Centro Estadual de Educação Tecnológica Paula Souza

Faculdade de Tecnologia de Americana

Curso Superior de Tecnologia em Segurança da Informação

Rodrigo Justino da Costa

Uma Solução Para a Transmissão de Conteúdo na Internet

AMERICANA
2011

Rodrigo Justino da Costa

Uma Solução Para a Transmissão de Conteúdo na Internet

Monografia apresentada à Faculdade de Tecnologia de Americana como parte dos requisitos para a obtenção do título de Tecnólogo em Segurança da Informação.

Orientador Prof. Marcus Vinícius Lahr Giraldi.

AMERICANA
2011

DEDICATÓRIA

Dedico a Deus por não ter permitido que eu morresse no parto, dando assim a possibilidade de eu concluir todos os seus propósitos. Dedico aos meus pais por me darem a melhor educação que já vi e a minha namorada por fazer a minha vida tão especial.

AGRADECIMENTOS

Agradeço a Deus por toda energia me dada para superar os diversos obstáculos que apareceram na minha vida. Ao professor Marcus por aceitar orientar-me e por toda a sua dedicação. Aos meus pais por toda a sabedoria que me ensinaram. Agradeço a minha namorada Thayla Camilo por sempre acreditar no meu potencial, por dar-me força para superar os tropeços da vida, por alimentar minha alma com amor e carinho, fazendo com que eu multiplique isto e divida com todos. Aos meus amigos do serviço que, apesar das pancadas que levei na costela e em outras partes do corpo, sempre estiveram ao meu lado. Ao Robson, Angelica e a Monica por sempre oferecerem um ombro amigo na faculdade. Ao Edson Gaseta pela sua ajuda e dedicação. Aos demais que merecem estar aqui, mas são muitos para serem citados, fica o meu obrigado.

EPÍGRAFE

“Quanto mais silencioso você for, mais será capaz de ouvir...”

Comunidade BackTrack

RESUMO

Este estudo aborda uma solução hoje empregada na internet para a transmissão de conteúdo, a Rede de Distribuição de Conteúdo (*Content Delivery Network*, CDN).

Com o avanço da internet se fez necessário o uso de novas metodologias e tecnologias para melhorar a transmissão de páginas de portais de notícias, sites de vídeos estático e dinâmico, documentos e qualquer outro objeto que tenha um grande volume de acesso.

O estudo mostra como o CDN funciona, a melhoria que ele agregou na grande rede de computadores e alguns trabalhos acadêmicos voltados na melhoria desta solução, que tem um papel fundamental e invisível para os dias atuais.

PALAVRAS CHAVE: Infraestrutura. Rede sobreposta. Conteúdo.

ABSTRACT

This study addresses a solution used today by the Internet for the transmission of content, Content Distribution Network (CDN).

With the advancement of Internet has made necessary the use of new methodologies and technologies to improve the transmission of pages of news portals, transmission of movies static and dynamic, documents and any other object that has a large volume of access.

The study shows how the CND works, your improvement in the vast network of computers and some academics works focused on improving this solution that has a key role and invisible to the present day.

KEY WORDS: Infrastructure. Overlay network. Content.

SUMÁRIO

LISTA DE FIGURAS

1) INTRODUÇÃO

Como a *Internet* se tornou um meio dominante para distribuição de informações devido a grande quantidade de meios de acesso, conforme figura 1, se fez necessário o uso de mecanismos para distribuir o tráfego de forma eficiente e segura. Entretanto, os métodos atuais de transmissão de dados estão sujeitos a falhas e lentidões imprevisíveis. Duas das principais causas destas falhas e demoras são as redes de comunicação de dados congestionadas e os servidores sobrecarregados, geralmente em função do excesso de requisições simultâneas que são geradas de diversas partes do mundo, conforme figura 2, acarretando recusar a conexão ou realizá-la lentamente.

Apesar do investimento na melhoria da infraestrutura de comunicação de dados e de servidores eles podem inesperadamente serem sobrecarregados, tanto pelo crescimento contínuo dos usuários da *internet* como, por exemplo, devido a promoções pontuais ou novos conteúdos disponibilizados nos *web sites*.

Isto gera um grande problema: “como prover acesso a toda esta informação de maneira rápida e com qualidade, sem concentrar todo o material em poucos pontos de acesso e levando em consideração a distribuição geográfica de acesso?”.

