) nota-se que de 6,94 *GigaBytes* capturados pelo *Myauth Gateway 3*, foi necessário a requisição de apenas 1,19 *GigaBytes* de dados nos servidores de origem. Segundo a métrica BHR o percentual de *Bytes* requisitados pelos clientes é de 0,83%. Então se constata que a quantidade de arquivos servidos pelo cache foi de 5,75 *GigaBytes*, um valor satisfatório em relação a quantidade de dados requisitados. Isso significa que o SuperCache não necessitou de fazer diversas solicitações diretamente ao servidor remoto, propiciando um melhor aproveitamento do *link* de *Internet* e fornecendo arquivos com um menor tempo de latência.

Gráfico 7: Análise de Arquivos servidos

Fonte: Dados da Pesquisa

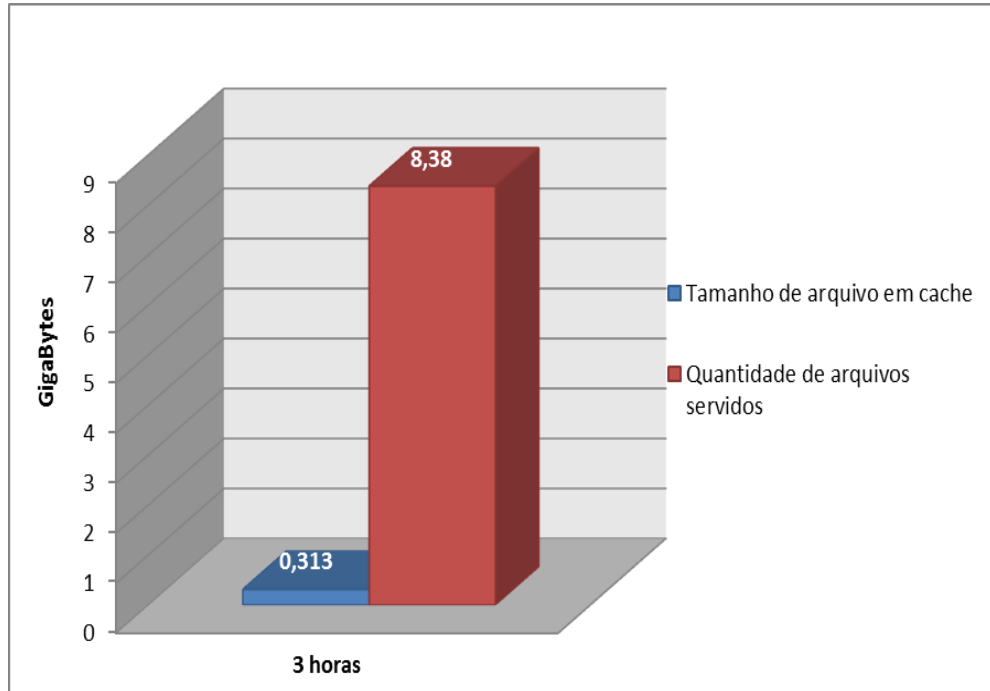
É importante ressaltar que STEINMACHER (2004) afirma que “uma das três principais vantagens do uso de *caching* é que a latência entre pedido e resposta é reduzida, fazendo com que as páginas sejam carregadas mais rapidamente”.

6.4 Análise de Estatísticas de Cache

Esta análise apresenta-se os dados estatísticos correspondentes ao confronto entre o tamanho de arquivos que já se encontram em cache e o total de arquivos fornecidos aos clientes pelo servidor de cache.

A análise do (GRAF.08) revela que durante o período de três horas a quantidade de arquivos servidos pelo SuperCache aos computadores clientes foi de 8,38 *GigaBytes*, e o tamanho de arquivos armazenados em cache é de 0,31 *GigaBytes*. Dessa forma é notável a eficiência do SuperCache na entrega de requisições as máquinas clientes.

É importante ressaltar que segundo OLIVEIRA (2004), “o uso de sistemas de cache *web* resulta no aumento da disponibilidade dos dados e objetos da *Internet*, além de minimizar o espaço utilizado em disco, uma vez que é feita apenas uma cópia do objeto para vários usuários”.

Gráfico 8: Estatísticas de Cache

Fonte: Dados da Pesquisa

7 CONSIDERAÇÕES FINAIS

Neste trabalho apresentou-se a implantação do SuperCache, um sistema de cache em conjunto com o *software Myauth Gateway 3* para o gerenciamento e economia no tráfego de *Internet* no Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais campus São João Evangelista.

Os objetivos propostos inicialmente como: estudo dos materiais necessários, instalação e configuração dos *softwares*, preparação dos cenários para teste, realização dos testes e análise dos dados obtidos, foram alcançados ao longo do desenvolvimento do projeto.

Essa pesquisa, apresentou uma análise de como obter economia no tráfego de *Internet* do IFMG-SJE através do SuperCache, medindo seu desempenho utilizando a métrica BHR, que atualmente é a mais indicada para este fim.

O SuperCache demonstrou sua eficiência em meio aos testes e análises realizadas. Isso se justifica pelo fato de que a ferramenta proveu uma economia de 67,22%, de acordo com o (GRAF.05), no tráfego de *Internet* do *campus* mediante aos cenários construídos. Além disso, segundo o (GRAF.06), seu uso proporcionou um aumento no volume de dados trafegados de 91%, entretanto consumindo menos a banda de *Internet*, isso porque as informações estavam armazenadas em um servidor local, desta forma a entrega das solicitações aconteceram mais rapidamente. A diminuição da latência foi algo notável com a utilização do serviço de cache, pelo fato do SuperCache proporcionar a entrega de determinados objetos quase que instantaneamente aos usuários.

Segundo as estatísticas de cache fornecidas pelo SuperCache mostradas no (GRAF.08), a quantidade de arquivos servidos foi de 8,38 *GigaBytes*, valor considerado inconsistente, uma vez que o total de tráfego, *download/upload*, capturado no mesmo ambiente pelo *software Myauth Gateway* como especificado no (GRAF.04), foi 6,94 *GigaBytes*. Por isso, utilizou-se o método BHR para verificar o percentual real de arquivos entregues aos clientes sem solicitação ao servidor de origem.

