

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS GUARAPUAVA
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

JONAS JOSÉ MOREIRA DE SOUZA

**SERVIDOR PROXY SQUID: UMA FERRAMENTA NO AUXÍLIO DO
USO ADEQUADO DA INTERNET**

TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA

2014

JONAS JOSÉ MOREIRA DE SOUZA

**SERVIDOR PROXY SQUID: UMA FERRAMENTA NO AUXÍLIO DO
USO ADEQUADO DA INTERNET**

Trabalho de Conclusão de Curso apresentado à disciplina Trabalho de Conclusão de Curso II do Curso de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná como requisito parcial para aprovação.

Orientador: Prof. Ms. Hermano Pereira

GUARAPUAVA

2014

**ATA DE DEFESA DE MONOGRAFIA DE TRABALHO DE CONCLUSÃO DE CURSO DO
CURSO DE TSI**

No dia 31 de julho de 2014, às 15:00 horas, nas dependências da Universidade Tecnológica Federal do Paraná Câmpus Guarapuava, ocorreu a banca de **defesa da monografia** de Trabalho de Conclusão de Curso intitulada: **“Servidor Proxy Squid: Uma Ferramenta no Auxílio do uso Adequado da Internet”** do acadêmico **Jonas José Moreira de Souza** sob orientação do professor **Prof. Me. Hermano Pereira** do Curso de Tecnologia em Sistemas para Internet.

Banca Avaliadora	
Membro	Nome
Orientador	Prof. Me. Hermano Pereira
Coorientador	
Avaliador 1	Prof. Dr. Luciano Ogiboski
Avaliador 2	Prof. Me. Sediane Carmem Lunardi Hernandes

Situação do Trabalho

Situação	<input type="checkbox"/> Aprovado <input checked="" type="checkbox"/> Aprovado com ressalvas <input type="checkbox"/> Reprovado <input type="checkbox"/> Não Compareceu
Encaminhamento do trabalho para biblioteca	<input checked="" type="checkbox"/> Pode ser encaminhado para biblioteca. <input type="checkbox"/> Manter sigilo para publicação ou geração de patente.

Guarapuava, 31 de julho de 2014.

RESUMO

JOSÉ MOREIRA DE SOUZA, Jonas. SERVIDOR PROXY SQUID: Uma ferramenta no auxílio do uso adequado da Internet. 47 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2014.

Resumo.

Os servidores de Proxy agem como mediadores entre a rede interna (Intranet) e a rede externa (Internet), sendo assim, são as únicas máquinas (computadores) que realizam acessos direto a Internet. Essa característica possibilita que os servidores Proxy sejam utilizados para melhorar a segurança e o desempenho da rede, através do *cache* de páginas Web, e de regras de controle de acesso. O serviço de *web cache* presente no Proxy armazena em disco as páginas acessadas pelos usuários, assim, quando uma mesma página for novamente solicitada por qualquer computador da rede, poderá ser apresentada diretamente do *cache*, sem necessidade de busca ao servidor de origem. Como consequência isso possibilita que o consumo de banda disponível para a Internet possa diminuir. Já o controle de acesso realizado através de regras predefinidas, vem a direcionar o uso da Internet para atividades de interesse da empresa, evitando acessos a *sites* desnecessários e diminuindo a exposição a vírus (programas maliciosos). Este trabalho utilizou como base para coleta de dados a Prefeitura Municipal de Boa Ventura de São Roque, onde por um período de 20 (vinte) dias, ficou ativo um sistema de captura responsável por registrar todos os sites acessados, gerando uma lista de URLs utilizadas em simulações de acesso através de um servidor Proxy. Os testes realizados demonstraram uma estimativa de economia de banda em até 86% (oitenta e seis por cento), considerando o uso de *web cache* e controle de acesso.

Palavras-chave: Proxy, Proxy Cache, Web cache

ABSTRACT

JOSÉ MOREIRA DE SOUZA, Jonas. Server Proxy Squid: A tool in help of use appropriate of Internet. 47 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2014.

The Proxy servers as mediators between the internal network (intranet) and the external network (Internet), so are the only machine (computers) that direct access to the Internet. This characteristic enables that Proxy servers are used to improve the security and the performance of the network through the cache of Web pages, and control rules access. The service present in the web cache proxy stores disk pages accessed by users, so when the same page is requested again by any computer on the network, will be introduced directly of the cache, without necessity of search the source server. As consequence this allows that the consumption of bandwidth available for Internet may decline. while the access control realized through the predefined rules, comes to direct the use of the Internet for activities of interest of company, avoiding access the unnecessary sites and decreasing the exposure to viruses (malicious programs). This study used as the basis for collecting data of City council of Boa Ventura de São Roque, where during 20 (twenty) they stayed active a system the capture responsible for the record all the sites accessed, creating a list the URLs used in simulations of access trough of Proxy server. The aim realizer are showed a estimate of save of bandwidth in until 86% (eighty six percent), considering the use web cache and acess control.

Keywords: Proxy, Proxy Cache, Web cache

LISTA DE FIGURAS

Figura 1	–	Funcionamento da rede sem utilização de Proxy	16
Figura 2	–	Funcionamento da rede com utilização de um servidor Proxy <i>web cache</i>	17
Figura 3	–	Modelo de arquitetura hierárquica	19
Figura 4	–	Modelo de arquitetura distribuído	20
Figura 5	–	Rede física antes da implantação do sistema de coleta	24
Figura 6	–	Rede física após a implantação do sistema de coleta	25
Figura 7	–	Gráfico de resultados em um acesso a cada site	35
Figura 8	–	Gráfico representando total de Bytes baixados direto do servidor de origem e Bytes servido pelo cache	36
Figura 9	–	Gráfico de estimativa de economia de banda	37

LISTA DE TABELAS

TABELA 1	– Teste realizado com servidor Proxy web cache Squid	34
TABELA 2	– Avaliação de relevância nos acessos.	47
	Isto eh um teste	

LISTA DE SIGLAS

FTP	Transfer Protocol
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
LAN	Local Area Network
MBps	Mega bits por segundo
RFC	Request For Comments
TCP	Transfer Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
WAN	Wide Area Network
WWW	Word Wide Web

SUMÁRIO

1	INTRODUÇÃO	9
2	BASE DA PESQUISA: CONCEITOS, TÉCNICAS E MÉTODOS	11
3	FUNCIONAMENTO DE PROXY	16
3.1	VANTAGENS E DESVANTAGENS DO SERVIDOR PROXY WEB CACHE	17
3.2	ARQUITETURAS DE WEB CACHE	18
3.3	MODELOS DE IMPLANTAÇÃO	21
4	ESTUDO DE CASO	23
4.1	SUJEITOS DA PESQUISA	23
4.2	IMPLANTAÇÃO DO SISTEMA DE COLETA	24
4.3	SERVIDOR PROXY SQUID	26
5	ANÁLISE DE RESULTADOS	30
5.1	ANÁLISES DE LOG DE CACHE DO SQUID	30
5.2	APRESENTAÇÃO DOS RESULTADOS	32
6	CONCLUSÃO	38
	REFERÊNCIAS	39
	Anexo A – TESTES REALIZADO NOS SITES ACESSADOS	41
	Anexo B – AVALIAÇÃO DE RELEVÂNCIA NOS ACESSOS	47

1 INTRODUÇÃO

A algum tempo a Internet vem se apresentando como uma ferramenta largamente utilizada para acesso e disseminação de informações. Com isso passou a fazer parte do ambiente de trabalho de grande parte das empresas. Mas para que essa tecnologia possa ser utilizada de maneira adequada e sem desperdícios, uma série de ferramentas foram desenvolvidas e disponibilizadas para que, em conjunto, venham a oferecer possibilidades de melhorar o desempenho no tráfego da rede, bem como proporcionar formas de controle e direcionamento do uso da Internet. Um exemplo de ferramenta utilizada é o Proxy, trabalhando com *web cache* para melhorar o tempo de resposta no acesso a uma determinada página Web e controle de acesso visando garantir que os clientes (colaboradores) utilizem o meio (Internet) de forma consciente, acessível e possivelmente segura. O objetivo da pesquisa foi explorar a ferramenta de Proxy Squid e estudar situações em que esta possa ser aplicada como possível solução para evitar desperdício de banda, com também realizar controle de acesso dentro da prefeitura. Considerando que a principal dificuldade relatada pela administração, se tratando do uso da Internet, esta relacionada com a contaminação da rede por vírus (programas maliciosos), distrações com redes sociais e programas de entretenimento, consumo de banda com *downloads*, filmes *online*, acesso a rádios, entre outros. Neste contexto, a implantação de um serviço de Proxy poderia ser uma das possíveis soluções a ser aplicada. Um Proxy não garante tecnicamente segurança contra vírus, no entanto, pode auxiliar na proteção impedindo o acesso a *sites* que não fazem parte do contexto de trabalho da empresa e com isso a possibilidade de contaminação poderia ser menor.

A importância desta pesquisa está na necessidade em encontrar formas de utilização otimizada da capacidade da rede (Link) visando satisfação do usuário final, como também apresentar o Proxy como uma ferramenta de auxílio para o direcionamento correto dos recursos disponíveis para Internet.

A estrutura da pesquisa se divide em introdução e mais quatro capítulos. O segundo capítulo apresenta um levantamento bibliográfico sobre o tema em estudo, conceitos, técnicas e métodos que serviram de base para o projeto, bem com a importância da utilização de um

servidor Proxy para auxiliar no bom uso da Internet. No terceiro capítulo são expostas as formas de funcionamento do Proxy (explanação de figuras sem Proxy ativo e funcionamento da rede com Proxy ativo e realizando *web cache*), arquiteturas de Proxy e opções de utilização no ambiente de trabalho. O quarta capítulo apresenta o estudo de caso realizado, tendo como base para coleta de dados a Intranet da Prefeitura de Boa Ventura de São Roque. E para finalizar, o quinto capítulo mostra a análise dos dados obtidos, bem como a apresentação dos resultados.

2 BASE DA PESQUISA: CONCEITOS, TÉCNICAS E MÉTODOS

O termo Proxy vem do inglês e segundo Lunard (2005, p. 03) significa “procuração, tecnicamente Proxy é um software que tem a procuração de um ou mais *hosts* para buscar na Internet uma informação”. O Proxy é um *software* atuando em um computador intermediário, que possui como ferramenta a utilização de regras ou filtros, com a função de proibir ou liberar acessos a *sites*. Como definem Ricci e Mendonça (2006, p. 01), “Proxy refere-se a um software que atua como gateway de aplicação entre o cliente e o serviço a ser acessado, interpretando as requisições e repassando-as ao servidor de destino”.

Web cache é um sistema de computador em uma rede que mantém cópias de objetos (páginas HTML, imagens e arquivos), ou seja, páginas da Web recentemente solicitadas na memória ou no disco a fim de acelerar a recuperação. Essa funcionalidade pode evitar a necessidade de um novo acesso a um servidor remoto em busca da mesma informação. Como observa-se através de Ramana e Srinath (2002, p. 03):

A principal função de um *web cache* é armazenar localmente, páginas da Web acessadas com frequência, tornando o acesso à Web mais rápido. Ele acumula os pedidos e envia um único pedido individual em seu lugar para o servidor de destino. Assim, Web caches podem ajudar a aliviar a carga em um servidor Web, reduzindo o número de solicitações recebidas.

Como visto anteriormente, Proxy e *web cache* possuem funções distintas. Enquanto o Proxy é utilizado para aprimorar a segurança na rede, o *web cache* faz seu trabalho possibilitando que o acesso a uma página Web seja feito de maneira ágil, porém, quando trabalhando em conjunto podem oferecer a possibilidade de melhoria, tanto na segurança quanto na redução do uso da banda disponível.