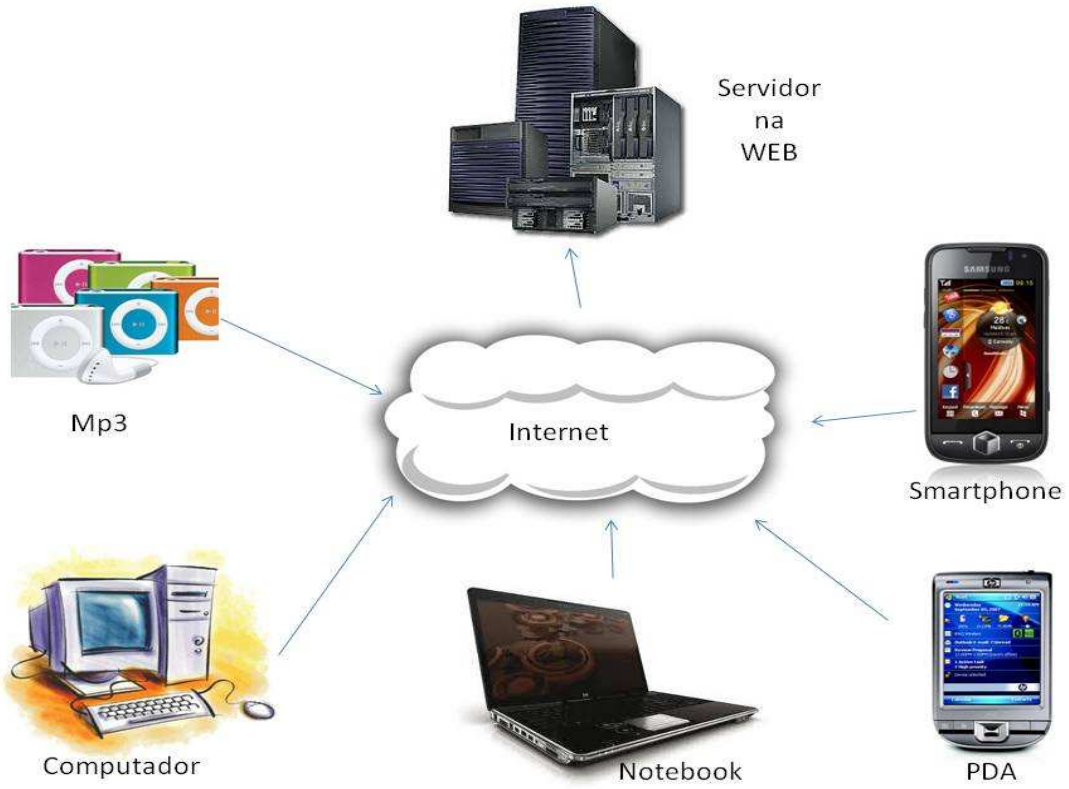

Figura 1: Estrutura Centralizada De Acesso A Conteúdo

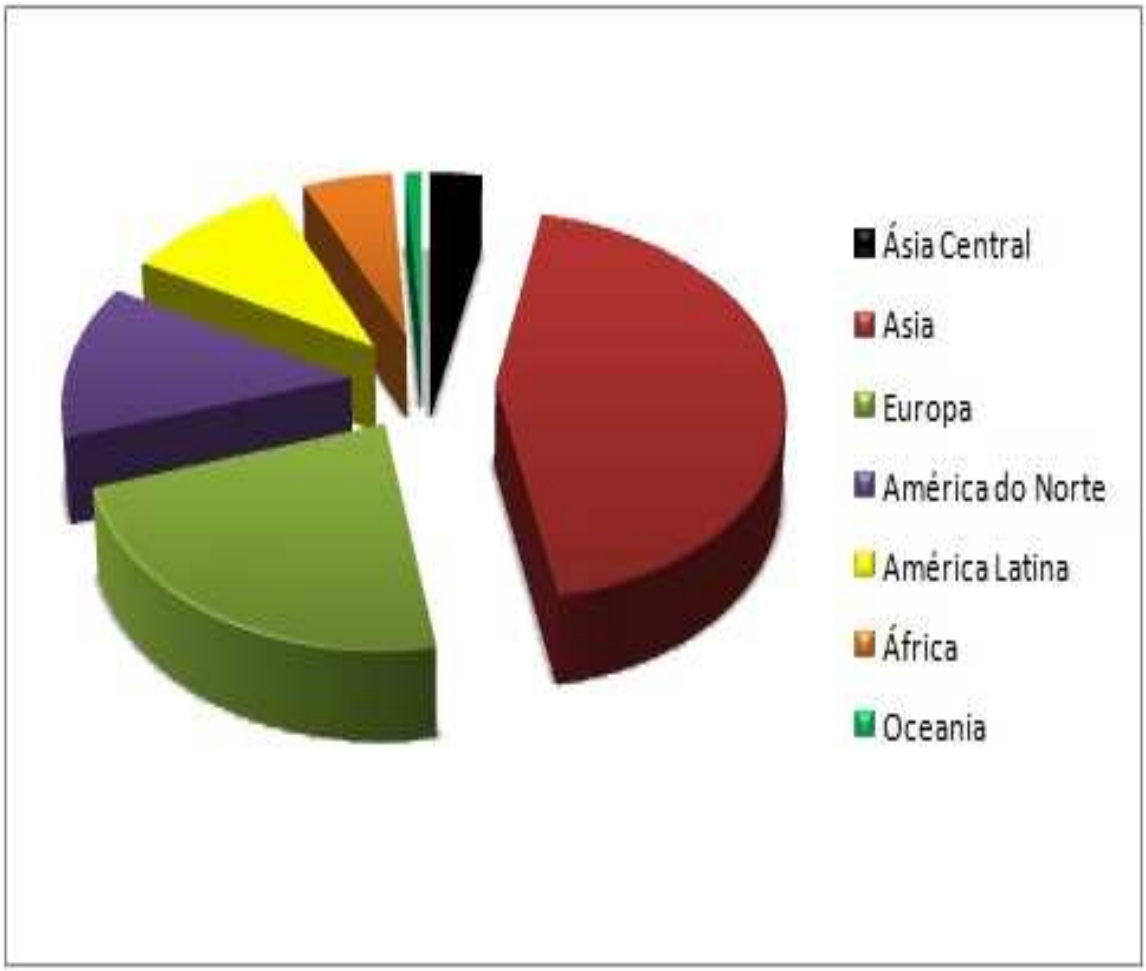

Figura 2: Distribuição Dos Acessos A *Internet* No Mundo
Fonte: HTTP://www.internetworldstats.com/stats.htm (2011)

2) O CONCEITO DE SERVIDOR DE CACHE

Quanto mais perto melhor. Esta é a ideia por trás de qualquer mecanismo de *cache*. Processador, disco rígido, placa de rede, placas controladoras SCSI, a maioria dos *hardwares* e vários *softwares* utilizam áreas de armazenamento de acesso rápido como uma maneira de prover uma alta taxa de transferência de informações entre eles. Os computadores fazem um bom uso do sistema de *cache* para armazenar a informação perto de onde será usado.

A maioria dos navegadores da *internet* possui um mecanismo de *cache* próprio. A intenção é economizar banda de acesso à *internet* e tempo ao carregar informações que já foram acessadas anteriormente e que foram armazenadas no disco local, ao invés de realizar novamente a cópia do objeto de seu provedor. Um

simples exemplo é quando o usuário clica no botão "voltar" de seu navegador para retornar a uma página acessada anteriormente. Neste caso toda a informação que não foi alterada será acessada diretamente do *cache* local provido pelo navegador, ao invés de solicitar todos os objetos ao site já visitado.

Outra situação onde é empregado o uso do *cache* do navegador é quando um usuário após um período faz um novo acesso a um *site* visitado anteriormente. Figuras como o logo do *site* tem uma taxa de alteração pequena, permitindo assim que ele possa ser recuperado localmente ao invés de ser copiado novamente.

2.1) Web Cache

O objetivo de um *web cache* é aumentar a velocidade de resposta ao usuário, reduzir a banda de acesso consumida e reduzir a carga de trabalho do servidor de origem. Quando é solicitado um objeto e este está armazenado no servidor de *cache* o seu acesso será mais rápido para o usuário do que se fosse acessado diretamente do servidor de origem. Porém, no caso de falha na resposta ao cliente por não possuir o objeto requisitado, este passo extra de passar a requisição pelo servidor de *cache* antes do servidor principal aumenta o tempo de resposta ao solicitante, porque ele gera uma transmissão a mais de dados e com isto aumenta o tráfego de informações na rede aumentando o tempo de resposta ao usuário ao invés de cumprir o seu propósito de diminuí-lo. A Figura 3 (exposta na próxima página) exemplifica o funcionamento básico de um *Web Cache*.