De acordo com o (GRAF.07), a quantidade real de objetos entregues foi 5,75 *GigaBytes*, um valor consideravelmente abaixo do especificado pelo próprio servidor de cache, porém altamente satisfatório.

Outro aspecto importante que foi observado é a capacidade do SuperCache de realizar cache de atualizações de *softwares*, *Windows Update*, e antivírus, otimizando estes processos que feitos da forma tradicional consomem muito *link* de *Internet*. Destaca-se também o nível de interatividade de configuração e gerenciamento entre o *Myauth Gateway* e SuperCache,

proporcionada por suas interfaces gráficas altamente amigáveis, que permitem aos administradores de rede um gerenciamento eficaz desses sistemas.

Apesar da eficiência do SuperCache mediante aos testes realizados, pode ser observado que este possui algumas deficiências que são influenciadas por modificações na forma de disponibilização de dados pelos servidores e domínios, os provedores dos arquivos acessados. O reporte dessas falhas ao mantenedor da ferramenta de cache colaborou para a disponibilização futura de uma atualização do SuperCache que visa, a correção dessas deficiências.

Entretanto, conclui-se que a economia gerada no tráfego de *Internet* aliada a um maior volume de dados entregues aos clientes, poderá proporcionar a quem implantar o servidor de cache, uma boa economia de banda para acesso a *Internet* e uma melhora no serviço oferecido aos usuários.

REFERÊNCIAS

- ANDRADE, V. S.; Sass, S. R. R. **Segurança de Rede: Monitoramento de Rede com Proxy Cache Squid**. Disponível em: <<http://sare.anhanguera.com/index.php/anupg/article/view/3143/1170>>. Acesso em: 10 Abr. 2013.
- AMOROSO, D. **Como monitorar o tráfego de rede da sua banda larga**. 2012. Disponível em: <<http://www.tecmundo.com.br/tutorial/8122-como-monitorar-o-trafego-de-rede-da-sua-banda-larga-video-.htm>>. Acesso em: 20 Set. 2013.
- BARBOSA, M. A. da S. e MARTINS, C. A. P. da S. **Proposta de método para análise de desempenho de estruturas hierárquicas de servidores web-caching**, 2002. Disponível em: <Http://www.biblioteca.pucminas.br/teses/engeletrica_barbosama_1.pdf>. Acesso em: 8 Mai. 2013.
- BARWINSKI, L. **O que é Proxy?** Disponível em: <<http://www.tecmundo.com.br/navegador/972-o-que-e-Proxy-.htm>>. Acesso em: 9 Abr. 2013.
- CHESHIRE, S. **Latency and the Quest for Interactivity**. Disponível em: <<http://www.stuartcheshire.org/papers/LatencyQuest.html>>. Acesso em: 10 Abr. 2013.
- DODONOV, E. **Um Mecanismo Integrado de Cache e Prefetching para Sistemas de Entrada e Saída de Alto Desempenho**, 2004. Disponível em: <<Http://eugeni.dodonov.net/files/dissertacao.pdf>>. Acesso em: 8 Mai. 2013.
- FONSECA, W. **O que é Cache?** Disponível em: <<http://www.tecmundo.com.br/navegador/201-o-que-e-cache-.htm>>. Acesso em: 9 Abr. 2013.
- FOROUZAN, B. A. **Comunicação de Dados e Redes de Computadores**. 3 ed. Porto Alegre: Bookman, 2006.
- BARBOSA, G. A. R.; PRATES, R. O.; ALMEIDA, V. **Combinando Métodos Quantitativos e Qualitativos para Caracterizar o Uso de Redes Sociais Online por Usuários Surdos do Brasil**. Disponível em: <http://www.imago.ufpr.br/csbc2012/anais_csbc/eventos/semish/artigos/SEMISH%20-%20Combinando%20Metodos%20Quantitativos%20e%20Qualitativos%20para%20Caracterizar%20o%20Uso%20de%20Redes%20Sociais%20Online%20por%20Usuarios%20Surdos%20do%20Brasil.pdf>. Acesso em: 5 de Out. 2013.
- ELIAS, R. **Swing: Abrindo uma URL**. Disponível em: <<http://javaerro.blogspot.com.br/2011/02/se-voce-fazendo-ou-fara-um-programa.html>>. Acesso em: 10 Out. 2013.
- JEFFRIES, A. **Feature: COSS (Cyclic Storage System Object)**. Disponível em: <<http://wiki.squid-cache.org/Features/CyclicObjectStorageSystem>>. Acesso: em 23 Out. 2013.
- MICROSOFT. **Visão Geral do Produto Windows Server 2008**. 2007. Disponível em:

<<http://www.microsoft.com/brasil/servidores/windowsserver2008/evaluation/overview.msp>>. Acesso em: 20 Set. 2013.

MORIMOTO, C. E. **Servidores Linux, guia prático**. Sul Editores, 2008.

OLIVEIRA, A. J. **Uso de caches na web-Influência das políticas de substituição de objetos 2004**. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tdc-26072004-12140.php>>. Acesso em: 10 Mai. 2013.

PONTES, E.; HIRARA, S.; HONÓRIO, S. **Segurança e Aceleração da Internet: Utilização de Proxy Servers para Manutenção de Web Caches**, 2008. Disponível em: <<http://www.pontes.inf.br/docs/proxy.pdf>>. Acesso em: 10 Mai. 2013.

REDDY, S. R. **Seminar Report on Web Caching**, 2005. Disponível em: <<http://sit.iitkgp.ernet.in/research/aut04seminar1/11r.pdf>>. Acesso em: 10 Mai. 2013.

RICARDO, F. S. **Como montar um Proxy reverse no servidor Apache**. Disponível em: <<http://www.vivaolinux.com.br/artigo/Como-montar-um-Proxy-reverse-no-servidor-Apache?pagina=3>>. Acesso em: 10 Abr. 2013.

REIS, C. F.; RAIMUNDO, G.; CARMO, F.T. **Projeto Squid**, 2006. Disponível em: <<http://ebookbrowse.com/73873164-tcc-projeto-squid-pdf-d378196055>>. Acesso em: 10 Mai. 2013.

SANTOS, J. **Unix File System**. Disponível em: <http://www.dc.fi.udc.es/~so-grado/Unix_File_System.pdf>. Acesso em: 10 Mai. 2013.