Um problema comum encontrado onde há um número considerável de máquinas (computadores) acessando simultaneamente a Internet, é o congestionamento da rede. Ryu et al. (2003) atribui suas causas e consequências:

O congestionamento é a necessidade agregada de largura de banda que excede a capacidade disponibilizada pela linha. Dessa forma, causa a degradação de desempenho

ocasionando perda múltipla de pacotes, baixa utilização do meio, tempo de atraso alto e colapsos.

Um dos efeitos do congestionamento que pode ser mais perceptível ao usuário da Internet é o aumento da latência ¹ no retorno de uma página Web. A utilização de um servidor Proxy com *web cache* permite armazenar objetos da Web (Internet) acessados frequentemente, logo, uma nova requisição ao mesmo objeto será atendida pelo servidor Proxy, diminuindo assim, o tráfego de informações para rede externa e conseqüentemente a latência.

O Proxy *web cache* posicionado entre a rede interna (Intranet) e a rede externa (Internet) acumula os pedidos de objetos da Web e encaminha um único pedido para o servidor de destino. Logo, ao receber os dados solicitados, salva uma cópia no *cache* e repassa ao solicitante. Quando implantado em lugares estratégicos na rede (LAN ou WAN), como um *gateway*, além de reduzir o tráfego da rede pode proporcionar um retorno mais rápido das páginas Web. Essa possibilidade se deve ao fato dos objetos armazenados em *cache* estarem na rede local, ou seja, mais próximos do usuário final em relação aos servidores de origem. Em se tratando de Proxy *web cache*, Danalis e Marckatos (2002, p. 235) inferem que estes, “[...] devem ser colocados em posições estratégicas na rede, a fim de maximizar a sua eficácia. Dependendo do resultado pretendido, um web cache deve ser implantado perto do servidor, na porta de entrada de uma rede de área local, ou ao lado do cliente”.

O *web cache* pode estar presente em diversos pontos durante o processo de navegação, Sulaiman et al. (2008, p. 02), apresentam alguns locais onde *web cache* podem ser encontrado, como por exemplo, “o browser de um usuário, o disco rígido do usuário, servidores localizados na instituição em que é empregado o usuário, provedor da instituição Internet Service (ISP), o centro regional de Internet, o centro nacional de Internet ou no nível global.”

A aplicabilidade de *web cache* pode ser feita nos navegadores da Web (cliente), no Proxy *web cache* (cliente e servidor) e nos servidores de Web (reverso). No cliente o *web cache* é incorporado ao browser, devido ao fato deste tipo de *web cache* utilizar apenas parte da memória principal ou um pequeno espaço no disco para armazenamento.

¹De maneira geral, a latência da rede pode ser entendida como o somatório dos atrasos impostos pela rede e equipamentos utilizados na comunicação.

Vale Destacar que o *cache* é embutido no navegador, não podendo assim, ser compartilhado por outro usuário. Os autores Sulaiman et al. (2008, p. 02), citam as utilidades desta forma de cache:

Esse cache é útil, especialmente quando os usuários apertam o botão "back" ou clicam em um link para ver uma página que acabaram de olhar. Além disso, se o usuário usa as mesmas imagens de navegação em todo o navegador, eles serão servidos a partir de caches que navegam quase que instantaneamente.

Cliente/Servidor: o *web cache* encontra-se em um servidor Proxy entre o cliente e o servidor de conteúdos Web, pois visa reduzir a largura de banda necessária em conexões de Internet, além de permitir o compartilhamento dos recursos. Novamente Sulaiman et al. (2008, p. 02) comentam que:

Este tipo de cache pode ser usado para armazenar um número significativo de arquivos em muitos servidores web e, com um benefício adicional, permite que muitos usuários compartilhem os recursos dos servidores mais próximos de Proxy ou seus caches vizinhos.

Em um sistema cliente servidor, o serviço de *web cache* age como um intermediário em requisições de conteúdos Web realizados por um usuário da rede interna, sendo assim, quando uma página Web é solicitada pela primeira vez, o servidor Proxy acessa o servidor de origem, faz o *download* da mesma e armazena em *cache*, disponibilizando-a para todos os usuários da rede interna. De acordo com Sulaiman et al. (2008, p. 02), o *web cache* implementado em um servidor Proxy:

[...] funciona com o mesmo princípio de *cache* do navegador, mas em uma escala muito maior. Ao contrário do cache do navegador que lida com apenas um único usuário, os servidores proxy podem lidar com centenas ou milhares de usuários da mesma forma.

Servidores Web: nesse caso é chamado de servidor de *web cache* reverso, fica logicamente posicionado em frente ao servidor Web, para tratar as requisições antes mesmo que estas cheguem ao seu destino. Sua função é aliviar a carga ocasionada por múltiplos acessos acumulando pedidos vindo de clientes para um mesmo objeto e repassando apenas um ao servidor Web. Em contraste com os tipos de *cache* citados anteriormente, o *cache* reverso responsabiliza-se apenas com objetos do servidor Web. Os autores Waleed et al. (2011, p. 04), defendem que, "mesmo no servidor de origem, as páginas web podem ser armazenadas em um cache do lado do servidor para reduzir a necessidade de cálculos redundantes ou recuperações de banco de dados. Assim, a carga do servidor pode ser reduzida".

Através da idéia de Sulaiman et al. (2008, p. 04), observa-se que os servidores *web cache* possuem um conjunto de normas que determinam a disponibilidade do *cache*:

Todos os caches tem um conjunto de regras que eles usam para determinar quando servir um objeto a partir do cache se estiver disponível. Algumas dessas regras são definidas nos protocolos (HTTP 1.0 e 1.1), e outras são definidas pelo administrador do cache (ou o usuário do cache do navegador, ou o administrador do Proxy).

No caso de um *web cache*, o servidor recebe uma requisição Web, analisa e executa as ações necessárias para responde-la. Caso o conteúdo seja encontrado no *cache*, será realizado uma verificação prévia quanto à compatibilidade em relação ao conteúdo original do servidor. Se nada foi alterado, o conteúdo será entregue ao requisitante, caso contrário, será feito um novo *download* e atualização do *cache*, e por fim, se o conteúdo não estiver em *cache* será realizada uma nova requisição ao servidor de páginas Web original, desta vez como um pedido próprio.

Um servidor Proxy *web cache* trabalha com armazenamento de objeto, seja na memória ou no disco rígido, sendo assim, o espaço reservado para tal função apresenta importância significativa para um bom desempenho. A manutenção de objetos em *cache* garante que o espaço seja ocupado com conteúdos realmente utilizados pelos usuários, assim sendo, os servidores de *web cache* fazem uso de algoritmos de substituição, que trabalham considerando vários fatores, entre eles o número de acessos, tempo de vida e o tamanho do objeto. De acordo com Yeu e Fedel (2014, p. 05):

Least Recently Used (LRU) — menos utilizado recentemente — Mantém o cache dos objetos referenciados recentemente, remove do cache o objeto referenciado há mais tempo. Problema: não leva em conta o tamanho do objeto – vários objetos pequenos podem substituir um objeto grande.

Least Frequently Used (LFU) — menos frequentemente utilizado. Remove o objeto menos popular, porém não considera tempo do último acesso. Considera o número de acessos. Problema: o cache pode ficar com objetos muito acessados e velhos.

First in First out (FIFO) - Objetos são removidos na mesma ordem que entram. Não é muito usado em servidores cache.

Size - Baseado no tamanho do objeto; Substitui primeiro objetos com grande tamanho. Problema: o cache pode ficar preenchido com objetos antigos.

Greed Dual Size – GDS. Atribui valores baseados no custo de um hit para os objetos armazenados no cache. O custo pode ser latência ou pacotes transmitidos pela rede. São mantidos em cache objetos menores.

Observa-se que *web cache* é um exemplo de aplicação de *cache* com a função de, armazenar as páginas da Web acessadas frequentemente na rede local e tem como objetivo melhorar o desempenho dos sistemas baseados na Web, armazenando e reutilizando objetos já pesquisados e que possam ser usados futuramente em uma nova pesquisa, dessa forma o acesso a esses objetos poderá ser mais rápido. Danalis e Marckatos (2002, p. 235), enfatizam

que, “além de reduzir o tráfego de saída através do agrupamento de pedidos duplicados de navegadores, os caches Web agem como tutores Web, auxiliando no envio do tráfego da Web de forma eficiente através de uma rede”.

Pesquisas realizadas mostram os benefícios que podem ser alcançados pelos usuários, pois a partir da aplicação de *web cache* durante a navegação, muitos congestionamentos na rede podem ser evitados. Segundo o Laboratório Nacional de Applied Research Network (NLNR) (apud RAMANNA, SRINATH, 2002) “estatisticamente falando, um *web cache* poderia eliminar pelo menos 30% do tráfego da Web, que normalmente sairia de uma rede de área ampla (WAN)”. De uma maneira mais abrangente, os benefícios potenciais de um *web cache* podem ser notados em diversos pontos, Zeng et al. (2004, p. 272) comentam que:

Do ponto de vista do usuário final, o *cache* pode reduzir significativamente a latência da rede e melhorar a sua experiência web. Do ponto de vista da infra-estrutura de Internet, *cache* pode reduzir a quantidade de tráfego Web. Como um resultado disso, o desempenho global da Internet pode ser melhorado e o congestionamento da rede minimizada. Além disso, para as empresas que pagam os ISPs para largura de banda de rede de área ampla com base na quantidade de tráfego, redução de meios de trânsito reduz os custos. Do ponto de vista do servidor Web, o *cache* pode reduzir significativamente a cargas do servidor e melhorar a capacidade de resposta do sistema.

Existem diversas maneiras de verificar a eficiência da utilização de um servidor Proxy *web cache*, entre elas podemos citar, taxa de acerto e quantidade de *bytes* servidos pelo *cache*. Nesse contexto Zeng et al. (2004, p. 271) inferem que:

Na literatura de caching, existem duas medidas de desempenho principais, taxa de acerto de *cache* e *cache hit byte*, usada para avaliar os sistemas de armazenamento em *cache* da Web. Taxa de acerto de *cache* é calculado como o número de objetos da Web solicitadas pelo usuário que são atendidas pelo *cache* dividido pelo número total dos objetos pedidos. Taxa de sucesso de *cache byte* é definido como o número de *bytes* atendidos a partir de conteúdo em *cache* dividido pelo número total de *bytes* servido.

Outras formas de medir o desempenho e ajudar a determinar o impacto do *cache* na rede, poderia ser através de análise estatística sobre a utilização da rede e taxas média/máxima de consumo de largura de banda.

3 FUNCIONAMENTO DE PROXY

Quando um cliente, através do navegador busca por uma determinada URL², esta é traduzida para o IP (*Internet Protocol*) da máquina (servidor Web) que hospeda a página, sendo iniciada uma sessão entre o cliente e o servidor, e então o arquivo correspondente a URL é transferido. Como mostra a figura 1, esse procedimento se repete toda vez que uma página é requisitada.