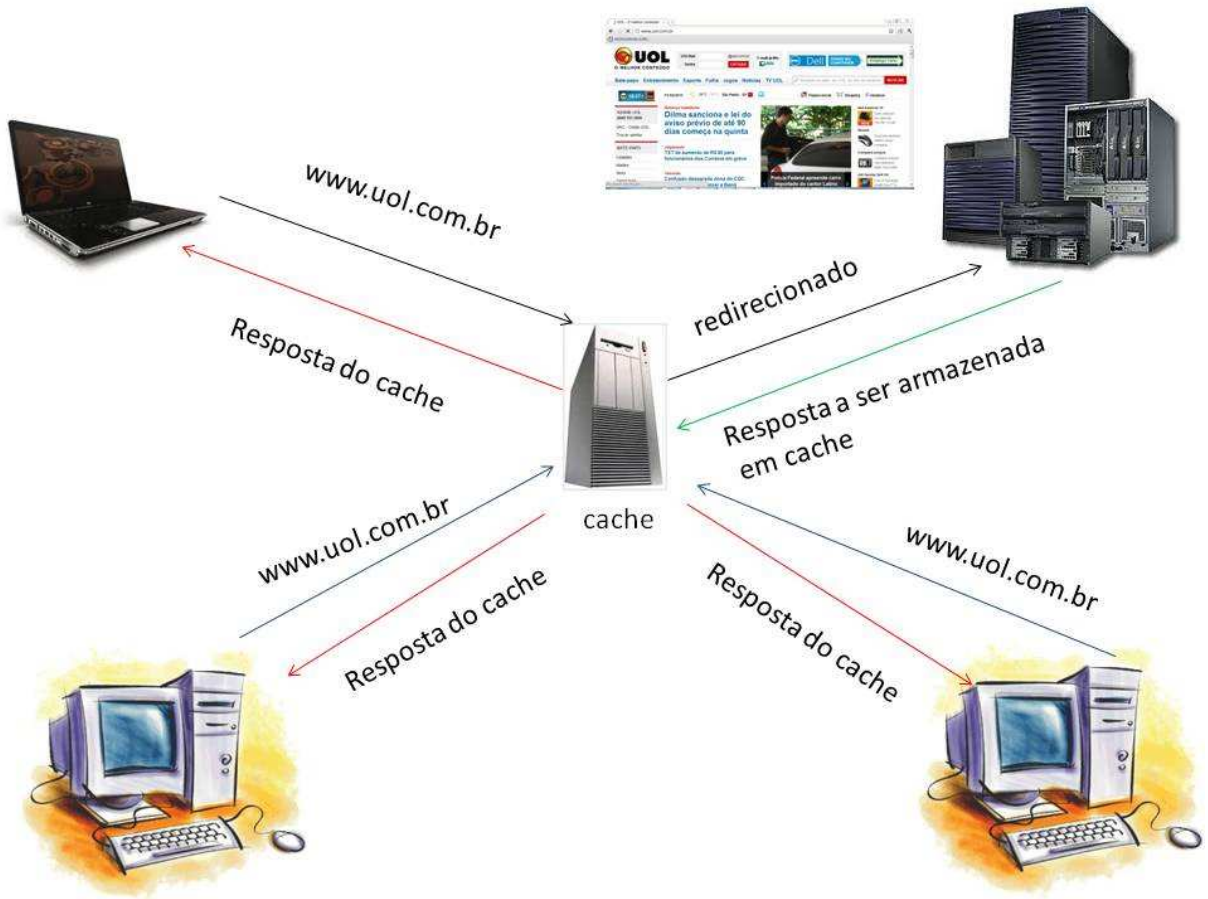

Figura 3: Exemplo De Funcionamento Do *Web Cache*

2.2) Operação Básica De Um Web Cache

Todo servidor de *cache* deverá decidir o que fazer em cada requisição que passa por ele. O servidor recebe a requisição e analisa sua lista de objetos armazenados para determinar se ele possui ou não o objeto. Ele também deve analisar se, caso tenha o objeto, ele está atualizado ou não. Para tal é questionado o servidor de origem até quando o objeto solicitado é considerado atualizado e, caso esteja desatualizado, é copiado o arquivo e retransmitido ao usuário. Finalmente o servidor determina se irá armazenar ou não o arquivo atualizado no seu disco, baseado na quantidade de espaço disponível e na estimativa do valor futuro deste objeto. A Figura 4 (exposta na próxima página) mostra o fluxo de informação na operação básica de um *Web Cache*.

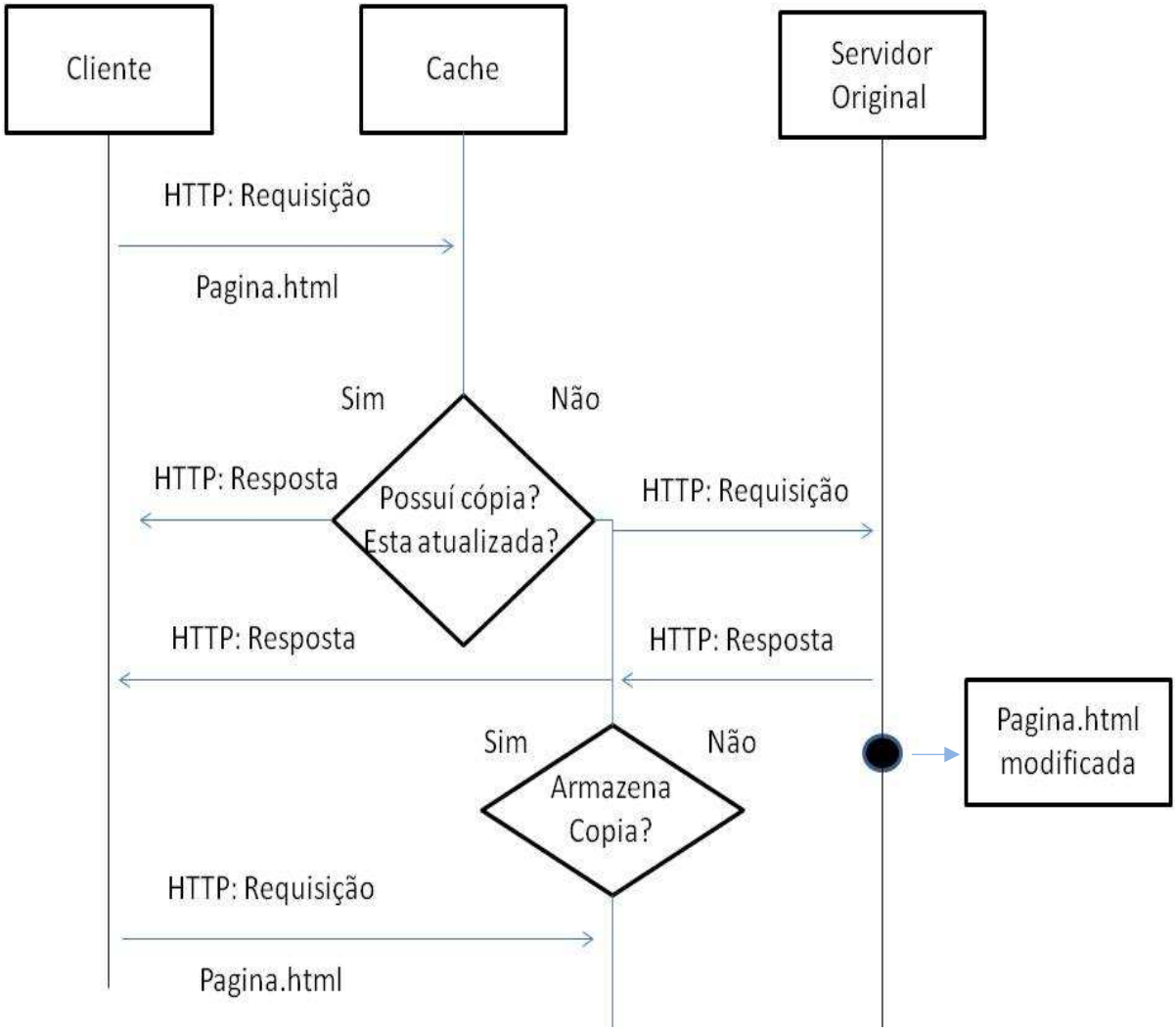

Figura 4: Diagrama de Sequência Do Funcionamento Básico Do *Web Cache*

3) CDN

CDN (*Content Distribution Network*, Rede de Distribuição de Conteúdo) é composto por um conjunto de servidores estrategicamente distribuídos geograficamente, próximos aos *backbones* destas regiões, com o intuito de armazenarem cópias de conteúdo *multimídia* ou retransmitir o conteúdo no caso de vídeo em tempo real e distribuí-los rapidamente ao usuário final. A Figura 5 mostra o funcionamento básico de um CDN.