SERWIN, S. **Microsoft ISA Server, Part I – introduction, installation, configuration, Web caching and Internet access**, 2013. Disponível em: <http://www.windowsecurity.com/articles-ututorials/firewalls_and_VPN/Microsoft_ISA_Server_Part_I__introduction_installation_configuration_Web_caching_and_Internet_access.html>. Acesso em: 10 Mai. 2013.

SIEMINSKI, A. **The impact of Proxy caches on Browser Latency**, 2005. Disponível em: <<http://www.tmrfindia.org/ijcsa/V2I22.pdf>>. Acesso em: 10 Abr. 2013.

SILVA, L. R. B. B. **Aceleração HTTP: Um comparativo de performance entre as soluções Squid e Varnish**, 2009. Disponível em <<http://www.ginix.ufla.br/files/mono-LucasSilva.pdf>>. Acesso em: 10 Mai. 2013.

STEINMACHER, I. F. **Técnicas de Web Caching e Prefetching com Prioridades**. Disponível em: <<http://www.igor.pro.br/publica/papers/TL.pdf>>. Acesso em: 10 Abr. 2013.

TEIXEIRA, A. O. **Proxy Reverso com o Apache**. Disponível em: <<http://www.ataliba.eti.br/txts/apache-Proxy.pdf>>. Acesso em: 12 Abr. 2013.

TMSOFT. **Manual MyAuth Gateway 3 Pré-requisitos**. Disponível em: <http://manual.tmssoft.com.br/?open=manual/myauth3_prerequisitos.htm>. Acesso em: 10 Set. 2013.

KINGHOST. **Varnish Cache**. Disponível em: <<http://www.kinghost.com.br/varnish>>. Acesso em: 12 Mai. 2013.

MYCROSOFT. **Informações sobre o Produto**, 2006. Disponível em: <<http://www.microsoft.com/brasil/servidores/isaserver/prodinfo/default.aspx>>. Acesso em: 12 Mai. 2013.

ZONA RESEARCH. **The Economic Impacts of Unacceptable Web Site Download Speeds**, inc., Abril 1999. Disponível em: <http://www.webperf.net/info/wp_downloadspeed.pdf>. Acesso em: 10 Abr. 2013.

APÊNDICE A- CÓDIGO DO APLICATIVO JAVA

```

package sites;
import java.awt.Desktop;
import java.util.concurrent.DelayQueue;
import javax.swing.JOptionPane;

public class pagina extends javax.swing.JFrame {
    public pagina() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton1 = new javax.swing.JButton();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jButton1.setText("abrir site");
        jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jButton1MouseClicked(evt);
            }
        });
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(165, 165, 165)
                    .addComponent(jButton1)
                    .addGap(223, Short.MAX_VALUE)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                        layout.createSequentialGroup()
                            .addGap(137, 137, 137)
                            .addComponent(jButton2)
                            .addGap(137, 137, 137))
                    .addContainerGap(165, Short.MAX_VALUE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(165, 165, 165)
                    .addComponent(jButton1)
                    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 100,
                        Short.MAX_VALUE)
                    .addComponent(jButton2)
                    .addContainerGap(165, Short.MAX_VALUE))
        );
    }
}

```

```

        .addGap(93, 93, 93))
    );
    pack();
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
}

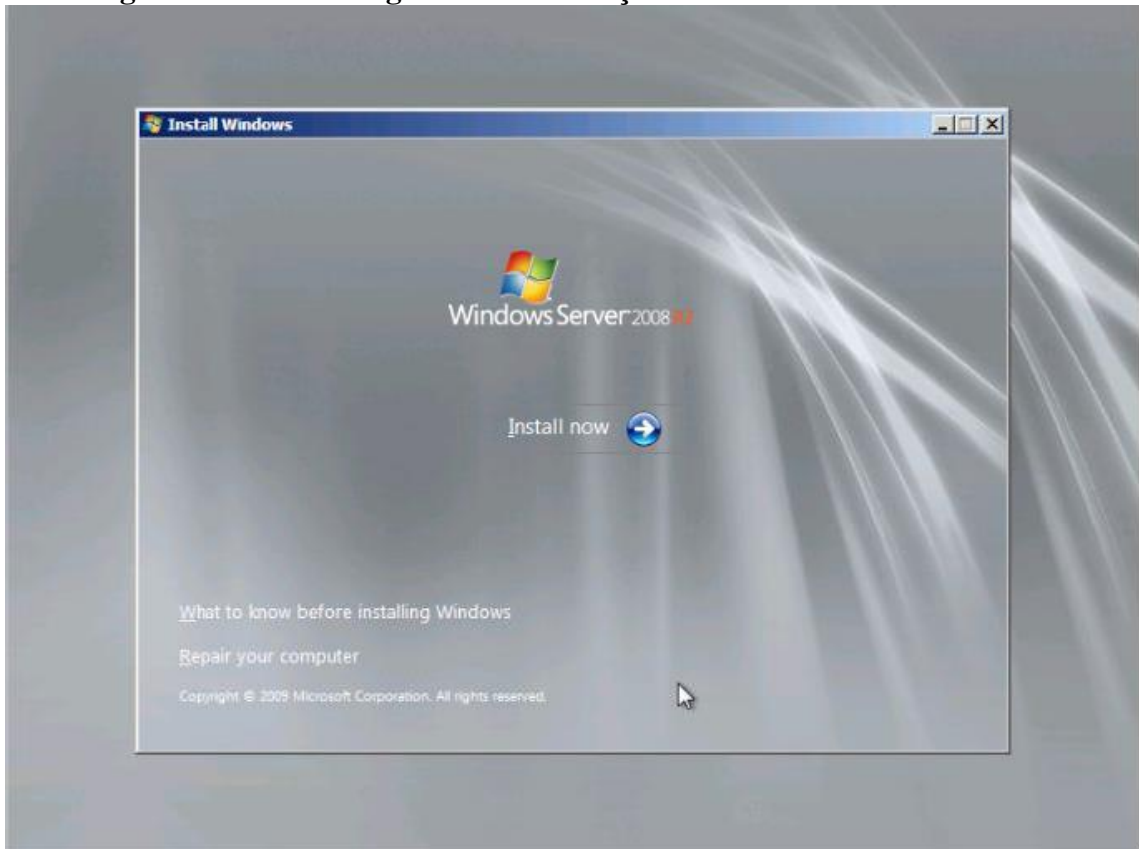
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {
    Desktop desk = java.awt.Desktop.getDesktop();
    int num = 0;
    while(num == 0){
        try {
            desk.browse(new
java.net.URI("http://www.webtvcn.com/canal/noticias/senador_boliviano_26_08_13"));
            Thread.sleep(100000);//1 minuto
            desk.browse(new java.net.URI("http://www2.videolog.tv/1007501"));
            Thread.sleep(100000);
            desk.browse(new java.net.URI("http://www.viddler.com/features"));
            Thread.sleep(100000);
            desk.browse(new
java.net.URI("http://etrustdownloads.ca.com/legacy/av/fi_MacOSX.tar"));
            Thread.sleep(20000);
            desk.browse(new
java.net.URI("http://www.dailymotion.com/video/x13suuy_petroleo-sobe-por-causa-da-crie-
siria_news"));
            Thread.sleep(100000);
            desk.browse(new java.net.URI("http://blip.tv/everydaydrinkers/eva-s-directive-movie-
cocktail-6634319"));
            Thread.sleep(100000);
            desk.browse(new
java.net.URI("http://bitgravity.com/products/bg_video_director.html"));
            Thread.sleep(100000);
            desk.browse(new java.net.URI("http://video.br.msn.com/watch/video/kangaroo-
adores-baby-lemur/1jiiiz6kd"));
            Thread.sleep(150000);
            desk.browse(new java.net.URI("http://dc377.4shared.com/download/0tPUdu-
_Bob_Esponja_O_Filme_dublado_em.3gp?tsid=20130830-132445-ca40524b"));
            Thread.sleep(30000);
            desk.browse(new java.net.URI("http://www.youtube.com/watch?v=h1A9Kc5iNkQ"));
            Thread.sleep(200000);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {