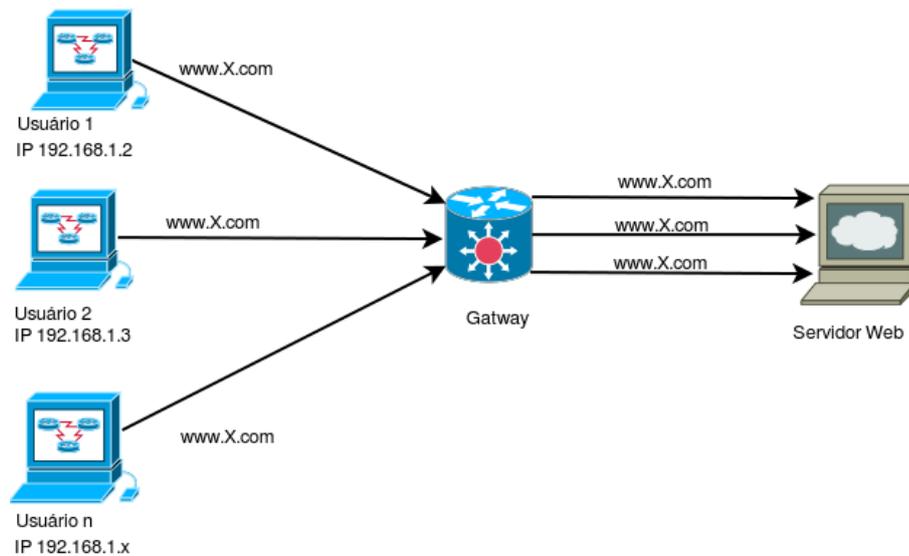


Figura 1: Funcionamento da rede sem utilização de Proxy

Conforme a figura 1, nas estações de trabalho que precisam acessar um *site* da Internet, as requisições de páginas Web são enviadas diretamente ao servidor e respondidas diretamente para endereço IP das estações solicitantes. Observa-se que as estações encaminham o pedido diretamente ao servidor Web, ao contrário do que ocorre com Proxy com *web cache* ativo. O servidor Proxy fica entre a rede Intranet e a Internet e atua como um procurador, assim toda requisição de acesso feita por um cliente (computador) da Intranet será direcionada ao Proxy e este fará o pedido ao servidor que hospeda a página ou documento requisitado. Assim somente um computador (servidor Proxy) terá acesso direto à rede externa. Diante disso, um

²URL - Uniform Resource Locator, localizador que permite identificar e acessar um serviço na Web .

filtro analisa as requisições do solicitante e define a necessidade de efetuar o direcionamento do Proxy, ou seja, ele controla o serviço de acesso, não permitindo que o cliente e o servidor público interajam diretamente. A figura 2 demonstra como funciona uma requisição através de um servidor Proxy *web cache*.

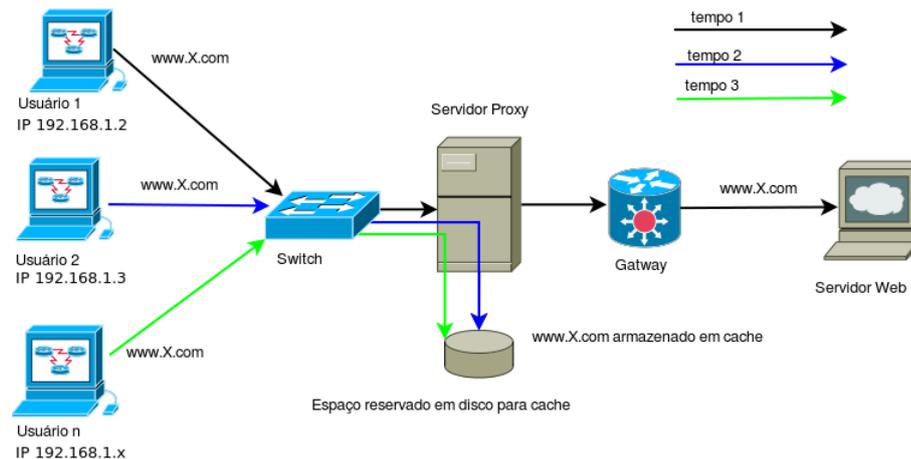


Figura 2: Funcionamento da rede com utilização de um servidor Proxy *web cache*

De acordo com um artigo publicado pela Revista “The Internet Protocol Journal”, Tels-tra (1999, p.3):

Se o *cache* contém a URL referenciada é verificado a validade em relação ao tempo de vida do objeto comparando com o “expira:” campo de data do conteúdo, se existir, ou por alguns métodos definidos localmente. Objetos obsoletos são revalidados com o servidor, e se o servidor revalida o conteúdo, o objeto é observado como atual. Objetos atuais são entregues ao cliente como um conteúdo do *cache*.

O Proxy no ambiente servidor recebe requisições das estações de trabalho para conexão a Internet, o papel do Proxy é buscar informações primeiramente no *cache* local, caso não encontre, fará a busca no *site* solicitado pela estação de trabalho. Quanto aos pedidos posteriores para o mesmo objeto, o Proxy *web cache* oferece o objeto de seu armazenamento em vez de passar o pedido para o servidor original, caso ele não esteja disponível, o Proxy tentará buscar o objeto diretamente.

3.1 VANTAGENS E DESVANTAGENS DO SERVIDOR PROXY WEB CACHE

Existem vantagens significativas em utilizar uma rede com *web cache*. Em primeiro lugar, demonstra-se que os documentos de *cache* podem melhorar o desempenho da Web significativamente. Reddy (2011, p. 3) comenta que as principais vantagens do uso de servidor *web cache* são:

1. Cache Web reduz o consumo de largura de banda, assim, diminui o tráfego na rede e diminui o congestionamento da rede.
2. Cache Web reduz a latência de acesso devido a duas razões:
 - a) documentos acessados são buscados nos caches do proxys em vez de servidores de dados remotos, o prazo de transmissão é minimizado.
 - b) Por causa da redução do tráfego de rede, documentos não encontrado em cache também pode ser recuperado relativamente mais rápido do que sem cache devido à menor congestionamento ao longo do caminho e menos carga de trabalho no servidor.
3. Cache Web reduz a carga de trabalho do servidor Web remoto, através da divulgação de dados entre as caches de proxy através da rede de área ampla.
4. Se o servidor remoto não está disponível devido a falha do servidor remoto ou de particionamento da rede, o cliente pode obter uma cópia em cache no proxy. Assim, a robustez do serviço Web é reforçada.
5. Um efeito colateral de cache Web é que ele oferece-nos uma oportunidade para analisar os padrões de uso de uma organização.

Contudo, o uso de *web cache* também possui algumas desvantagens, isto está relacionado com a maneira que o mesmo é configurado ou posicionado fisicamente na rede, a falta de planejamento nestas etapas poderá ocasionar falhas, e dentre as mais significativas, Reddy (2011, p. 4) cita que:

1. A principal desvantagem é que um cliente pode estar a olhar para os dados obsoletos devido à falta de atualização adequada.
2. A latência de acesso pode aumentar no caso de um cache miss (conteúdo não encontrado no cache) devido ao processamento extra. Assim, a taxa de acerto de cache devem ser maximizados e os custos de um cache miss deve ser minimizada ao projetar um sistema de cache.
3. Um proxy cache único é sempre um gargalo. Um limite deve ser definido para o número de clientes que um proxy pode servir. Uma eficiência limite inferior (isto é, o sistema de proxy deveria ser pelo menos tão eficiente como meio de contato direto com os servidores remotos)
4. Um único proxy é um ponto único de falha.
5. Usando um proxy cache irá reduzir os hits (busca) no servidor remoto original, que pode decepcionar um monte de provedores de informações, uma vez que eles não podem manter um verdadeiro registro dos acessos a suas páginas. Assim, eles podem decidir não permitir que os seus documentos sejam armazenados em cache.

3.2 ARQUITETURAS DE WEB CACHE

A arquitetura de um *web cache* é definida como a forma que um servidor Proxy interage com um sistema de arquivos, neste caso, arquivos Web. As formas podem ser apresentadas como hierárquica, distribuída, híbrida, entre outras. Na forma hierárquica, o *web cache* é colocado em vários níveis na rede, de acordo com a necessidade identificada, ou seja, neste modelo de implementação a requisição a um objeto (página Web) partindo de um cliente procura primeiramente no *cache* do navegador, se não encontrada então segue até o servidor Proxy local, se este também não possui o objeto solicitado, encaminha para o servidor do próximo nível, se o objeto não for encontrado no último servidor da hierarquia, este se responsabiliza em fazer uma

requisição para o servidor de origem do objeto, e em seguida enviar para todos os servidores de nível inferior, onde será armazenado uma cópia e repassado para o cliente que a solicitou. Sobre o modo de implantação hierárquica Reddy (2011, p. 6) explica que:

Se o documento não for encontrado em qualquer nível de *cache*, o *cache* de último nível contata diretamente do servidor original. Quando o documento for encontrado, seja em um *cache* ou no servidor original, que viaja para baixo na hierarquia, deixando uma cópia em cada um dos *caches* intermediários ao longo de seu caminho. Outras solicitações para o mesmo documento viajam através da hierarquia de *cache* até que o documento é atingido em algum nível de *cache*

A figura 3 apresenta o modelo de arquitetura hierárquica.

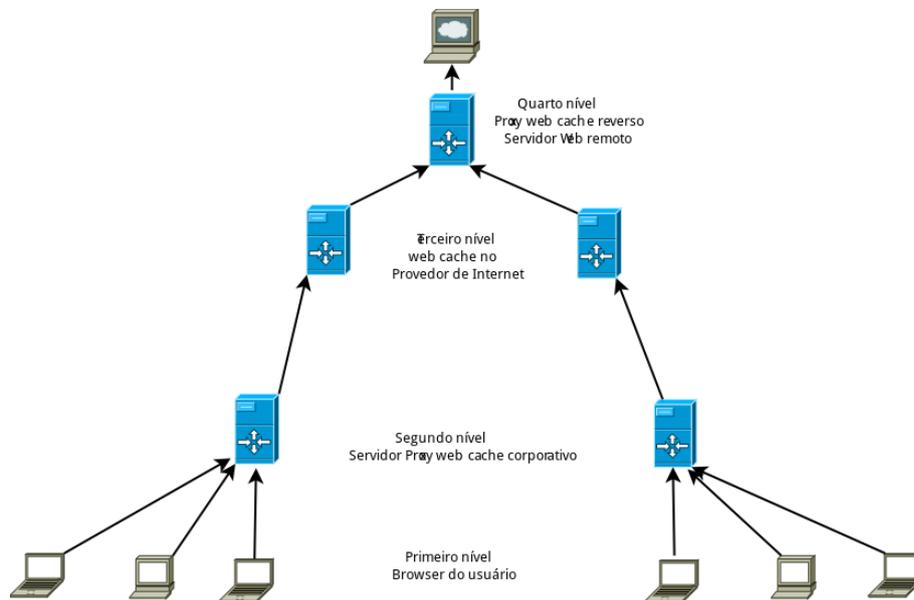


Figura 3: Modelo de arquitetura hierárquica

No modelo de arquitetura distribuído não existem servidores de *web cache* em níveis intermediários, apenas nas extremidades da rede. A cooperação acontece entre os servidores de modo a reduzir a redundância de objetos armazenados. Isso é possível porque cada servidor contém um resumo do conteúdo em *cache* dos servidores vizinhos e mantém o endereço do servidor que contém o objeto solicitado pelo cliente, ou seja, se um objeto é solicitado e o mesmo não está presente no *cache*, o servidor repassa, com base no resumo, ao servidor que detém o objeto em *cache*.

Essa abordagem implica no aumento do tráfego na rede entre os servidores. Sobre os mecanismos de partilha de conteúdo nessa arquitetura Mukesh e Charanjit (2013, p. 3), inferem que:

Caches Institucionais podem transferir as consultas aos seus caches institucionais de cooperação para todos os cooperantes locais. Isso tudo é feito usando Inter Cache Protocol (ICP). Mas esta abordagem aumenta o consumo de largura de banda e latência. Caches institucionais podem manter o resumo do conteúdo de seus caches cooperantes. Ele elimina a necessidade de consultar os caches cooperantes. Esses resumos são trocados periodicamente. Para uma distribuição eficaz de síntese uma estrutura hierárquica de nós intermediários pode ser aplicado, apenas distribuindo a informação sobre a localização dos documentos e não os documentos propriamente dito

A figura 4 mostra de maneira resumida o funcionamento de um sistema de Proxy distribuído, onde existem três redes, cada uma com um servidor Proxy individual, porém cooperando entre si.