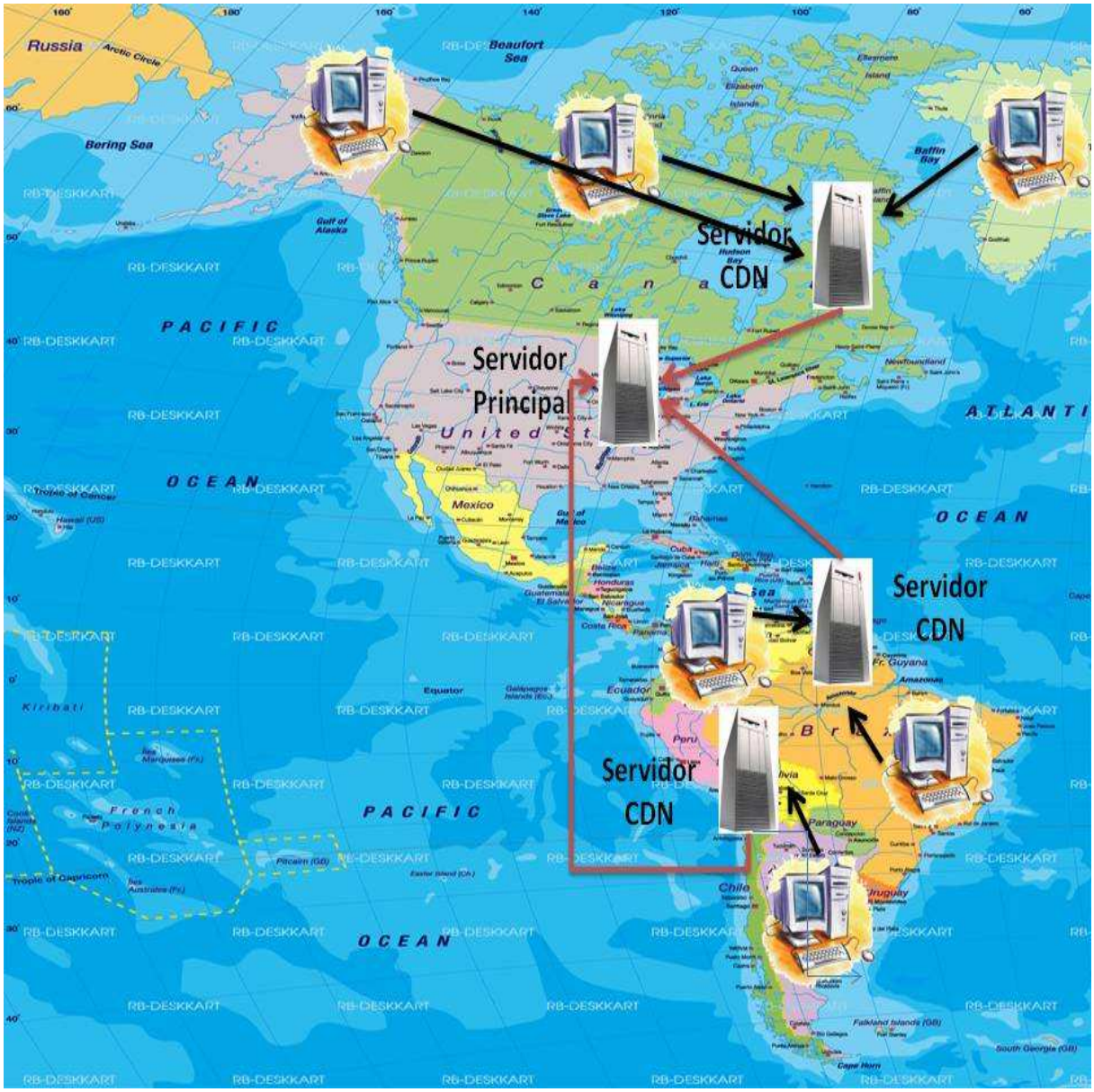

Figura 5: Exemplo De Servidores CDN Na América

Basicamente, um CDN é uma rede sobreposta construída por um grupo de servidores de CDN distribuídos estrategicamente no planeta. No momento que um usuário requisita uma informação, como uma música ou um clipe, a solicitação é redirecionada para o CDN mais próximo de sua localização através de um mecanismo de redirecionamento de requisição para que o conteúdo solicitado seja primeiramente transmitido por ele ao invés do servidor central de transmissão, também conhecido como servidor *root*, provendo assim uma alta taxa de velocidade na transmissão dos dados.

A função do CDN é distribuir o conteúdo estático, como arquivos de texto, documentos e principalmente conteúdo multimídia do provedor original, armazená-lo

em servidores de *cache* (denominados servidores secundários) e através do uso de algoritmo de roteamento encaminhar a solicitação do cliente para este conteúdo até um servidor de CDN mais próximo geograficamente dele. O conteúdo replicado entre os *caches* são previamente selecionados ou por estatística de uso e com isto é realizado a replicação do conteúdo mais acessado ou pela escolha do administrador do conteúdo. A intenção é que a taxa de resposta entre a requisição de um conteúdo e a entrega deste pelos *caches* seja próxima de 100%, reduzindo assim o tempo de resposta entre a comunicação do cliente com o servidor e o consumo de banda de rede nesta comunicação. Um dos benefícios trazidos por esta rede é o fato que um servidor de CDN pode prover armazenamento e entrega de conteúdo de vários sites diferentes. Esta abordagem permite a criação da malha de distribuição com maior eficiência e maior redução dos custos de implantação, visto que este será dividido entre as partes interessadas.

A arquitetura do sistema de CDN é composta por sete componentes, conforme mostrado na Figura 6, dispostos em oito passos, conforme descrito abaixo:

1. O **servidor original** delega a URL de um objeto para ser replicado e entregue pelo CDN para o sistema de roteamento de requisição;
2. O servidor original pública o objeto que será replicado e entregue pelo CDN no sistema de distribuição;
3. O **sistema de distribuição** move o conteúdo para os servidores de *cache*;
4. O **cliente** solicita o conteúdo do objeto para o servidor original. Porém, devido à delegação da URL a solicitação é encaminhada para o sistema de roteamento de requisição;
5. O sistema de **roteamento de requisição** encaminha a solicitação para o servidor de *cache* mais adequado na malha de CDNs;
6. O **servidor de *cache*** selecionado entrega o objeto solicitado para o cliente. Em paralelo é enviado uma informação de acesso para o sistema de contabilização;
7. O **sistema de contabilização** armazena a informação de acesso para transformá-la em estatística de acesso para fins de replicação e relatório de faturamento no caso do uso de CDNs de terceiros;
8. O **relatório de faturamento** é baseado nas informações contabilizadas de acesso de cada objeto de todas as URLs delegadas. Este relatório é utilizado

para emitir o faturamento às empresas que contrataram o serviço de CDN de terceiros.