```

```
        public void run() {  
            new pagina().setVisible(true);  
        }  
    });  
}  
private javax.swing.JButton jButton1;  
}
```

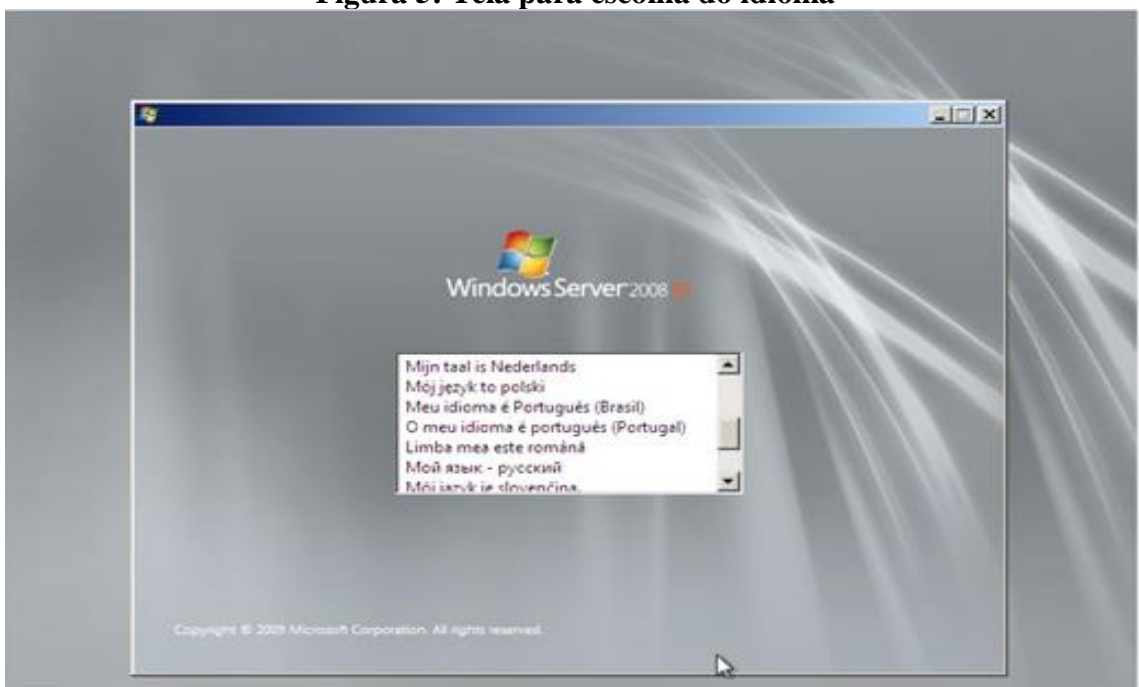
APÊNDICE B - TELAS DE INSTALAÇÃO DO WINDOWS SERVER 2008 R2 ENTERPRISE

Figura 4: Tela do Programa de Instalação do Windows 2008 Server R2