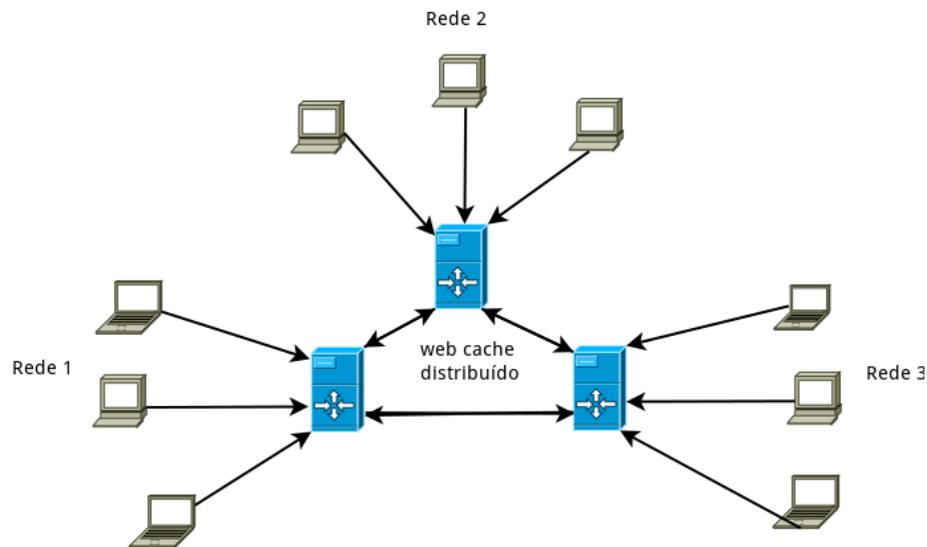


Figura 4: Modelo de arquitetura distribuído

A arquitetura híbrida é uma combinação de esquema de *cache* hierárquico e distribuído. Assim pode haver cooperação entre os servidores *cache* tanto do mesmo nível quanto de níveis superiores. Essa abordagem trabalha para resolver os problemas encontrados nas duas arquiteturas vistas anteriormente e fornece aos servidores a capacidade dinâmica para lidar com excessos de pedidos, dessa forma o servidor pode repassar um pedido a outro, caso esteja com sobrecarga. Novamente Mukesh e Charanjit (2013, p. 03) falam que:

O comportamento dinâmico de servidores proxy para lidar com Cache cooperantes com base em suas cargas ou seja, se um servidor proxy está congestionado pode negar mais pedidos. Esses pedidos serão tratadas por alguns outros servidores proxy no mesmo ou em outro nível. Ele irá resultar em tempo reduzido de conexão e transmissão e também menor latência na recuperação.

A possibilidade de transferência de pedido entre os servidores vem a trabalhar em favor do balanceamento de carga. Mukesh e Charanjit (apud GADDE et al., 1997, p. 95) apontam para

uso da arquitetura híbrida como uma possível solução para lidar com atrasos e desconexões frequentes de servidores Proxy.

3.3 MODELOS DE IMPLANTAÇÃO

Existem maneiras diferenciadas de se utilizar os benefícios de um *web cache* e cada uma é aplicada de acordo com a necessidade de cada usuário, seja em uma empresa com diversas máquinas (computadores) ou um escritório com apenas alguns usuários. A utilização de um servidor Proxy *web cache* pode ocorrer de maneira explícita, onde o cliente (usuário de um computador da rede) necessita configurar manualmente seu navegador, para que o mesmo faça uso do servidor Proxy nas requisições de páginas Web. Essa forma de uso deixa o cliente a vontade para decidir se fará uso do *web cache* presente no servidor Proxy, ou encaminhará a solicitação diretamente para o servidor Web. Pode se observar na publicação da revista "The Internet Protocol Journal", Telstra (1999, p. 8) que em relação a forma explícita de uso:

Alguns sistemas de *proxy cache* são implementados como uma opção habilitado pelo usuário, no qual o usuário indica um servidor de *cache* para o navegador como um agente de proxy, e o navegador em seguida, dirige todas as solicitações da Web para o cache proxy. Em qualquer etapa, o usuário pode instruir o navegador para desativar o uso do *proxy cache*, e solicitar o navegador para realizar a transação diretamente como cliente.

A forma explícita de uso pede que o usuário esteja ciente dos benefícios na utilização de um *web cache*, além da familiarização com as configurações necessárias. Outro modo de utilização chamada de explícita forçada, onde, semelhante a vista anteriormente exige uma configuração no navegador do usuário, porém não dá a possibilidade de escolha entre usar ou não o servidor Proxy *web cache*, pois uma vez desabilitado o direcionamento do navegador, o usuário ficará impossibilitado de realizar a navegação na Internet. Sobre essa forma Telstra (1999, p. 8) fala que:

Aqui, todos os tráfegos Web na porta TCP 80 (a porta usada pelo protocolo de transporte Web HTTP) é impedido de acesso de saída direta, e os clientes são forçada a configurar seus navegadores para usar o *cache* do provedor para acesso externo à Internet. Esta técnica é vulgarmente denominado *cache* forçado.

Este modelo é útil quando além de fazer uso de *web cache* para economia de banda, se deseja fazer um controle mais acirrado sobre os usuários. Isso é possível aplicando autenticação, onde será exigido um usuário e uma senha previamente cadastrado para iniciar uma navegação na Internet. A terceira forma de uso é a transparente, que como o próprio nome sugere é transparente para o usuário, ou seja a navegação ocorre através do servidor Proxy sem que o usuário perceba. As requisições são direcionadas automaticamente para o Proxy por meio de uma regra aplicada no *firewall*. Telstra (1999, p. 08) faz algumas afirmações em torno do modelo transparente:

Como o *cache* é transparente ao usuário do navegador, o mesmo pode não estar explicitamente consciente de que o *cache* está sendo realizado durante o processamento de solicitações do usuário. Aqui, a rede irá interceptar mensagens HTTP destinado a servidores Web remotos, e apresentar estes pacotes para o *proxy cache* . Uma vez que a página é localizada, o *proxy cache* deve, então, responder ao solicitante original assumindo a identidade do destino original.

Esse tipo de aplicação implica em um estudo mais detalhado das configurações para melhor atender aos clientes, haja visto que um servidor Proxy *web cache* configurado incorretamente poderia ter um efeito contrário se tratando de melhoria no tráfego para rede externa.

4 ESTUDO DE CASO

Os dados que serão apresentados fazem parte da pesquisa de campo realizada na sede da Prefeitura de Boa Ventura de São Roque - PR.

4.1 SUJEITOS DA PESQUISA

Localizado na região central do Estado do Paraná, o município de Boa Ventura de São Roque teve sua emancipação em 18 de setembro de 1996. Hoje possui cerca de sete mil habitantes e tem sua economia baseada na agricultura e pecuária, além de algumas indústrias. Devido ao volume de dados movimentados diariamente e por não possuir no momento nenhum sistema que trate de otimização de uso de banda disponível ou mesmo controle de acesso, a sede da Prefeitura localizada na Rua Moises Miranda nº 402, foi utilizada como base de coleta de dados para análise, a fim de observar se a implantação de um servidor Proxy com *web cache*, poderá de alguma forma proporcionar economia, seja no uso da banda disponível ou através de bloqueio de *sites*.

O fluxo básico de dados da prefeitura está relacionado com *downloads* e *uploads* de arquivos e relatórios para órgãos responsáveis, servidor de aplicação que pode ser acessado pelas secretarias, disponibilização de editais, acesso ao portal da transparência, além de outras atividades administrativas. Para acesso a Internet a prefeitura conta com um link de fibra ótica de 20 Mbs da Copel.

A metodologia de coleta utilizada foi a captura de pacotes que partiam dos clientes (computadores) da rede interna com destino à Internet. O objetivo da captura foi a constuição de um posterior ranking dos *sites* mais acessados em um determinado período de tempo (dias). Em seguida com base no ranking desenvolveu-se uma análise em laboratório com a finalidade de apurar o custo em Bytes de cada *site* acessado para o link da Prefeitura. Posteriormente uma nova análise com o uso de um servidor Proxy com *web cache* foi realizada para verificar através de comparativo a existência de vantagens em relação à primeira análise.

No momento da coleta, a sede da prefeitura contava com 30 (trinta) estações fixas (computadores) de trabalho, além de notebooks de empresas visitantes que se conectam a rede durante negociações. Na figura 5 pode se observar a topologia física encontrada antes da implantação do sistema de coleta.

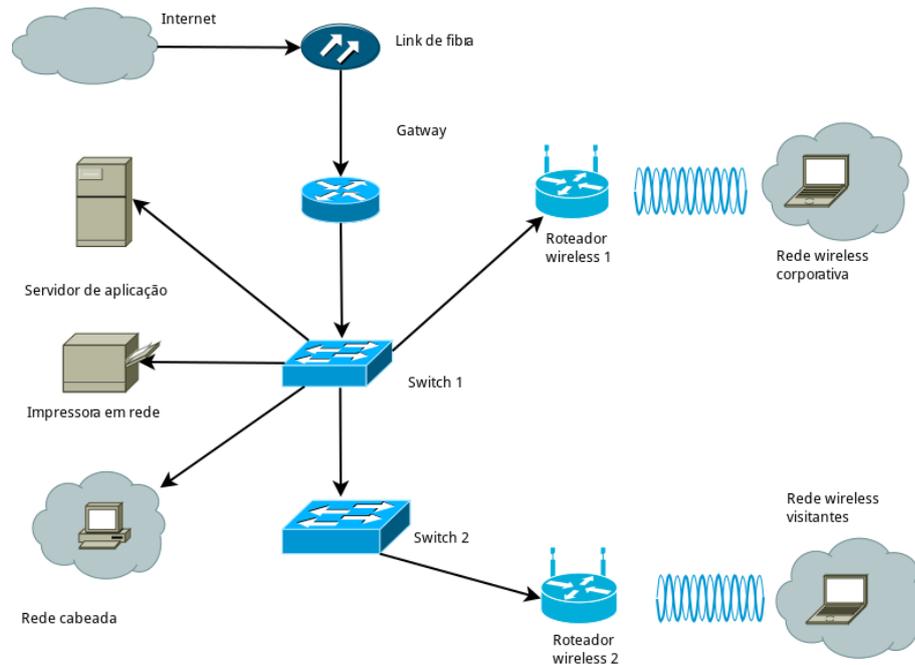


Figura 5: Rede física antes da implantação do sistema de coleta

4.2 IMPLANTAÇÃO DO SISTEMA DE COLETA

Para que fosse possível a captura de todo tráfego da rede, foi posicionada uma máquina (computador) com duas placa de rede (eth0 e eth1) entre o *switch* principal e o receptor ótico, atuando como uma ponte entre a Internet e a Intranet. O sistema operacional utilizado na máquina ponte é o Ubuntu 10.4 que, além de gratuito, oferece as funcionalidades necessárias para o propósito, como por exemplo, *ehtables* e *tcpdump*.

Segundo Sand e Kleber (2013, p. 10), “*ehtables* atua na camada de enlace, permitindo configurar e manter as tabelas de regras que inspecionam os quadros Ethernet. As regras instaladas pela *ehtables* permitem a filtragem transparente do tráfego de rede enviado através da Linux bridge”. Já o *tcpdump* é uma ferramenta de análise de tráfego na rede, como disposto no site oficial *Tcpdump* (2013), “*tcpdump* é um analisador de pacotes de linha de comando”.