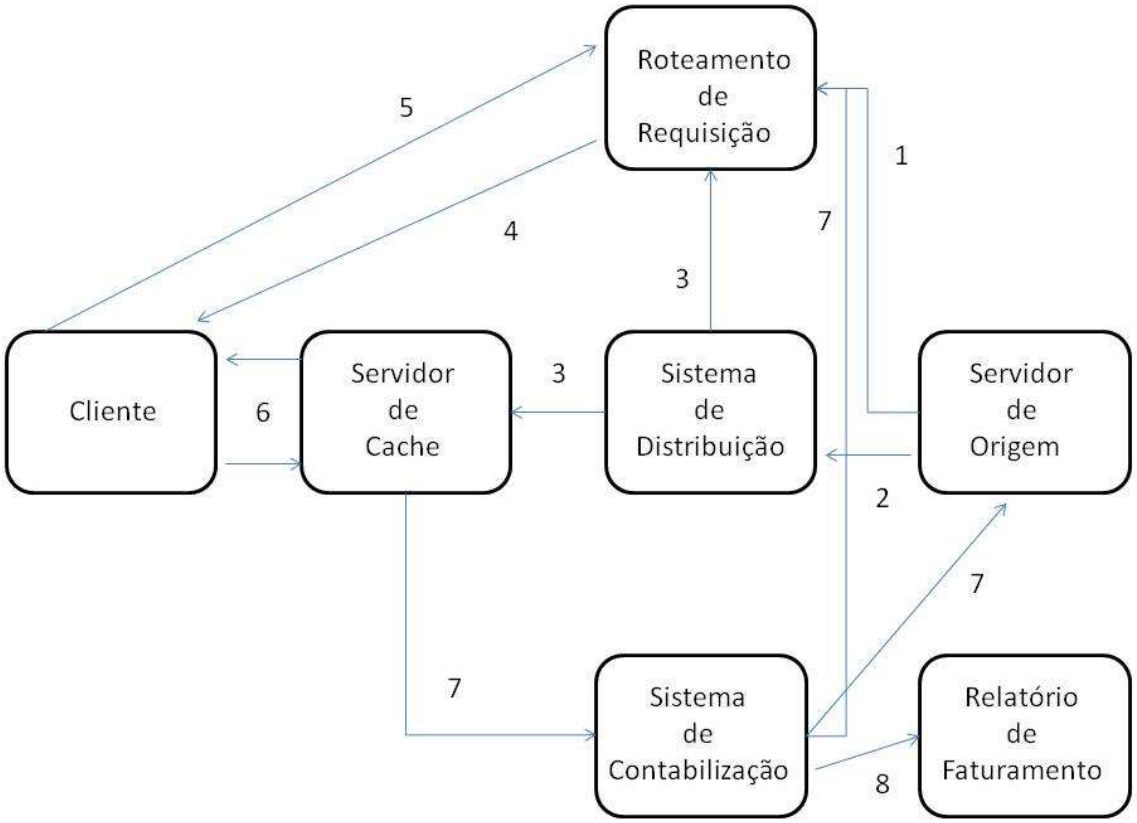

Figura 6: Componentes Da Arquitetura Do CDN

3.1) Arquitetura De Distribuição

O sistema de distribuição mais utilizado é composto por uma rede sobreposta na *internet* de comunicação entre os servidores de CDN. Desta forma é possível utilizar a *internet* para manter uma árvore de comunicação e replicação de dados entre os servidores, retransmitindo o conteúdo do servidor original para os servidores de *cache* designados. Ela envolve a réplica do conteúdo, a localização do objeto e o roteamento da requisição.

A localização da réplica do conteúdo diz respeito a quantas réplicas terá cada objeto e em quais *caches* eles serão armazenados na rede. Esta questão é dividida em duas partes: localização do servidor de *cache* e localização do objeto replicado. Localização do servidor de *cache* diz respeito sobre como determinar os lugares na internet que serão instalados os servidores de *cache*. Já a localização do objeto replicado diz respeito sobre como determinar qual conteúdo será replicado entre os servidores de *cache*.

O sistema de roteamento de requisição é usado para selecionar o servidor de *cache* mais adequado que possua a cópia do objeto requisitado pelo cliente e posteriormente provido a ele. A determinação do servidor se dá primeiramente pela escolha dos servidores mais próximos do cliente e posteriormente pela escolha do servidor com menor carga de trabalho. A distância entre o cliente e o servidor geralmente é medida através de quantos roteadores (*hop counts*) o pacote de dados trafega e pelo tempo gasto entre a requisição e recebimento da resposta (*round-trip*) entre o cliente e o servidor.

A carga de trabalho é medida através do “empurro” da informação pelo servidor (*server push*) onde os servidores de *cache* propagam as informações sobre o consumo de seus recursos (memória, banda de dados, cpu e disco) para agentes de comunicação ou pela sondagem de clientes, onde os agentes de comunicação são os responsáveis por solicitar aos servidores de *cache* os dados sobre o uso de seus recursos computacionais. Apesar de existir alguns outros métodos estas são as duas principais formas empregadas.

O roteamento da requisição emprega uma variedade de técnicas para determinar como será encaminhada a requisição do cliente para o servidor de *cache*. Estas metodologias incluem multiplexação do cliente, redirecionamento de HTTP, dns indireto, *anycasting* e roteamento entre pares.

3.2) Roteamento De Requisição

O sistema de roteamento de requisição é composto, principalmente, por cinco categorias:

3.2.1) Multiplexação de Cliente

Nesta metodologia o navegador do usuário ou o servidor de *proxy* da rede recebe uma lista com os endereços de acesso dos servidores CDNs. A maneira como ele irá escolher em qual servidor se conectar depende da maneira utilizada para formar esta lista. Existem três maneiras de produzi-la:

1. No primeiro caso o servidor de DNS que atende ao cliente entrega a ele uma lista contendo os endereços IPs dos servidores de *cache* que possuem o objeto desejado. Nenhuma mudança é requerida no cliente, mas como ele apenas recebe a lista cabe a ele determinar através da medição da rede efetuada qual é o melhor servidor dentre os informados para que haja a comunicação.
2. No segundo caso uma aplicação em Java é a responsável por montar a lista de servidores. Neste caso é necessário que a aplicação seja instalada no cliente para que haja a comunicação.
3. No terceiro caso as informações sobre os servidores são transmitidas no cabeçalho HTTP. Neste caso é necessária a alteração da metodologia de comunicação tanto por parte dos servidores *web* como nos navegadores utilizados pelos clientes.

3.2.2) DNS Indireto

Os servidores de DNS permitem o mapeamento de um domínio da *internet* para vários endereços IPs distintos e a escolha de um deles para que ocorra a comunicação entre o cliente e o servidor que hospeda o domínio. A diferença no uso de DNS indireto esta no fato que a escolha do servidor de *cache* ocorre no próprio site da *web* ou na infraestrutura de DNS responsável pela resposta do endereço IP do domínio, ao invés do servidor de DNS do cliente ser o responsável pela pesquisa.

3.2.3) Redirecionamento HTTP

O protocolo HTTP permite que um servidor *web* responda a requisição de um cliente com uma mensagem que diga a ele para reenviar a requisição para outro servidor. O seu refinado gerenciamento o torna vantajoso ao permitir o uso de inúmeras configurações e opções. Sua desvantagem esta na sobrecarga de dados gerada pela retransmissão da requisição na rede.

3.2.4) Anycasting

O *anycast* provê uma comunicação ponto a ponto entre o cliente e o primeiro servidor mais próximo que estiver dele. Com a distribuição de vários servidores na *internet* com o mesmo IP e baseado no fato que os roteadores próximos a eles sabem como contatá-los o *anycast* repassa ao roteador a tarefa de conectar o cliente ao servidor baseado na tabela de roteamento do roteador.

3.2.5) Roteamento Entre Pares

Neste sistema as informações da rede são construídas na troca de comunicação entre os membros da rede com eles mesmos ao invés de utilizar outra abordagem de comunicação para a montagem da rede. Isto proporciona uma rede mais tolerante a falhas de comunicação do que nas demais formas de roteamento.

3.3) Serviço De Requisição

Dentro da rede sobreposta de servidores CDNs o serviço de requisição é composto por duas partes: localizar o servidor de *cache* mais adequado que possua

a replica do objeto solicitado e o redirecionamento da requisição para o servidor adequado.

No que diz respeito à localização do servidor a primeira escolha de como distribuí-lo na rede é se as informações referentes à sua localização serão recebidas de forma reativa ou proativa. Em seguida é determinado se o roteamento é realizado na camada de roteamento ou na camada de aplicação do modelo TCP/IP, e por fim é comparado o custo de utilizar a tabela de “roteamento baseada em voto” contra a coleta de informações através de “sondagem da rede”.

3.3.1) Coleta Reativa Versus Proativa

Broadcast e *multicast* são abordagens tipicamente reativas para localizar o servidor desejado.

3.3.2) Camada De Roteamento Versus Camada De Aplicação

Um ambiente proposto com mecanismos de *anycasting* na camada IP se torna atraente porque este modelo de redirecionamento de pacote evita *link* de comunicação sobrecarregado e utiliza o melhor caminho para alcançar o alvo. Em contrapartida não é possível determinar nenhuma informação da taxa de uso do servidor, o que acarreta no uso de outra ferramenta para esta medição e no aumento do tráfego de informações na rede.

O roteamento na camada de aplicação permite incluir informações sobre a taxa de uso de cada servidor sem aumentar a quantidade de informação na rede. Sua desvantagem esta no fato de ao se utilizar o redirecionamento no nível da camada de aplicação requerer atualização periódica de sua estrutura, enquanto no roteamento da camada de rede a base de dados da localização dos servidores é construída incrementalmente na medida em que o tráfego de rede é monitorado.