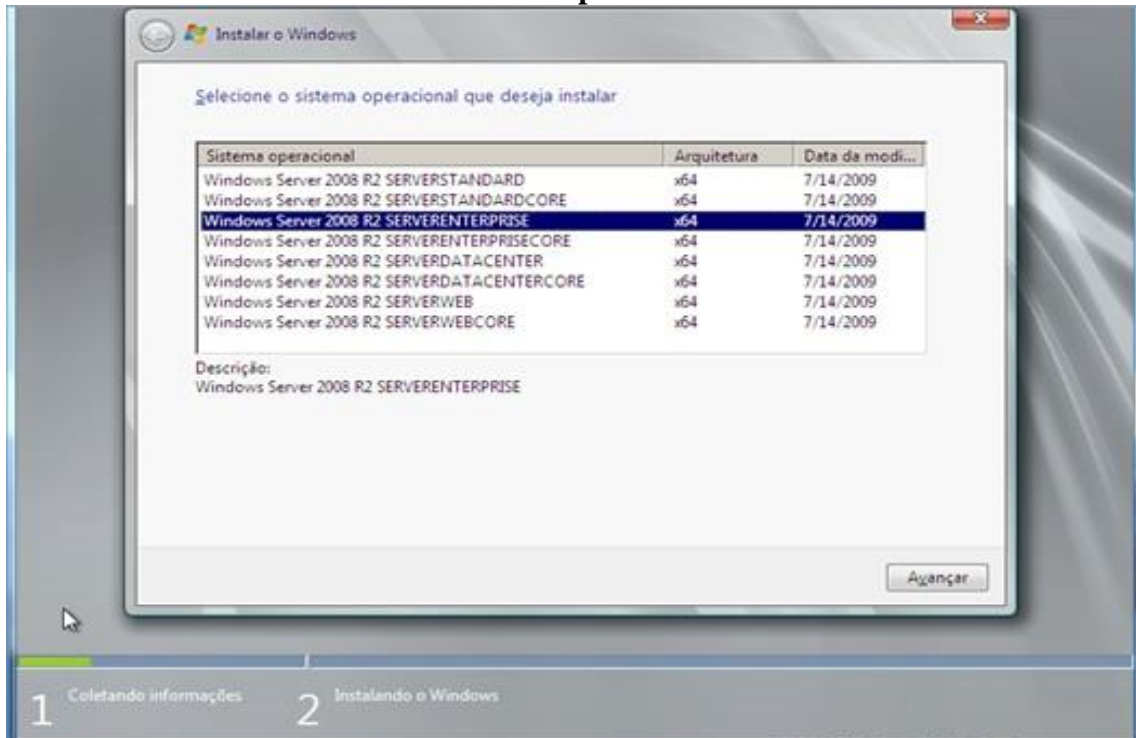
Fonte: Elaborado pelos autores

Figura 5: Tela para escolha do idioma



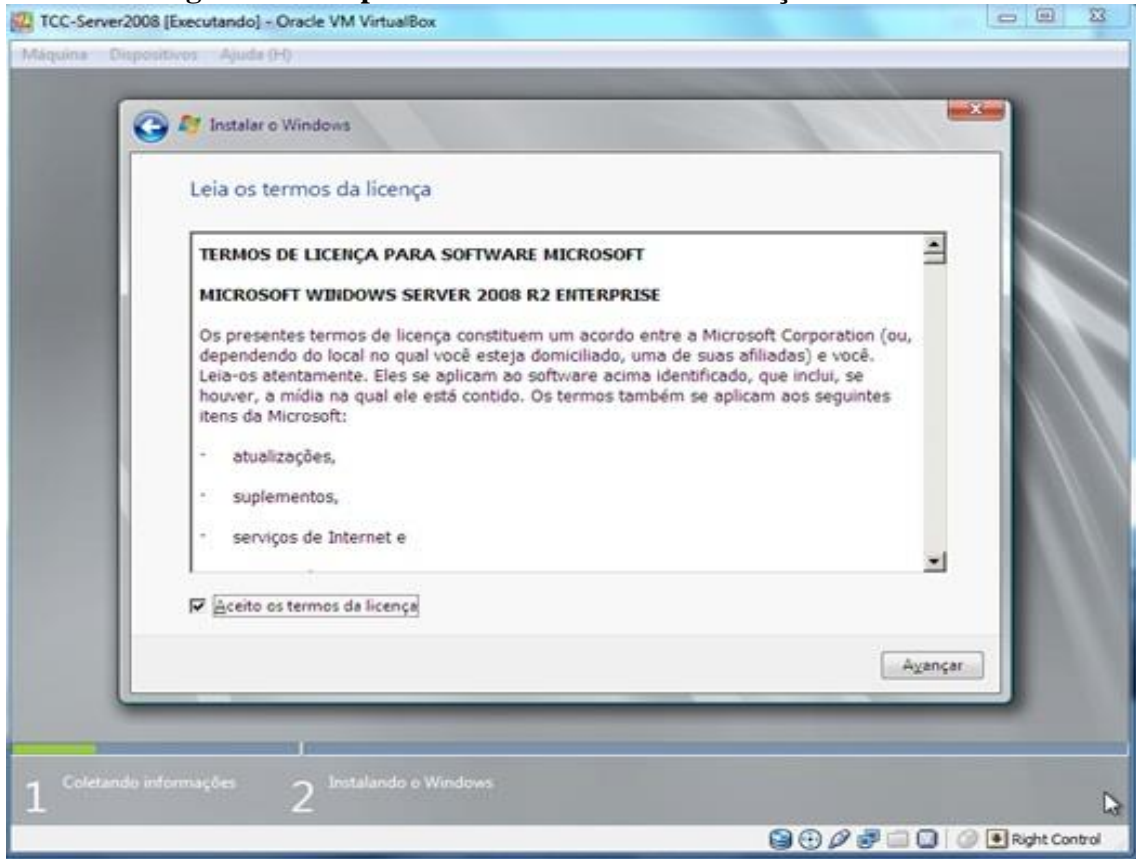
Fonte: Elaborada pelos autores

Figura 6: Tipo de instalação desejada, nesse caso Windows Server 2008 R2 Server Enterprise