Essas duas ferramentas (*ehtables* e *tcpdump*) foram utilizada possibilitando a coleta de dados para a presente pesquisa, como já mencionado anteriormente. O *ehtables* em conjunto

com a bridge³ brctl, forneceu a possibilidade de criar uma ponte ligando duas redes, neste caso eth0 e eth1. Sendo assim, a figura 6 apresenta a topologia depois de implantado o sistema de coleta.

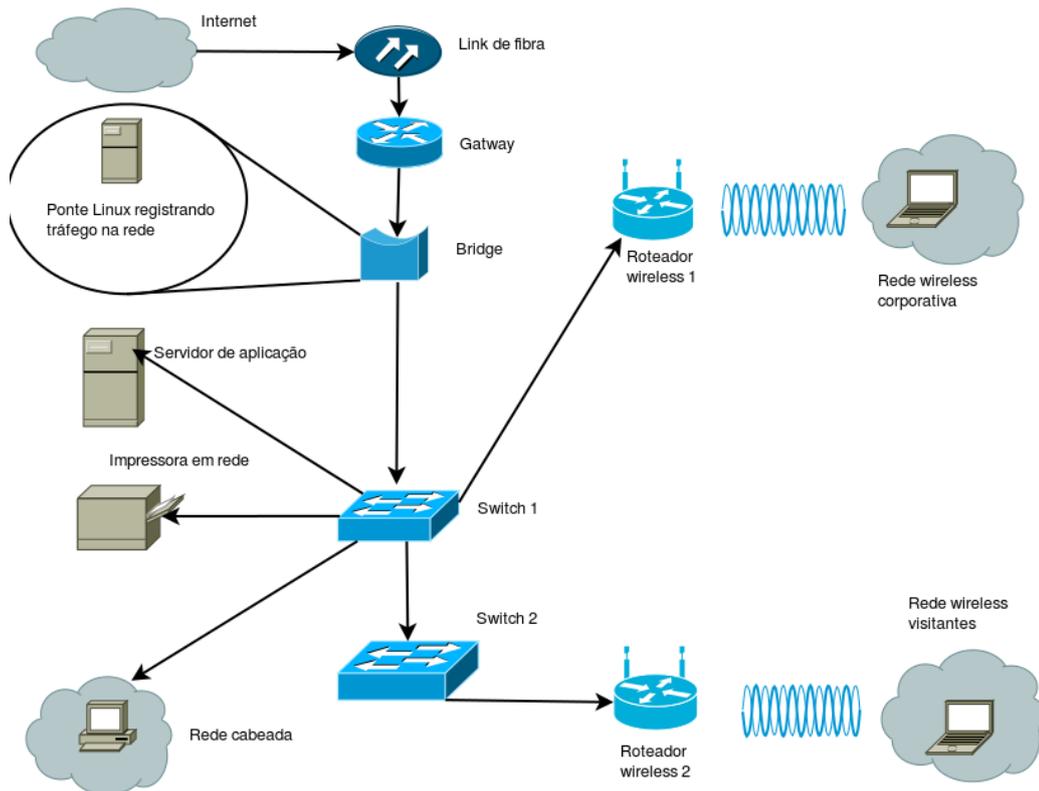


Figura 6: Rede física após a implantação do sistema de coleta

Para habilitar o tráfego através da bridge possibilitando a captura de pacotes as seguintes linhas de comando foram adicionadas:

```
brctl addb br0 (cria uma bridge).
brctl addif br0 eth0 (cria uma interface (eth0) na bridge).
brctl addif br0 eth1 (cria uma interface (eth1) na bridge).
ifconfig br0 up (ativa a bridge recém criada).
ifconfig br0 192.168.1.246 netmask 255.255.255.0 up (colocando um ip na bridge).
route add default gw 192.168.1.1 dev br0 (adiciona uma rota ligando a Internet à Intranet )
```

Com essa sequência de comandos, criou-se e habilitou-se uma bridge (ponte) por onde

³Bridge: dispositivo que conecta duas ou mais redes de computadores transferindo, seletivamente, dados entre ambas

irá passar todo tráfego da rede da prefeitura. Depois disso foi usado ngrep para capturar e salvar em um arquivo pcap o tráfego que passa pela br0 , utilizando se as seguintes linhas de comando:

```
ngrep -O nome_documento.pcap -t -d br0 "(GET|POST)" tcp port 80.
```

Observou-se a presença de uma expressão regular “(GET|POST)”, que indica a captura de pedidos realizados por GET ou POST).

4.3 SERVIDOR PROXY SQUID

O Squid caracteriza-se em um software especializado e robusto, que faz operação de Proxy, *web cache* e FTP (protocolo de transferência de arquivos). Ele pode ser usado em vários sistemas operacionais, incluindo Windows e está licenciado sob a GNU GPL. Utiliza-se o Squid para o acesso a protocolos como HTTPS (acesso a sites que utilizam uma camada adicional de segurança), HTTP (acesso a sites) e FTP. Squid oferece um rico conjunto de opções de otimização de tráfego, a maioria dos quais são ativadas por padrão para a instalação simples e de alto desempenho. O Squid age de forma a aperfeiçoar o fluxo de dados entre cliente e servidor para melhorar o desempenho e armazenar em *cache* conteúdo frequentemente usado para economizar largura de banda. De acordo com o site oficial squid-cache (2014):

Os sistemas Squid, estão sendo executados em uma taxa de sucesso de aproximadamente 75% , efetivamente quadruplicando a capacidade dos servidores Apache por trás deles. Isto é particularmente visível quando uma grande onda de tráfego chega direcionado para uma página particular, através de um link da web a partir de outro site, como a eficiência de *cache* para essa página será quase 100%.

O Squid pode oferecer ainda diversas possibilidades de configurações que podem auxiliar no gerenciamento da rede, dentre elas:

- **Autenticação** – onde para poder fazer uso da Internet o usuário terá que se autenticar através de usuário e senha. Com isso pode se fazer um controle mais acirrado dando ou não permissão de acesso para determinado grupo de usuários dentro de uma empresa.
- **Registro de Logs de Acessos** – tudo que for acessado passando pelo proxy poderá ser arquivado, isso poderá inibir muitas tentativas de acessos indevidos por parte dos usuários além de facilitar o trabalho em uma possível auditoria.
- **Controle centralizado** - com o uso do Proxy em um lugar estratégico, a rede passará a ter um ponto centralizado de acesso à Internet, o que torna a gerência da rede mais fácil

e eficiente. Uma única máquina é capaz de prover acesso a várias outras. O fato de uma única máquina possuir acesso a rede externa(Internet) poderá contribuir para segurança e todos os esforços poderão ser concentrados na máquina servidor.

Outra funcionalidade presente no Squid que pode ser explorada é a opção de controle de acesso. Essa função possui bastante flexibilidade e pode ser implantada através da definição de regras de acesso (ACL, Access Control List). De acordo com Lunard (2005, p. 31).

As Listas de Controle de Acesso (ACL) permitem especificar endereços de origem e destino, domínios, horários, usuários, portas ou métodos de conexão ao Proxy, que servirão de base para permitir ou negar o acesso baseando-se em conjunto dessas ACLs. Isto permite uma grande flexibilidade na configuração do SQUID.

As ACLs assim como as demais configurações necessárias, devem ser definidas no arquivo de configuração squid.conf. Neste arquivo são definidas praticamente todas as suas configurações, modo de operação, normal (configurado no browser) ou transparente, espaço em disco disponibilizado para *cache* arquivos de log parâmetros de autenticação. Um exemplo de uso de uma ACL para bloqueio de um site:

```
acl sites_bloqueados src "/etc/squid/sites_bloqueados.txt"
```

Lembrando que o arquivo `sites _ bloqueados .txt` deve conter deve conter todas as URLs dos sites que desejamos bloquear.

```
http_access deny bloqueio
```

A opção `deny` diz para o Squid bloquear os sites que estejam na regra acima. A seguir serão apresentadas as principais TAGs de configuração para que o Squid possa funcionar. Esta configuração é apenas um exemplo e não pode ser tomada como definitiva.

```
hierarchy_stoplist cgibin ?
```

Esta opção vem habilitada por padrão no squid.conf e refere-se a conteúdo dinâmico, ou seja, conteúdos que mudam rapidamente e por isso não é interessante mante-los em *cache*. Se a URL contém o padrão aqui especificado, o Squid irá buscar o conteúdo no servidor de origem.

```
acl QUERY urlpath_regex cgibin ? no_cache deny QUERY
```

Com esta ACL o Squid entende que não deve armazenar em *cache* conteúdos dinâmicos.

```
cache_mem 64 MB
```

Aqui está sendo definida a quantidade de memória disponibilizada apenas para manipulação dos arquivos em trânsito, não sendo esta utilizada para *cache*.

```
cache_dir ufs /var/cache/squid 300 64 64
```

Esta TAG determina em qual diretório será armazenado os arquivos de cache e também define o espaço em disco que pode ser ocupado por ele, neste caso o diretório de cache é */var/cache/squid*, com o tamanho de 300 MB, tendo 64 diretórios com 64 sub diretórios em cada um deles.

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/passwd—
auth_param basic children 5
auth_param basic realm Servidor Proxy ARL
auth_param basic credentialsttl 2 hours
```

TAG's referentes ao processo de autenticação.

```
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern . 0 20% 4320
```

Estas TAGs são usadas pelo Squid para definir o tempo de vida dos objetos em *cache* e baseada nelas o Squid sabe se o objeto em *cache* deve ser revalidado antes da apresentação ao cliente solicitante.

```
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
Safe_ports port 443 563 # https, snews
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # httpmgmt
acl Safe_ports port 488 # gsshttp
acl CONNECT method CONNECT
```

Estas ACL's fazem parte da configuração padrão do Squid e é o mínimo recomendável para seu uso, não sendo necessária nenhuma alteração nas mesmas.

5 ANÁLISE DE RESULTADOS

Após observar e capturar o tráfego da rede na prefeitura, pode-se montar um cenário onde foi possível simular o acesso aos mesmos *sites* acessados no período de coleta de dados. Para essa experiência foi utilizado:

- 1) Computador com sistema operacional Ubuntu 12.4 como servidor Proxy Squid;
- 2) Computador com sistema operacional Windows Xp como cliente.

A metodologia de testes utilizada foi análise com base na diferença em Bytes ao se acessar uma página diretamente no servidor ou em um servidor Proxy *web cache*. Para isso foram necessárias três etapas: (i) acesso direto (sem Proxy), para medir o tamanho real dos objetos que compõe a página acessada; (ii) acesso com Proxy *web cache* para os que os objetos sejam armazenados; e (iii) acesso com Proxy *web cache* desta vez esperando que o conteúdo seja apresentado pelo cache.

Em cada uma das etapas o acesso ao *site* foi capturado através do uso da ferramenta *Wireshark*, que como o *tcpdump*, captura pacotes e salva em arquivo com extensão *pcap*. Em seguida pode-se realizar a soma dos *Bytes* dos objetos que compõe a página, dessa forma obteve-se os valores em *Bytes* de cada umas das etapas, podendo assim expor os resultados. É importante salientar que para garantir que os dados retornados fossem realmente do *cache* do servidor Proxy o *cache* do navegador foi desabilitado.

5.1 ANÁLISES DE LOG DE CACHE DO SQUID

Através da análise do arquivo *cache.log* do Squid pode-se observar a ação tomada pelo Proxy em relação a solicitação:

TCP_IMS_HIT: A solicitação estava em *cache*;

TCP_REFRESH_UNMODIFIED: Objeto foi atualizado junto ao servidor de origem;

TCP_MISS: A solicitação não estava em *cache*;

TCP_DENIED: A requisição foi negada. Caso em controle de acesso.

A seguir será apresentado um trecho de log de acesso a uma página www.uol.com.br.

Primeiro acesso com servidor Proxy *web cache*:

```

403187197.387 100 127.0.0.1 TCP_MISS/200 2208 GET
http://imguol.com/admanager/1307/ads/150/46505.gif - DIRECT/200.147.68.8 image/gif
403187197.405 87 127.0.0.1 TCP_MISS/200 3276 GET
http://imguol.com/admanager/1402/ads/154/46512.gif - DIRECT/200.147.68.8 image/gif
14803187197.421 86 127.0.0.1 TCP_MISS/200 2569 GET
http://imguol.com/admanager/1307/ads/155/46509.gif - DIRECT/200.147.68.8 image/gif
1403187197.471 81 127.0.0.1 TCP_MISS/2002554 GET
http://imguol.com/admanager/1402/ads/151/47059.gif - DIRECT/200.147.68.8
image/gif 1403187197.504 96 127.0.0.1 TCP_MISS/200 2374 GET
http://imguol.com/admanager/1406/ads/143/50760.gif - DIRECT/200.147.68.8 image/gif

```

Analisando o log do primeiro acesso, observa-se que em todas as linhas a presença da expressão TCP_MISS, isso significa que todos os objetos que compõe a página foram buscados diretamente do servidor Web.

Segundo Acesso:

```

1402839011.136 0 127.0.0.1 TCP_IMS_HIT/304 407 GET
http://imguol.com/admanager/1307/ads/150/46505.gif - NONE/- image/gif
1402839011.137 0 127.0.0.1 TCP_IMS_HIT/304 407 GET
http://imguol.com/admanager/1402/ads/151/47059.gif - NONE/- image/gif
1402839011.138 0 127.0.0.1 TCP_IMS_HIT/304 407 GET
http://imguol.com/admanager/1406/ads/143/50760.gif - NONE/- image/gif
1402839011.139 0 127.0.0.1 TCP_IMS_HIT/304 407 GET
http://imguol.com/admanager/1406/ads/174/52069.gif - NONE/- image/gif
1402839011.181 45 127.0.0.1 TCP_MISS/204 201 GET http://imp.ads.imguol.com/view? -
DIRECT/200.221.2.30 -

```

No segundo acesso observa-se a presença da expressão TCP_IMS_HIT indicando que a maioria dos objetos solicitados foram atendidos diretamente pelo cache do servidor Proxy.

Um terceiro acesso foi realizado depois de certo tempo para que fosse possível vi-

sualizar a atitude do Squid em relação a objetos ultrapassados, ou seja, desatualizados em comparação com o conteúdo do servidor de origem.

1402837801.410	63	127.0.0.1	TCP_REFRESH_UNMODIFIED/304	405	GET	http://imguol.com/admanager/1307/ads/155/46509.gif - DIRECT/200.147.68.8 image/gif
1402837801.413	188	127.0.0.1	TCP_REFRESH_UNMODIFIED/304	497	GET	http://imguol.com/admanager/1402/ads/154/46512.gif - DIRECT/200.147.68.8 image/gif
1402837801.414	235	127.0.0.1	TCP_REFRESH_UNMODIFIED/304	503	GET	http://imguol.com/admanager/1307/ads/150/46505.gif -DIRECT/200.147.68.8 image/gif
1402837801.427	301	127.0.0.1	TCP_REFRESH_UNMODIFIED/304	494	GET	http://imguol.com/admanager/1403/ads/21/50845.gif - DIRECT/200.147.68.8 image/gif
1402837801.541	428	127.0.0.1	TCP_MISS/200	770	GET	http://bs.serving-sys.com/BurstingPipe/adServer.bs? - DIRECT/63.241.108.124image/gif
1402837801.554	123	127.0.0.1	TCP_MISS/200	2187	GET	http://imguol.com/admanager/1406/ads/152/52944.gif - DIRECT/200.147.68.8 image/gif

Observa-se no terceiro acesso, a predominância de TCP_REFRESH_UNMODIFIED, indicando que o Squid revalidou a maioria dos objetos antes de apresentar o conteúdo ao cliente solicitante. O que garantiu que a visualização da página retornada pelo *cache* fosse a mesma de uma retornada diretamente pelo servidor de origem, porém com um custo bem menor.

5.2 APRESENTAÇÃO DOS RESULTADOS

Um ponto importante a considerar em relação ao uso de *web cache* é que o acesso ao servidor de origem será reduzido, essa redução poderá causar insatisfação de alguns provedores de informação, que utiliza o número de acessos a uma determinada página como uma das bases para investimento em propaganda. Reddy (2011, p. 04) faz a seguinte observação:

Usando um proxy cache irá reduzir os hits no servidor remoto original, que pode decepcionar um monte de provedores de informação, uma vez que eles não podem manter um verdadeiro registro dos acessos a suas páginas. Assim, eles podem decidir não permitir que os seus documentos sejam armazenados em cache.

Com o intuito de impedir que suas páginas sejam armazenadas em *cache*, alguns sites fazem uso de Meta Tags⁴ no-cache, pois a mesma direciona a busca para o servidor original e impede que o conteúdo da página seja mantido em *cache* pelo servidor Proxy. Um exemplo de

⁴Esta é uma tag HTML especial que é usado para armazenar informações sobre uma página da Web, mas não é exibido em um navegador da web

utilização de Meta Tag segundo Fielding et al. (1997, p. 75) seria:

```
<METAHTTP-EQUIV="CACHE-CONTROL"CONTENT="NO-CACHE">
```

Colocada no cabeçalho da página HTML, essa Meta Tag irá forçar qualquer cache intermediário a obter uma nova cópia a partir do servidor de origem.

Para que se utilize o máximo oferecido por um servidor Proxy *web cache*, deve-se procurar meios de manter em cache mesmo os sites que fazem uso de Tag *no-cache*. Para isso o Squid oferece a possibilidade de anular esse recurso (*no-cache*), através de uma diretiva ignore no-cache adicionada no squid.conf. Essa Meta Tag poderá ser ignorada e o conteúdo poderá ser mantido e apresentado pelo servidor Proxy.

Para a realização dos testes a seguinte linha foi adicionada no squid.conf:

```
refresh_pattern -i(gif|png|jpg|jpeg|ico)$ 10080 90% 43200 override-expire ignore-no-cache ignore-no-store ignore-private
refresh_pattern -i(iso|avi|wav|mp3|mp4|mpeg|swf|flv)$ 43200 90% 432000 override-expire ignore-no-cache ignore-no-store ignore-private
refresh_pattern -i (deb|exe|zip|tar|tgz|ram|rar|ppt|doc)$ 10080 90% 43200 override-expire ignore-no-cache ignore-no-store ignore-private
```

As linhas acima adicionadas ao squid.conf faz com que os objetos com as extensões especificadas sejam mantidos em cache, ignorando a Meta tag *no-cache*.

A coleta de dados na prefeitura ficou ativa por um período de 20 (vinte) dias e gerou uma lista com aproximadamente 350 (trezentos e cinquenta) URLs, com um total de 34.810 (trinta e quatro mil oitocentos e dez) acessos. Observou-se que a maioria dos acessos, 74% (setenta e quatro por cento), estavam concentrado basicamente em 30 (trinta) das 350 URLs capturadas. Com essa informação, foi estabelecido um *ranking* composto pelas URLs mais acessadas durante o período de coleta de dados.

Como o objetivo principal dos testes estavam baseados na utilização de *web cache* como forma de economia de banda através da redução do tráfego na rede da prefeitura, não seria interessante manter no *ranking* os sites como TCE (Tribunal de Contas do Estado), bancos, rádios, blogs e vídeos, pois entende-se que o conteúdo dos mesmos, não podem ser mantidos em *cache*, ou devem ser tratados com controle de acesso.

Sendo assim, a simulação dos acessos foram realizadas em 20 (vinte) *sites*, que junto correspondiam a 70,12 % (setenta vírgula doze por cento) do total de acessos. A tabela 1 apresenta o custo em *Bytes* de um acesso a determinada URL diretamente do servidor de origem e também o custo de um acesso utilizando *web cache*, mostrando os *Bytes* que ainda precisaram ser buscados no servidor e os *Bytes* que estavam em *cache*.

Teste realizado com servidor Proxy <i>web cache</i> Squid				
Valores dados em Bytes				
Nº de acessos	URLs acessadas	Sem Proxy	Com Proxy	
			Servidor web	Cache
1172	http://www.boaventura.pr.gov.br	1405855	272152	1133703
305	http://globoesporte.globo.com	1365109	127758	1237351
13665	http://ads.lomadee.com	839799	21415	818384
323	http://veja.abril.com.br	661227	293812	367415
248	http://www.climatempo.com.br	602979	86735	516244
213	http://www.comentei.com.br	402832	55223	347609
220	www.ofuxicogospel.com	362003	149614	212389
835	www.rolonarede.com.br	344121	89653	254468
237	http://www.guiadasemana.com.br	258742	45041	213701
1418	http://www.aliexpress.com	246485	69921	176564
254	http://www.tedioo.net	139712	675	139037
508	http://www.globo.com	119716	57335	62381
1486	http://www.parperfeito.com.br	109680	26111	83569
972	http://www.uol.com.br	103817	45605	58212
527	http://www.bol.uol.com.br	81155	39778	41377
287	http://www.nike.com	63867	4437	59430
568	http://www.netshoes.com.b	42192	1074	41118
611	http://www.ig.com.br	39407	2885	36522
178	http://entretenimento.uol.com.br	28458	961	27497
575	http://noticias.bol.uol.com.br	13071	5937	7134

Tabela 1: Teste realizado com servidor Proxy *web cache* Squid

Os dados expostos na tabela 1 são referentes a um único acesso por *site* e demonstraram em média uma economia de 80,69% (oitenta vírgula sessenta e nove por cento) tendo como ponto máximo o site <http://www.tedioo.net> com 99,52% do seu conteúdo apresentado pelo cache e como ponto mínimo <http://www.bol.uol.com.br> com 50,99% em *cache*.

A figura 7 apresenta a visualização gráfica dos resultados.

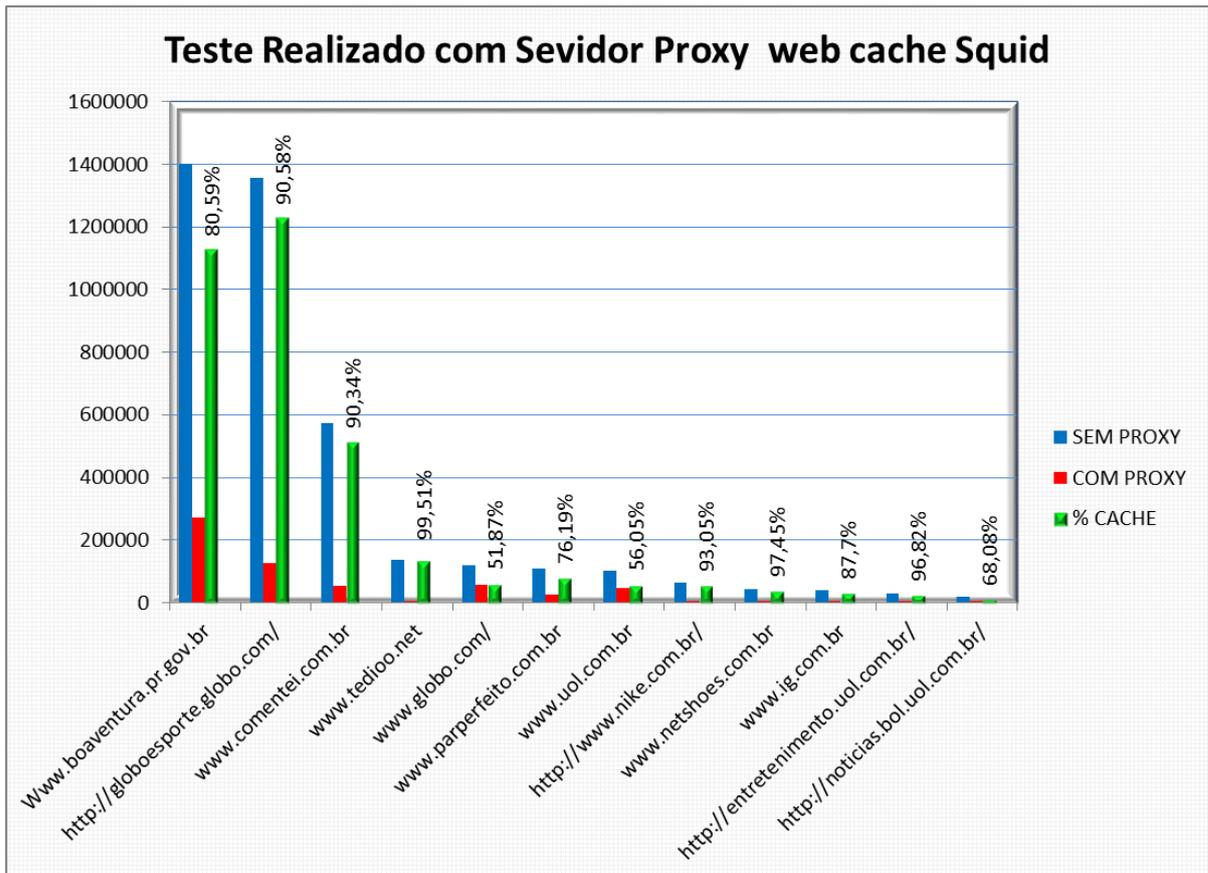


Figura 7: Gráfico de resultados em um acesso a cada site

Observa-se no gráfico dos resultados obtidos uma diferença significativa entre acessar uma determinada página diretamente no servidor de origem e acessar no servidor Proxy. *Sites* que possuem maior quantidade de conteúdo dinâmico tiveram menor porcentagem armazenado em *cache* em relação aos *sites* com maior parte formado por conteúdos estáticos.

Considerando que as 20 (vinte) URL testadas formaram um total de 24.602 (vinte e quatro mil sessentos e dois) acessos e foram responsáveis por um total de 14.2 GB (quatorze vírgula dois *gigabytes*) que trafegaram da Internet para Intranet, a implantação de um *web cache* na prefeitura poderia reduzir o tráfego para aproximadamente 2,8 GB (dois vírgula oito *gigabytes*), como mostra o gráfico da figura 8.

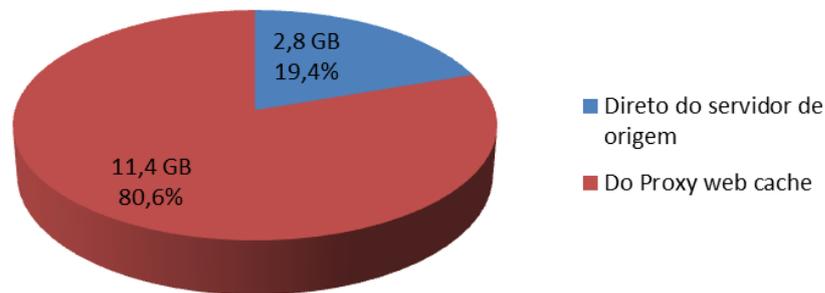


Figura 8: Gráfico representando total de Bytes baixados direto do servidor de origem e Bytes servido pelo cache

O *cache* de página Web contribui para redução do tempo de resposta de uma solicitação enviada em busca de um conteúdo Web. Porém, pode-se reduzir ainda mais o tráfego de dados utilizando o controle de acesso, assim como *web cache*. Essa funcionalidade pode ser aplicada no servidor Proxy Squid, contribuindo para melhorar e aperfeiçoar o uso da banda disponível. Segundo a administração da prefeitura 7 (sete) das 20 (vinte) URL testadas são consideradas irrelevantes e representaram 29,56% (vinte e nove vírgula cinquenta e seis por cento) do total de Bytes que trafegaram na rede com Proxy *web cache* ativo. Sendo assim, a aplicação de regras de controle de acesso foi utilizada para maximizar os resultados. Salienta-se que o sistema de controle de acesso impede o acesso a redes sociais, filmes online, downloads rádios, entre outros e dessa maneira pode auxiliar, mesmo que indiretamente, para segurança amenizando o risco de contaminação por vírus, tendo em vista que em alguns casos, o vírus surge de sites que não fazem parte do ramo de negócio praticado na empresa.

Após a realização dos testes, pode-se levantar uma estimativa da economia alcançada pela prefeitura, considerando as URLs contidas no ranking, a aplicação de um servidor Proxy com *web cache* e controle de acesso diminuiu 86,39% (oitenta e seis vírgula trinta e nove por cento) do tráfego da rede externa para rede interna. No gráfico da figura 9 verifica-se as diferenças entre a realização dos acessos de forma direta (sem Proxy *web cache*), com servidor

Proxy *web cache* e finalmente com Proxy *web cache* e controle de acessos.

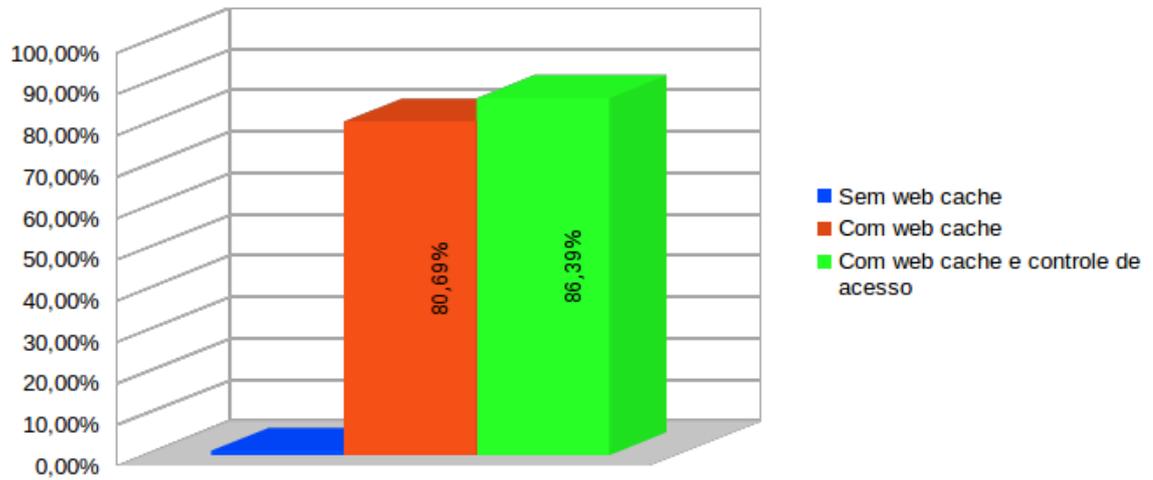


Figura 9: Gráfico de estimativa de economia de banda

Salienta-se que os testes foram simulados em um cenário, e levaram em consideração os *sites* mais acesados na prefeitura durante um período de 20 (vinte) dias, sendo assim os resultados são apresentados com estimativa, podendo variar em um ambiente real.

6 CONCLUSÃO

A concentração de acessos em um número pequeno de *sites* sugere um padrão na utilização da Internet na prefeitura. Também notou-se que a grande maioria dos *sites* não tem nenhuma relação direta com o trabalho desenvolvido pelos colaboradores. Isso mostra que a forma como a Internet é utilizada no ambiente de trabalho deve ser controlada.

Os testes realizados demonstraram uma estimativa positiva em relação ao uso de um servidor Proxy com *web cache* na prefeitura de Boa Ventura de São Roque. O Squid se mostrou eficiente e dinâmico, tanto na realização de cache quanto no controle de acesso, oferecendo diversas possibilidades de configuração, permitindo ao administrador da rede tratar e corrigir os problemas relatados pela administração. Contudo, as configurações utilizadas nos testes foram relativamente simples, pois almejavam apenas demonstrar a possibilidade de manter em cache local parte dos objetos acessados pelos usuários da rede. Uma utilização em uma escala maior, evidencia a necessidade de estudos mais aprofundados, como a identificação de arquiteturas e algoritmos de substituição, que melhor atendam as necessidades particulares em cada empresa. Além disso, a utilização de *web cache* exige do administrador da rede um acompanhamento constante, visando manter e melhorar o serviço através de análise de logs. Uma boa configuração certamente resultará em eficiência e eficácia. Porém, uma configuração ruim, aliada a uma falta de acompanhamento, poderá ter um efeito contrário levando muitas vezes a descontinuidade no uso de *web cache*.

REFERÊNCIAS

- DANALIS, A.; MARCKATOS, E. Web caching: A survey. **Institute of Computer Science–FORTH**, Greece, 2002. Disponível em: <<http://www.irma-international.org/viewtitle/18424/>>. Acesso em: 07 nov. 2013.
- FIELDING, R. et al. Rfc 2068, hypertext transfer protocol—http/1.1. **Status: PROPOSED STANDARD**, RFC 2068, January, 1997.
- GADDE, S.; RABINOVICH, M.; CHASE, J. Reduce, reuse, recycle: An approach to building large internet caches. In: **IEEE. Operating Systems, 1997., The Sixth Workshop on Hot Topics in**. 1997. p. 93–98. Disponível em: <http://pdf.aminer.org/000/252/591/reduce_reuse_recycle_an_approach_to_building_large_internet_caches.pdf>. Acesso em: 06 julho. 2014.
- LUNARD, R. **SQUID Solução Definitiva**. [S.l.]: Rio de Janeiro, 2005.
- MUKESH, D.; CHARANJIT, S. Study on web caching architecture: A survey. **International Journal of Advanced Research in Computer Science and Software Engineering**, India, v. 3, n. 11, p. 580–584, 2013. Disponível em: <http://www.ijarcsse.com/docs/papers/Volume_3/11_November2013/V3I11-0385.pdf>. Acesso em: 06 julho. 2014.
- RAMANA, S. S.; SRINATH, H. **Web Caching- A Technique to Speedup Access to Web Contents**. 2002. Disponível em: <<http://www.ias.ac.in/resonance/Volumes/07/07/0054-0062.pdf>>. Acesso em: 07 nov. 2013.
- REDDY, S. R. **Seminar Report on Web Caching**. School of Information Technology Indian Institute of Technology- Kharagpur, 2011. Disponível em: <<http://sit.iitkgp.ernet.in/research/aut04seminar1/11r.pdf>>. Acesso em: 06 julho. 2014.
- RICCI, B.; MENDONÇA, N. **SQUID Solução Definitiva**. [S.l.]: Rio de Janeiro, 2006.
- RYU, S.; CHRISTOPHER, R.; CHUNMING, Q. Advances in internet congestion control. **IEEE Communications Surveys**, v. 5, n. 1, p. 28–39, 2003. Disponível em: <<http://www.cse.bgu.ac.il/files/Courses/233/HelpMaterial/935/Advances in Internet Congestion Control.pdf>>. Acesso em: 06 julho. 2014.
- SAND, L. C.; KLEBER, V. C. **GT-ATER: Aceleração do Transporte de Dados com o Emprego de Redes de Circuitos Dinâmicos**. 2013. Disponível em: <http://labora.inf.ufg.br/gt-ater/resultados/GT-ATER_fase1_RT1.pdf>. Acesso em: 13 julho. 2014.
- SQUID-CACHE: Optimising web delivery. 2014. Disponível em: <<http://www.squid-cache.org/>>. Acesso em: 20 jun. 2014.
- SULAIMAN, S. et al. Caching and prefetching: What, why, and how? **IEEE**, p. 01–08, 2008. Disponível em: <<http://www.irma-international.org/viewtitle/18424/>>. Acesso em: 07 nov. 2013.

TCPDUMP: Site. 2013. Disponível em: <<http://www.tcpdump.org>>. Acesso em: 20 jun. 2014.

TELSTRA, G. H. Web caching. **Internet Protocol**, v. 2, n. 3, 1999. Disponível em: <http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_16-2/ipj_16-2.pdf.pdf>. Acesso em: 07 nov. 2013.

WALEED, A.; SHAMSUDDIN, S. M.; ISMAIL, A. S. A survey of web caching and prefetching. **Int. J. Advance. Soft Comput. Appl**, v. 3, n. 1, p. 18–44, 2011. Disponível em: <http://gmm.fsksm.utm.my/mariyam/PUBLISHED_PAPERS_DRCTM/YEAR_2011/IJASCA_Waleed.pdf>. Acesso em: 06 julho. 2014.

YEU, Y. C.; FEDEL, G. d. S. Aceleração no acesso à internet: estudo sobre o servidor proxy/cache squid. **Revista Tecnológica da Fatec Americana**, v. 2, n. 1, p. 12–34, 2014. Disponível em: <<http://www.fatec.edu.br/revista/wp-content/uploads/2013/06/Aceleraçao-no-acesso-a-internet-estudo-sobre-o-servidor-proxy-cache-Squid1.pdf>>. Acesso em: 06 julho. 2014.

ZENG, D.; WANG, F.-Y.; LIU, M. Efficient web content delivery using proxy caching techniques. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on**, IEEE, v. 34, n. 3, p. 270–280, 2004. Disponível em: <http://opim.wharton.upenn.edu/kartikh/reading/zeng_ieee.pdf>. Acesso em: 06 julho. 2014.

ANEXO A – TESTES REALIZADO NOS SITES ACESSADOS

Script em linguagem perl que retorna a soma dos Bytes da página acessada.