3.3.3) Tabela De Roteamento Por Voto Versus Sondagem Da Rede

Através da tabela de roteamento por voto é possível construir um gráfico de conectividade ao medirem-se todos os dados que estão envolvidos para que um servidor base alcance outro servidor. No uso da sondagem da rede, alguns servidores de medição são os responsáveis por determinar a distância entres os servidores que compõem a rede e, quando solicitado por um cliente o acesso a conteúdo, retornar a ele como resposta o endereço IP do servidor mais próximo que ele possuí em sua base de dados.

4) PESQUISAS SOBRE O TEMA

4.1) Universidade de Maryland

O departamento de ciências da computação da universidade de Maryland, Estados Unidos, procurou montar uma arquitetura que permitisse à reposta para seguinte pergunta:

> Dado um conjunto de *Multicast Service Nodes* (MSNs) distribuídos através de uma rede com largura de banda de acesso a *internet* limitados, como construir um *backbone* de multi-transmissão de dados tal que a latência sobreposta seja minimizada para os clientes. (Universidade de Maryland, 2003, pag.2).

A solução proposta se destina a prover um serviço de multi-transmissão de dados para um grande número de MSNs de transmissão de conteúdo *multimídia* em tempo real, que são sensíveis a grandes latências, através da criação de uma Infra-estrutura de redes sobrepostas de multi-transmissão (*Overlay Multicast Network Infrastructure (OMNI))*

Com a popularização do acesso a banda larga este tipo de transmissão tem se tornado frequente, com *sites* transmitindo eventos, jogos, programas de televisão e shows. Porém, em comparação ao conteúdo estático a transmissão ao vivo tem uma desvantagem: ela não permite o armazenamento em *cache* dos dados para aperfeiçoar a comunicação e diminuir o uso da rede desde o provedor do conteúdo até o cliente que o solicita. A qualidade do *streaming* é prejudicada por dois fatores: a carga de acesso experimentada pelo servidor e o atraso enfrentando na entrega dos dados na transmissão ponto a ponto. Devido a esta limitação é necessário prover um caminho eficiente que interliga o emissor e o receptor.

Para solucionar este problema foi formulado um sistema que analisa a latência média entre os servidores de MSN, fazendo a sua diminuição para o mínimo possível baseado na quantidade de usuários conectados nos servidores e na sua importância nesta comunicação. A solução empregada se adapta dinamicamente de acordo com a importância da estrutura da árvore de CDN, melhorando assim o caminho percorrido pelos dados transmitidos de acordo com a relevância dele. Para tanto ela se preocupa com os pontos abaixo:

- Descentralização: Uma solução centralizada não é aplicável devido às mudanças nas condições da rede, aumento e diminuição de servidores na transmissão dos dados, a latência na comunicação entre eles e entre eles e o cliente.
- Adaptação: O OMNI deverá se adaptar as mudanças na rede, carga de uso e mudanças na infra-estrutura dos servidores.
- Viabilidade: Através de mudanças incrementais o OMNI deverá fazer a mudança na árvore de transmissão de dados de tal maneira que em dado momento ela deveria satisfazer todas as restrições nos diferentes MSNs que a compõe. Porém, uma variação nesta restrição poderia impactar na interrupção da comunicação com os clientes. Sendo assim a solução proposta deverá se adaptar através de vários passos sequenciais, para que através desta maneira ele possa se adequar a alguma mudança nas restrições dos servidores.

Segundo o artigo a solução descrita por ele satisfaz todas as propriedades mencionadas acima.

4.1.1) Sua Solução

Em um cenário de *webcast* típico as transmissões de *streaming* começam em data e hora predeterminada. Num primeiro momento os servidores de MSNs se organizarão para montarem a árvore de transmissão, mesmo não tendo qualquer informação sobre o tamanho da população de clientes à ser atendidos.

A formulação da árvore é realizada através do envio da requisição de participação (*JoinRequest*) pelo MSN para o servidor principal (*root*) e baseado na latência encontrada no *JoinRequest* entre eles é realizada a montagem do caminho dos dados a serem transmitidos.

O algoritmo provê uma solução plausível para a montagem da árvore usando apenas a medida da latência entre o servidor MSN e o servidor *root*. Esta árvore continuará a ser transformada conforme começar a transmissão dos dados, a entrada e saída de outros MSNs e a alteração das condições da rede, porém a árvore continuará se adaptando para prover a melhor solução possível.

Após a fase de construção da árvore inicial começa a fase de modificação da árvore para adaptação as mudanças do cenário. Foram definidas cinco mudanças possíveis na estrutura da árvore para atender as suas necessidades de otimização.

- Promoção dos filhos: É realizada para determinar a primeira mudança após a criação da árvore. É definido quem é o grande pai, sendo este o servidor primário na árvore, depois o primeiro nível de servidores abaixo dele são denominados de servidores pai e todos os que estiverem abaixo deles são denominados como servidores filhos.
- Mudança entre pai e filho: Neste caso é realizada a mudança na árvore no posicionamento entre um servidor pai e um servidor filho. Sendo assim para melhorar a rede o servidor filho assume o lugar do pai e vice-versa.
- Mudança de nível 2: Ela ocorre quando dois servidores no mesmo nível fazem a troca de suas posições e herdam os servidores filhos nesta mudança

- Transferência de nível 2: Representa a mudança de dois servidores filhos entre os seus servidores pais. Diferente da mudança de nível 2 neste caso não existe a mudança de todos os servidores filhos.
- Mudança de nível 1 para nível 2: Ela ocorre quando dois servidores de MSN que não estão no mesmo nível e também não estão na mesma ramificação fazem a mudança entre eles de posição na árvore.

4.2) Instituto de Tecnologia de Massachusetts

Já o Instituto de Tecnologia de Massachusetts (MIT), através da pesquisa realizada pelos pesquisadores David Karger, Alex Sherman , Andy Berkheimer, Bill Bogstad, Rizwan Dhanidina, Ken Iwamoto, Brian Kim, Luke Matkins, Yoav Yerushalmi, e publicada no artigo "*web caching with consistent hashing*", propôs uma solução voltada para a transmissão de dados de conteúdo estático através do uso de servidores de *cache*.

Servidores de *cache* têm sido usados para aumentar a eficiência e a segurança na transmissão dos dados através da *internet*. Um servidor entre o usuário e o servidor *root* pode armazenar uma cópia do conteúdo solicitado para distribuí-la a outros usuários de forma mais rápida, principalmente se o servidor *root* estiver sobrecarregado ou o caminho na rede estiver congestionado. Porém, por melhor que pareça a idéia de prover um servidor de *cache* para um grupo de vários usuários com o intuito de melhorar o desempenho da transmissão, existem vários empecilhos nesta solução. Caso o servidor de *cache* falhe então todos os usuários conectados a ele não conseguirão transmitir e receber dados nem entre ele e o servidor *root*. Além disto, existem as limitações da quantidade de conteúdo que o *cache* pode armazenar em função do seu espaço em disco disponível e a quantidade de usuários que o servidor de *cache* pode atender, o que impacta no *hit rate* (taxa de acertos) entre o objeto solicitado e o fato de ele estar armazenado.

Na tentativa de prover uma solução para as dificuldades apresentadas tem se utilizado do recurso de servidores de *cache* cooperados. Neste cenário toda a requisição é enviada para o servidor de *cache* primário. Caso ele não tenha o conteúdo desejado, ao invés de encaminhar a solicitação para o servidor *root*, a

requisição é redirecionada para os servidores secundários de *cache* que estão cooperados com o primário. Esta solução diminui a quantidade de solicitações não atendidas pelo servidor primário.