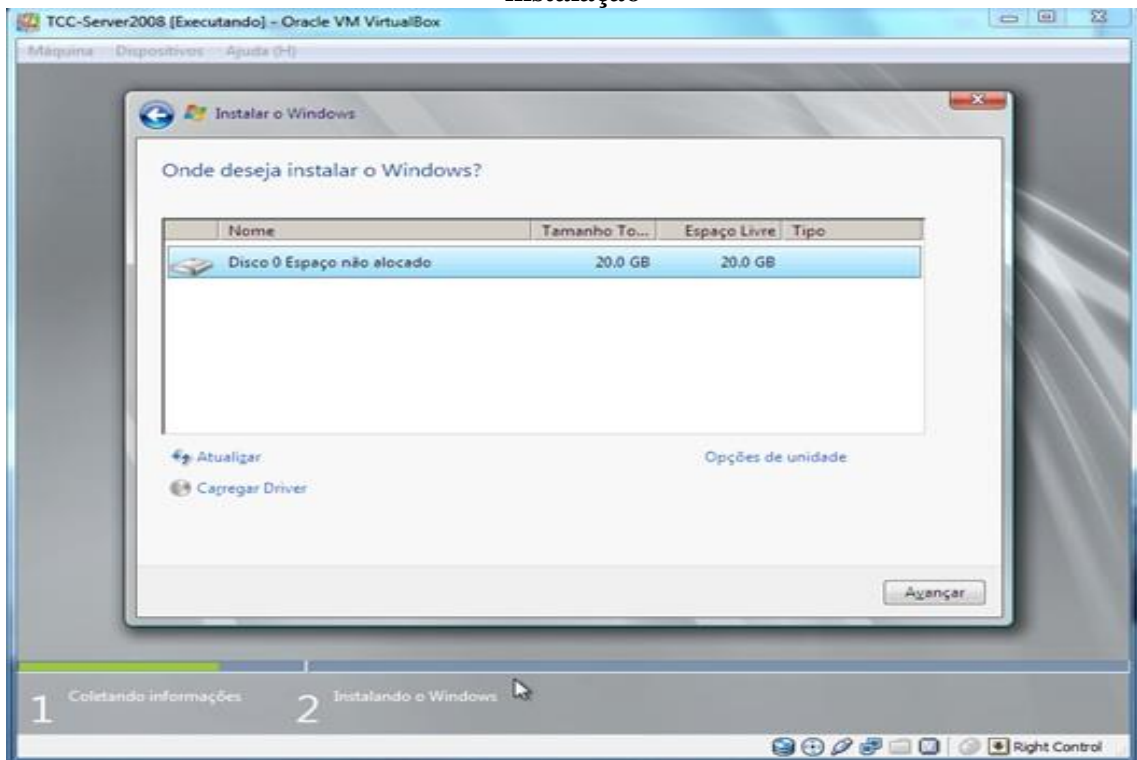
Fonte:: Elaborada pelos autores

Figura 7: Tela para aceitar o Termo de Licença da Microsoft



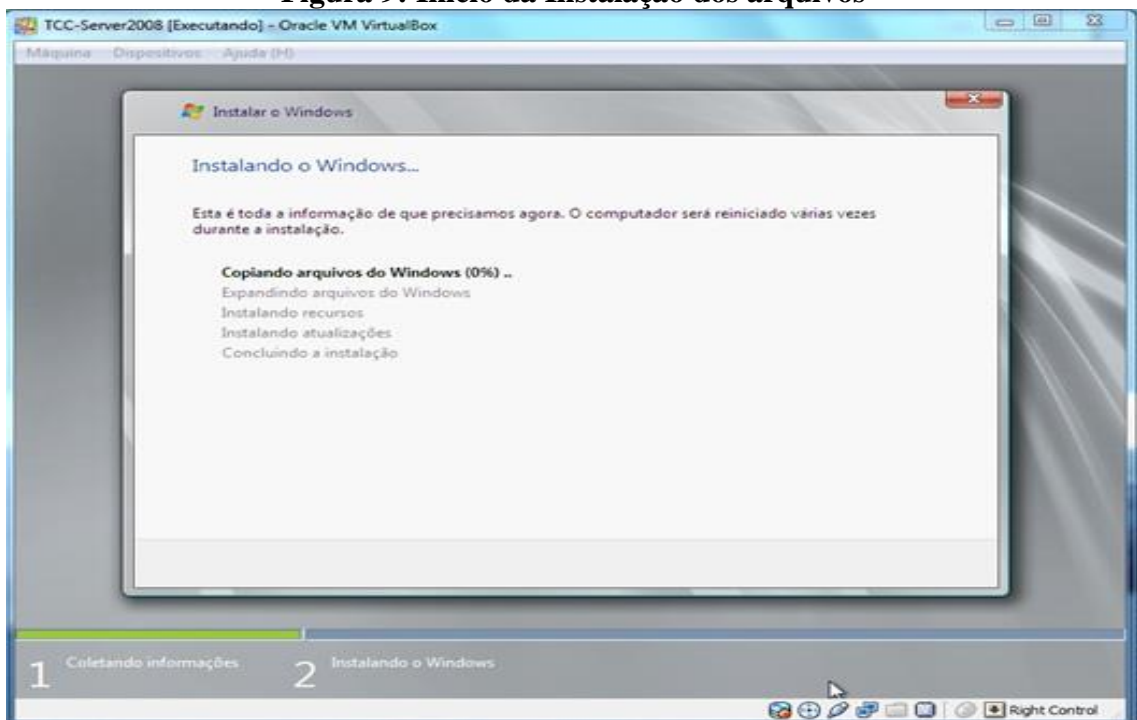
Fonte: Elaborada pelos autores

Figura 8: Tela onde deve ser escolhida em qual unidade de disco se deseja realizar a instalação