```
#!/usr/bin/perl
```

```
$bytes = 0;
open (ARQ, $ARGV[0]) ;
while $l = < ARQ > {
    $l = m/length ([0-9]+)/;
    $bytes += $l;
}
close(ARQ);
print "Bytes TCP: ".$bytes.;
```

Resultados gerado pelo escript passando como argumento os arquivos gerado em cada uma das etapas dos testes:

- **http://ads.lomadee.com/**

```
root@hp-g42:/etc/squid3/testes# perl conta.perl lamadee_direto.txt
```

```
Bytes TCP: 834788
```

```
root@hp-g42:/etc/squid3/testes# perl conta.perl lomadee_acesso1.txt
```

```
Bytes TCP: 839799
```

```
root@hp-g42:/etc/squid3/testes# perl conta.perl lomadee_acesso2.txt
```

```
Bytes TCP: 21415]
```

- **http://www.aliexpress.com** root@hp-g42:/etc/squid3/testes# perl conta.perl aliexpress_direto.txt

```
Bytes TCP: 246485
```

```
root@hp-g42:/etc/squid3/testes# perl conta.perl aliexpress_acesso1.txt
```

Bytes TCP: 246456

root@hp-g42:/etc/squid3/testes# perl conta.perl aliexpress_acesso2.txt

Bytes TCP: 69921

● **Www.boaventura.pr.gov.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl boaventura_direto.txt

Bytes TCP: 1405855

root@hp-g42:/etc/squid3/testes# perl conta.perl boaventura_acesso1.txt

Bytes TCP: 1401911

root@hp-g42:/etc/squid3/testes# perl conta.perl boaventura_acesso2.txt

Bytes TCP: 272152

● **www.ig.com.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl ig_direto.txt

Bytes TCP: 39407

root@hp-g42:/etc/squid3/testes# perl conta.perl ig_acesso1.txt

Bytes TCP: 39677

root@hp-g42:/etc/squid3/testes# perl conta.perl ig_acesso2.txt

Bytes TCP: 2885

● **www.netshoes.com.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl netshoes_direto.txt

Bytes TCP: 42192

root@hp-g42:/etc/squid3/testes# perl conta.perl netshoes_acesso1.txt

Bytes TCP: 41571

root@hp-g42:/etc/squid3/testes# perl conta.perl netshoes_acesso2.txt

Bytes TCP: 1074

● **www.uol.com.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl uol_direto.txt

Bytes TCP: 103817

root@hp-g42:/etc/squid3/testes# perl conta.perl uol_acesso1.txt

Bytes TCP: 103770

root@hp-g42:/etc/squid3/testes# perl conta.perl uol_acesso2.txt

Bytes TCP: 45605

●**www.rolonarede.com.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl rolonarede_direto.txt

Bytes TCP: 344121

root@hp-g42:/etc/squid3/testes# perl conta.perl rolonarede_acesso1.txt

Bytes TCP: 350596

root@hp-g42:/etc/squid3/testes# perl conta.perl rolonarede_acesso2.txt

Bytes TCP: 89653

●**www.parperfeito.com.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl parperfeito2_direto.txt

Bytes TCP: 109680

root@hp-g42:/etc/squid3/testes# perl conta.perl parperfeito2_acesso1.txt

Bytes TCP: 101285

root@hp-g42:/etc/squid3/testes# perl conta.perl parperfeito2_acesso2.txt

Bytes TCP: 26111

●**www.comentei.com.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl comentei_direto.txt

Bytes TCP: 402832

root@hp-g42:/etc/squid3/testes# perl conta.perl comentei_acesso1.txt

Bytes TCP: 571891

root@hp-g42:/etc/squid3/testes# perl conta.perl comentei_acesso2.txt

Bytes TCP: 55223

●**http://noticias.bol.uol.com.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl bol_direto.txt

Bytes TCP: 13071

root@hp-g42:/etc/squid3/testes# perl conta.perl bol_acesso1.txt

Bytes TCP: 18062

root@hp-g42:/etc/squid3/testes# perl conta.perl bol_acesso2.txt

Bytes TCP: 5937

●**www.globo.com/**

root@hp-g42:/etc/squid3/testes# perl conta.perl globo_direto.txt

Bytes TCP: 119716

root@hp-g42:/etc/squid3/testes# perl conta.perl globo_aceso1.txt

Bytes TCP: 119132

root@hp-g42:/etc/squid3/testes# perl conta.perl globo_aceso2.txt

Bytes TCP: 57335

●**http://globoesporte.globo.com/**

root@hp-g42:/etc/squid3/testes# perl conta.perl globoesporte_direto.txt

Bytes TCP: 1365109

root@hp-g42:/etc/squid3/testes# perl conta.perl globoesporte_aceso1.txt

Bytes TCP: 1356222

root@hp-g42:/etc/squid3/testes# perl conta.perl globoesporte_aceso2.txt

Bytes TCP: 127758

●**http://www.nike.com.br/**

root@hp-g42:/etc/squid3/testes# perl conta.perl nike_direto.txt

Bytes TCP: 63867

root@hp-g42:/etc/squid3/testes# perl conta.perl nike_aceso1.txt

Bytes TCP: 62753

root@hp-g42:/etc/squid3/testes# perl conta.perl nike_aceso2.txt

Bytes TCP: 4437

●**http://veja.abril.com.br/**

root@hp-g42:/etc/squid3/testes# perl conta.perl veja_direto.txt

Bytes TCP: 661227

root@hp-g42:/etc/squid3/testes# perl conta.perl veja_aceso1.txt

Bytes TCP: 664690

root@hp-g42:/etc/squid3/testes# perl conta.perl veja_aceso2.txt

Bytes TCP: 293812

- **http://www.guiadasemana.com.br/**

root@hp-g42:/etc/squid3/testes# perl conta.perl guiasemana_direto.txt

Bytes TCP: 258742

root@hp-g42:/etc/squid3/testes# perl conta.perl guiasemana_acesso1.txt

Bytes TCP: 269436

root@hp-g42:/etc/squid3/testes# perl conta.perl guiasemana_acesso2.txt

Bytes TCP: 45041

- **http://entretenimento.uol.com.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl entretenimento.uol_direto.txt

Bytes TCP: 28458

root@hp-g42:/etc/squid3/testes# perl conta.perl entretenimento.uol_acesso1.txt

Bytes TCP: 28545

root@hp-g42:/etc/squid3/testes# perl conta.perl entretenimento.uol_acesso2.txt

Bytes TCP: 961

- **http://www.bol.uol.com.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl bol.uol_direto.txt

Bytes TCP: 81155

root@hp-g42:/etc/squid3/testes# perl conta.perl bol.uol_acesso1.txt

Bytes TCP: 81039

root@hp-g42:/etc/squid3/testes# perl conta.perl bol.uol_acesso2.txt

Bytes TCP: 39778

- **www.ofuxicogospel.com/**

root@hp-g42:/etc/squid3/testes# perl conta.perl ofuxicogospel_direto.txt

Bytes TCP: 362003

root@hp-g42:/etc/squid3/testes# perl conta.perl ofuxicogospel_acesso1.txt

Bytes TCP: 151314

root@hp-g42:/etc/squid3/testes# perl conta.perl ofuxicogospel_acesso2.txt

Bytes TCP: 149614

- **http://www.tedioo.net**

root@hp-g42:/etc/squid3/testes# perl conta.perl tedionet_direto.txt

Bytes TCP: 139712

root@hp-g42:/etc/squid3/testes# perl conta.perl tedionet_acesso1.txt

Bytes TCP: 155108

root@hp-g42:/etc/squid3/testes# perl conta.perl tedionet_acesso2.txt

Bytes TCP: 675

● **www.climatempo.com.br**

root@hp-g42:/etc/squid3/testes# perl conta.perl climatempo_direto.txt

Bytes TCP: 602979

root@hp-g42:/etc/squid3/testes# perl conta.perl climatempo_acesso1.txt

Bytes TCP: 614532

root@hp-g42:/etc/squid3/testes# perl conta.perl climatempo_acesso2.txt

Bytes TCP: 86735

ANEXO B – AVALIAÇÃO DE RELEVÂNCIA NOS ACESSOS

O Secretário de Administração em exercício na prefeitura, realizou uma avaliação a cerca dos acessos efetuados pelos colaboradores nos *Sites* testados e classificou-os de acordo com a relevância para o serviço prestado nesse setor.

URLs acessadas	Relevante	Irrelevante
http://www.boaventura.pr.gov.br	X	
http://globoesporte.globo.com	X	
http://ads.lomadee.com		X
http://veja.abril.com.br	X	
http://www.climatempo.com.br	X	
http://www.comentei.com.br		X
www.ofuxicogospel.com		X
www.rolonarede.com.br		X
http://www.guiadasemana.com.br	X	
http://www.aliexpress.com		X
http://www.tedioo.net		X
http://www.globo.com	X	
http://www.parperfeito.com.br		X
http://www.uol.com.br	X	
http://www.bol.uol.com.br	X	
http://www.nike.com	X	
http://www.netshoes.com.b	X	
http://www.ig.com.br	X	
http://entretenimento.uol.com.br	X	
http://noticias.bol.uol.com.br	X	

Tabela 2: Avaliação de relevância nos acessos.