Mas esta metodologia tem seus pontos falhos. Para descobrir qual servidor do segundo nível possuí o conteúdo alguns esquemas de *cache* cooperados realizam pesquisas baseadas em *broadcast*. Além de este método consumir bastante recurso da rede o servidor primário deve aguardar todos os secundários reportarem as suas falhas na resposta ao conteúdo solicitado antes de redirecionar a requisição para o servidor *root*. Outro problema é a quantidade de objetos duplicados entre os servidores de *cache*. Qualquer servidor com espaço em disco pode armazenar um objeto que foi solicitado a ele, mas que ele não tinha. Isto gera perda de tempo, espaço em disco e excesso de transmissão de dados. No pior caso esta duplicação impedirá, por falta de espaço em disco, que novos objetos sejam armazenados para transmissões futuras.

Como solução os pesquisadores desenvolveram uma metodologia de pesquisa baseada em *hash*. Seu trabalho provê a eliminação da comunicação entre os servidores cooperados no caso do objeto solicitado não se encontrado no servidor de *cache* primário. Quando o objeto não é encontrado ao invés do *cache* primário redirecionar a requisição para o *cache* de segundo nível o próprio navegador da *web* utilizado pelo usuário será capaz de determinar pelo uso de funções de *hash* qual servidor secundário possuí o objeto e contatá-lo diretamente, ou no caso de ele determinar que nenhum *cache* proverá o conteúdo então solicitá-lo diretamente ao servidor *root*. As funções de *hash* implementadas ajudam o navegador a mapear e decidir qual será o *cache* que irá prover o objeto requisitado. Esta implementação necessita apenas de uma única transmissão para determinar com exatidão se o objeto esta ou não armazenado, diminuindo assim o tráfego na rede. Isto também diminuiu o tempo necessário para descobrir se um servidor de *cache* irá ou não responder a requisição. Como forma de diminuir a sobrecarga dos servidores a pesquisa inovou ao fazer com que a tarefa de pesquisa seja realizada pelo navegador, ao invés dos servidores. Como a pesquisa é direcionada a um único servidor ela diminuiu a quantidade de falhas na resposta, pois apenas o servidor primário responderá com uma falha a solicitação do objeto, mesmo que os servidores secundários também não possuam o objeto.

5) UM EXEMPLO DE CDN NO MUNDO REAL

A Akamai, empresa que vende o serviço de CDN para outras empresas, foi fundada em 1998 em Cambridge, Massachusetts, nos Estados Unidos da América. Empresa de sucesso neste ramo, possuí mais de 15.000 servidores operando em 69 países e 1000 redes diferentes (ACMS: The Akamai Configuration Management System, 2005, pg.1**)** para prover o *cache* de dados a qualquer companhia interessada em distribuir de forma mais eficiente seus conteúdos para os usuários que o requisitam. Isto é feito de forma transparente para o usuário, que para vários *sites* além do conteúdo *multimídia* ele também recebe da Akamai outros conteúdos tais como gráficos, figuras e textos. O benefício se dá no fato de receber o conteúdo de um servidor mais próximo dele, ou até mesmo de um que provenha a melhor taxa de transmissão de dados, permitindo o rápido acesso as informações. Seus servidores são distribuídos próximos a *backbones* com grande volume de tráfego de informação. Sites como Yahoo, IBM, The New York Times e FedEx não seriam capazes de prover aos seus usuários um canal de comunicação eficaz ao transmitir seus *giga bytes* de figuras, textos, animações em *flash* e vídeos por segundo sem hospedá-los na Akamai.

A Akamai construiu sua própria estrutura de DNS. Baseado no conceito de DNS Indireto ela garante a rápida reposta à solicitação do conteúdo ao utilizar resolução de nome indireta do endereço eletrônico para o endereço IP do servidor CDN mais próximo.

5.1) Localizador De Recurso Akamai

Foi desenvolvida pela empresa uma aplicação capaz de fazer a resolução de nome (DNS) do endereço eletrônico (URL) de uma figura, por exemplo, em um novo endereço eletrônico que apenas a Akamai é capaz de distinguir. O software Akamaizer cria o que eles chamam de Localizador de Recurso Akamai (*Akamai Resource Locators* - ARL). A ARL contem campos adicionais que auxiliam na

localização do objeto. A figura 7 apresenta um exemplo de ARL da figura *www.yahoo.com/logo.jpg*.

Figura 7: Exemplo De Uma ARL

- Número Serial: identifica um recipiente virtual de conteúdo. É construído um recipiente que armazenará e apenas ele armazenará um conjunto de objetos que serão entregues pelos mesmos servidores de *cache*.
- Domínio Akamai: endereço de algum dos servidores da Akamai. Ao se realizar a resolução de endereço eletrônico da Akamai para o endereço IP deste é garantido que o objeto requisitado não será solicitado no servidor original.
- Tipo: adiciona um interpretador na ARL.
- Código do provedor: identificador único do cliente da Akamai.
- Data do Objeto: Contem a data ou o *hash* MD5 da data de expiração do objeto
- URL absoluto: contêm o endereço original do objeto, utilizado para acessar o objeto no servidor original quando for necessário copiá-lo para o servidor de *cache*.

5.2) O Sistema de DNS da Akamai

Ao solicitar conteúdo do endereço *www.yahoo.com* o servidor de DNS do usuário reencaminhará a solicitação para o sistema de DNS da Akamai devido a

uma entrada do tipo nome canônico (*Canonical Name* – CNAME) que referencia a localização correta do objeto requisitado, desta maneira a solicitação chega ao servidor primário de DNS da Akamai. Neste momento utilizando-se do campo domínio Akamai o sistema de DNS determina qual é o melhor servidor dentro da sua hierarquia a entregar o conteúdo para o usuário. Ao contrário de outros sistemas de DNS o deles utiliza fatores como condição da rede, taxa de utilização do servidor e localização de origem do usuário.

Sua implementação se dá em dois níveis. O primeiro nível é responsável por receber a solicitação do usuário e localizar qual é o servidor secundário mais próximo do cliente. O segundo nível é responsável por determinar a localização mais próxima do objeto requisitado em relação ao usuário e redirecionar o seu acesso. Este processo ocorre em um intervalo de tempo relativamente pequeno devido ao contínuo monitoramento de todos os servidores.

O sistema de resolução de nomes da Akamai permite o armazenamento em *cache* das resoluções DNS providas, de forma similar ao sistema convencional de DNS. Seu propósito é eliminar o excesso de pesquisas pelo mesmo objeto, diminuindo também o gargalo gerado por vários níveis hierárquicos. O tempo de vida da resolução DNS (*Time to Live* - TTL) é configurado baseado no nível que ele se encontra. No primeiro nível o período válido é de 20 minutos, enquanto no segundo nível é de 10 segundos devido ao maior número de servidores neste nível e consequentemente na maior variação da taxa de utilização destes servidores.

5.3) O Sistema de Gerenciamento de Configuração

A Akamai dispõe de um sistema capaz de medir diversos parâmetros da rede, chamado de Sistema de Gerenciamento de Configuração (SGC). Através desta monitoria é construído um arquivo de configuração com as informações necessárias para atualizar ou reconfigurar vários subsistemas de sua rede, como o caminho a ser percorrido em função da mudança de rota de acesso e latência no trajeto ou o que deve ser armazenado em *cache* de um determinado cliente em determinados *caches*. Sua função é manter atualizado o máximo de seus servidores membros com as últimas informações disponíveis para que a entrega do objeto requisitado pelo

usuário ao servidor seja a mais rápida possível. A replicação deste arquivo tanto é realizada através do envio pela central das informações sobre as alterações necessárias a todos os servidores como a apenas um pequeno conjunto de servidores.