Fonte: Elaborada pelos autores

Figura 9: Início da Instalação dos arquivos



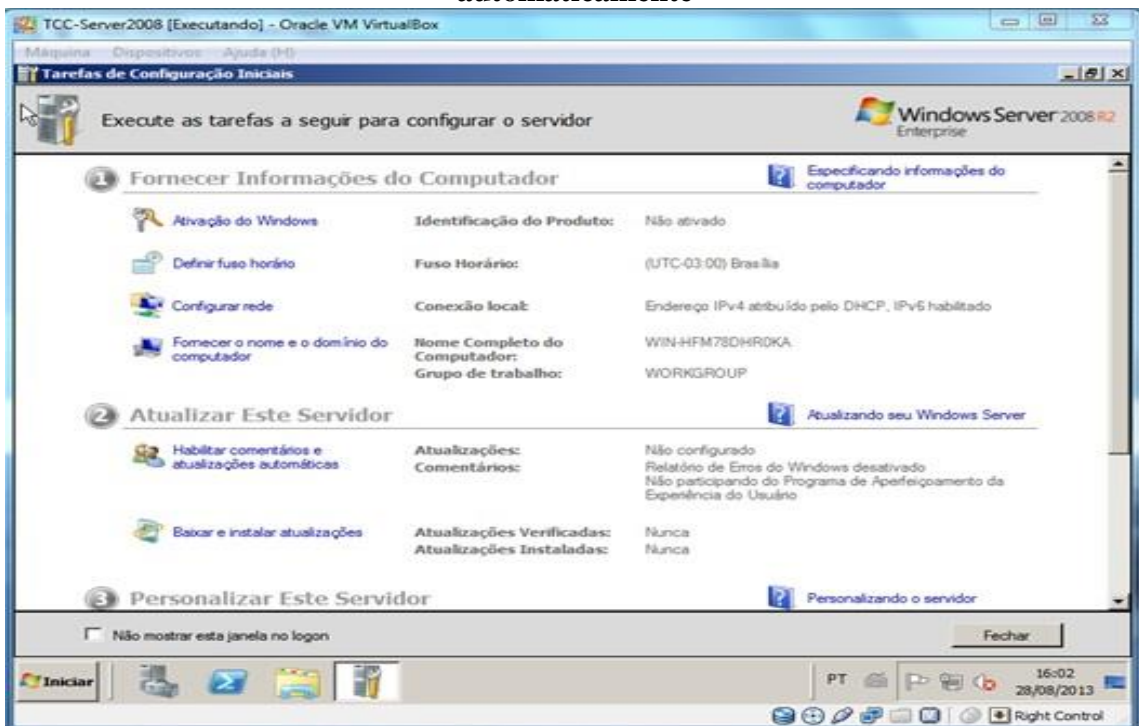
Fonte: Elaborada pelos autores

Figura 10: Reinicialização e cadastro de uma senha para o administrador



Fonte: Elaborada pelos autores

Figura 11: Tela onde um painel contendo as primeiras ações a serem realizadas é aberto automaticamente



Fonte: Elaborada pelos autores

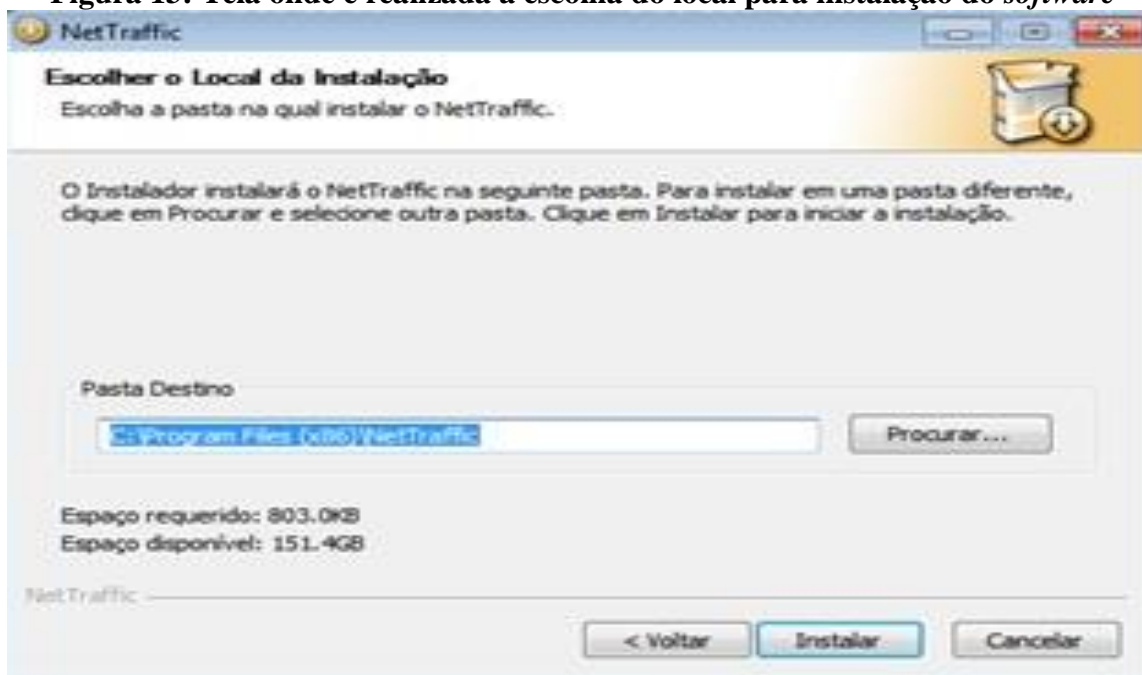
APÊNDICE C – TELAS DE INSTALAÇÃO DO ASSISTENTE *NETTRAFIC*

Figura 12: Tela do Assistente de instalação do Nettraffic



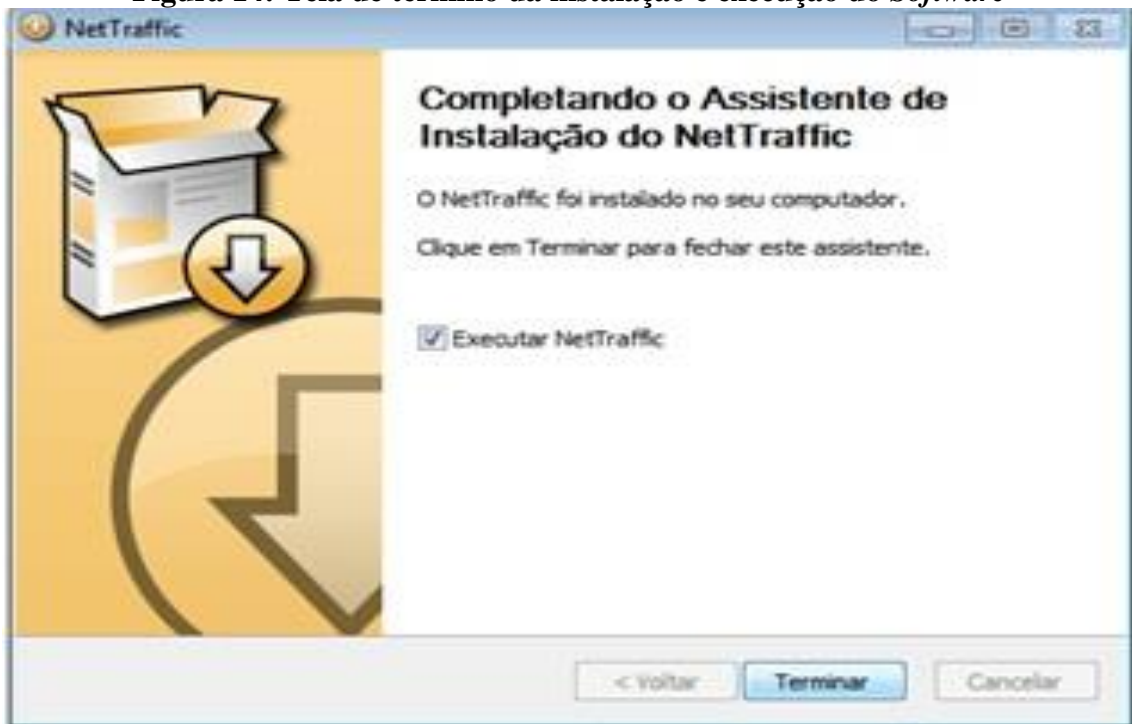
Fonte: Elaborada pelos autores

Figura 13: Tela onde é realizada a escolha do local para instalação do *software*



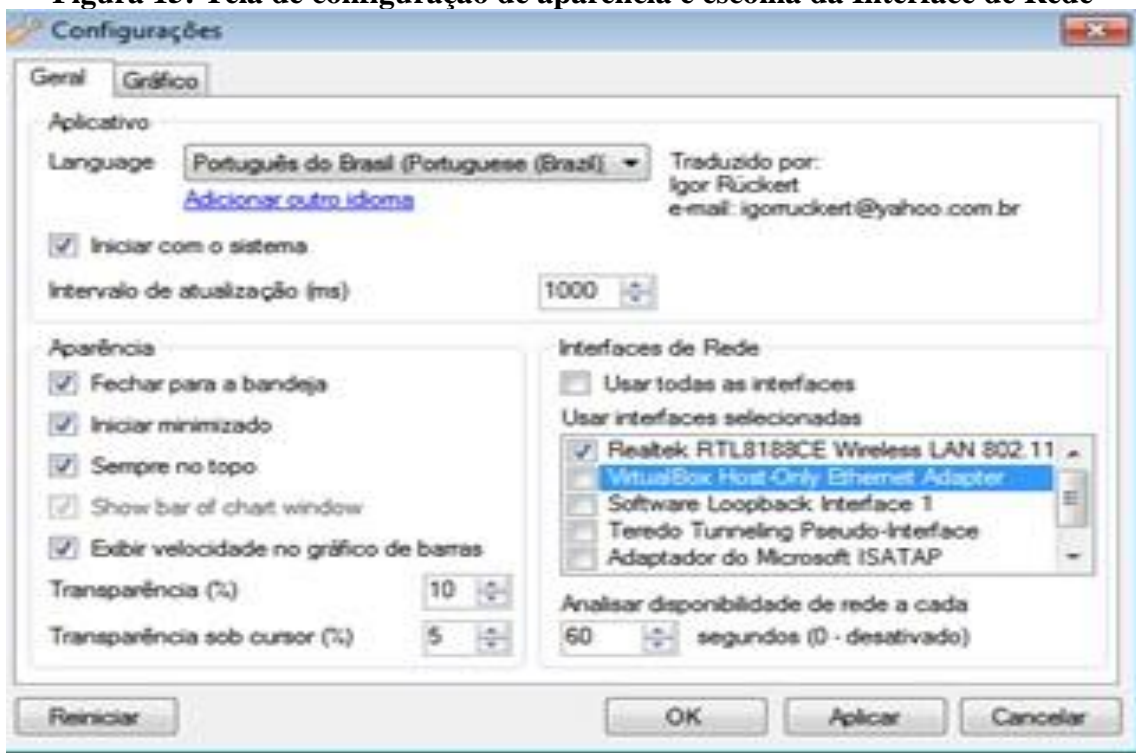
Fonte: Elaborada pelos autores

Figura 14: Tela de término da instalação e execução do *Software*



Fonte: Elaborada pelos autores

Figura 15: Tela de configuração de aparência e escolha da Interface de Rede



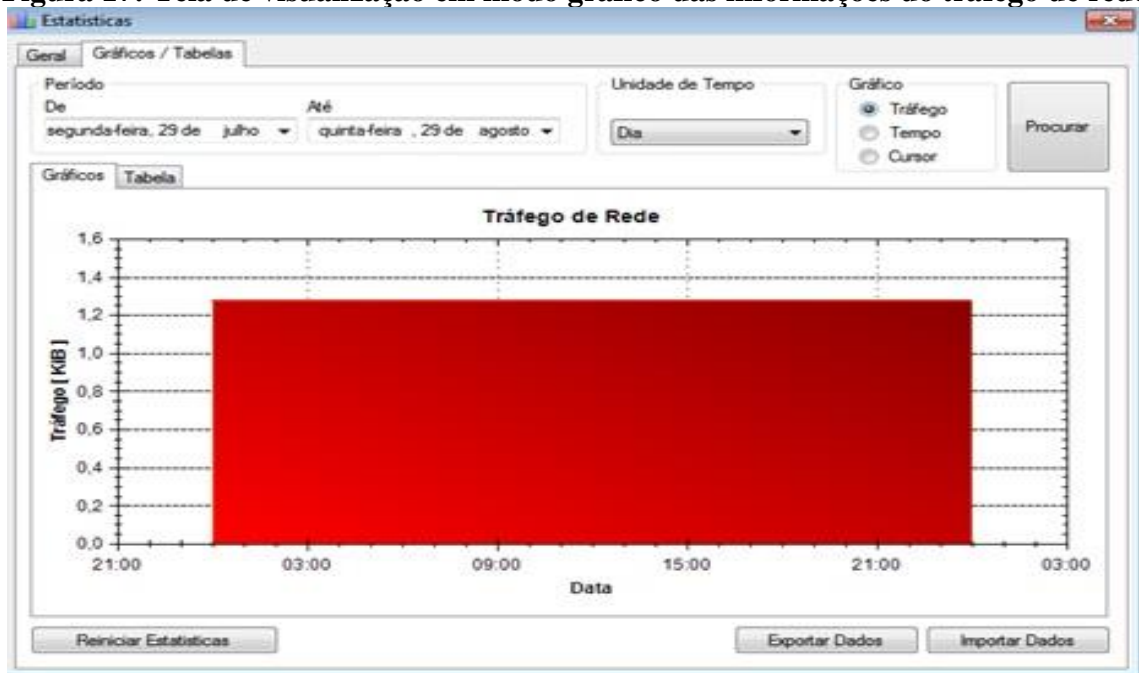
Fonte: Elaborada pelos autores

Figura 16: Visualização das Estatísticas após a instalação do *software* em modo geral



Fonte: Elaborada pelos autores

Figura 17: Tela de visualização em modo gráfico das informações do tráfego de rede



Fonte: Elaborada pelos autores