O seu funcionamento consiste primeiramente no envio da mensagem de atualização pelo Publicador para o Ponto de Armazenamento (PA) do SGC. O PA que recebe a mensagem realiza uma checagem para garantir que esta é referente à última versão do arquivo de atualização e posteriormente grava em disco o novo arquivo recebido. Em paralelo ao armazenamento é executado um algoritmo chamado Vetor de Troca que permite configurar um grupo de PA para receberem a mesma atualização. Somente após a validação do algoritmo por parte dos PAs é enviado ao publicador a resposta de aceitação da atualização.

Quando a atualização esta disponível em todos os PAs estes o oferece para download através do protocolo HTTP para todos os demais servidores da Akamai. Cada um destes executa um serviço chamado Recebedor que coordena a comunicação e o recebimento no servidor do arquivo de atualização disponibilizado pelo PA mais próximo dele. De posse deste arquivo o servidor aplica a atualização necessária em suas configurações, conforme solicitado pelo publicador. A figura 8 demonstra de maneira simplificada todo o processo de funcionamento.

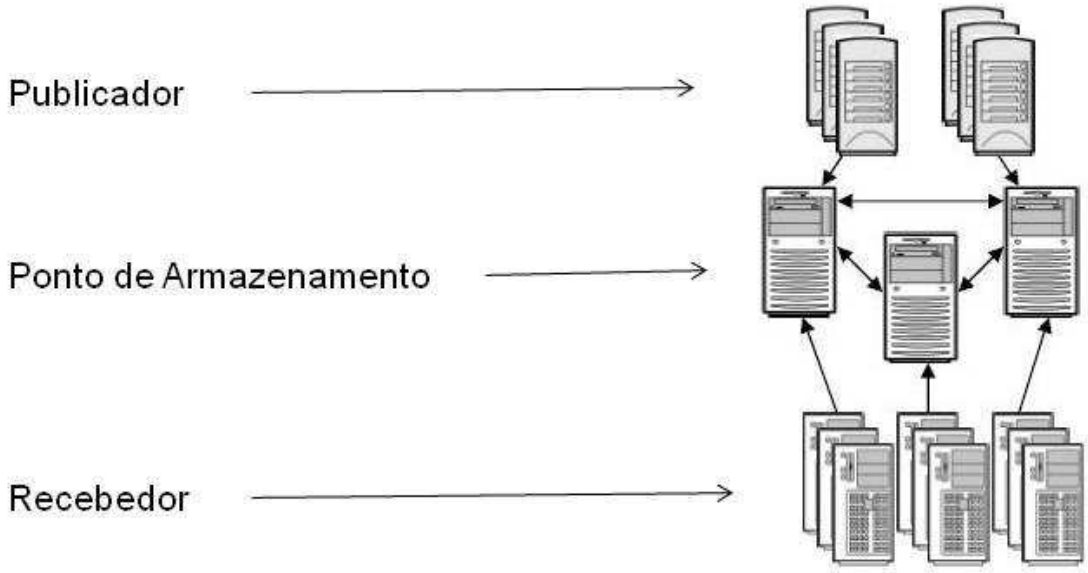

Figura 8: Estrutura de funcionamento do SGC

6) CONSIDERAÇÕES FINAIS

Em um ambiente tão diversificado e sem padrões tecnológicos homogêneos como a *internet* se faz necessário o desenvolvimento de novas metodologias que atendam as atuais e futuras necessidades dos usuários, empresas e da própria rede de dados. Prover acesso a conteúdo para mais de 2.095.000.000 de pessoas no mundo (*http://www.internetworldstats.com/stats.htm*, acessado em 21 de Outubro de 2011)é uma tarefa árdua, que envolvem investimentos em servidores e seus componentes, canais de comunicação com a internet, *backup* dos dados, funcionários, energia elétrica e alguns outros mais, agregando um custo altíssimo para as empresas. Para dificultar ainda mais existe toda a sobrecarga gerada pelo acesso centralizado aos dados. Se os vídeos disponibilizados pelo site *www.youtube.com* estivessem apenas localizados na sua sede em San Bruno, Califórnia, a probabilidade da *internet* nos Estados Unidos da América ficar extremamente lento é altíssimo. Vale ressaltar que o youtube em 2006 foi o responsável por cerca de 60% de todo o acesso a vídeo na *internet*, isto equivalendo em torno de 100 milhões de vídeos assistidos diariamente (*http://www.usatoday.com/tech/news/2006-07-16-YouTube-views_x.htm, acessado em* 21 de Outubro de 2011).

Enquanto não houver em todo o mundo acesso a *internet* com taxa de acesso suficiente para prover acessibilidade a qualquer conteúdo sem a necessidade de armazená-lo em várias localidades será preciso o emprego de alguma solução capaz de minimizar o tempo de acesso aos dados entre o usuário e o servidor. Neste ponto o CDN demonstra ser uma solução ágil, robusta e escalável para melhorar a comunicação em ambos os lados. Chega ao ponto de ser tema de vários trabalhos acadêmicos em universidades renomeadas como o departamento de ciências da computação do Tennessee e o de Washington nos Estados Unidos da América, departamento de elétrica e engenharia elétrica da universidade de ciências e tecnologia de Hong Kong e o laboratório de ciências da computação do Instituto de Tecnologia de Massachusetts (conhecido comumente como MIT). Esta rede possuí algoritmos poderosos para determinar quais vídeos deverão ser enviados aos servidores de *cache* devido a sua alta taxa de acesso na região, algoritmos de

indexação para eliminar objetos duplicados entre os *clusters* de servidores e metodologias de comunicação que minimizam a sobrecarga de encaminhamento de solicitações para diminuir o tempo de resposta do usuário.

Sem dúvida além do seu sucesso entre os provedores de conteúdo o seu tempo de vida como solução para melhorar a qualidade dos serviços prestados na *internet* será longínquo.

6) BIBLIOGRAFIA

Jacobson, V.; Smetters, K., D.; Thornton, D., J.; Plass, F., M.; Briggs, H., N.; Braynard, L. - **Networking Named Content**. Disponível em: <HTTP://pages.cs.wisc.edu/~akella/CS838/F09/838-Papers/ccn.pdf> acessado em 07/08/2011.

Banerjee, S.; Kommareddy, C.; Kar, K.; Bhattachargee, B.; Khuller, S. - **Construction of an Efficient Overlay Multicast Infrastructure for Real-Time Application**. Disponível em: <HTTP://www.google.com.br/url?sa=t&rct=j&q=construction%20of%20an%20efficient%20overlay%20multicast %20infrastructure%20for%20real-time%20application&source=web&cd=3&ved=0CDoQFjAC&url=HTTP%3A%2F%2Fciteseer.ist.psu.edu%2Fviewdoc%2Fdownload%3Bjsessionid%3D05EFC7CB49EAA53D0DE521398EDD7249%3Fdoi %3D10.1.1.13.7471%26rep%3Drep1%26type%3Dpdf&ei=ommjTsn_BIbpgQe7hoTcBA&usg=AFQjCNEfJ2LgY6uHwNLkgdTwWBVKDOfS5A&cad=rja> acessado em 29 de Julho de 2011.

Peng, G. - **CDN: Content Distribution Network**. Disponível em: <HTTP://arxiv.org/PS_cache/cs/pdf/0411/0411069v1.pdf> acessado em 15 de Agosto de 2011.

Ni,J.; Tsang, K., H., D.; Yeung, H, S, I.; Hei, X. - **Hierarchical Content Routing in Large-Scale Multimedia Content Delivery Network**. Disponível em : <HTTP://www.ifp.illinois.edu/~jianni/publication/icc03-content%20routing.pdf> acessado em 15 de Agosto de 2011.

Hofmann, M.; Beaumont, L. - **Content Networking, Architecture, Protocols, And Practice**. Disponível em: <HTTP://books.google.com.br/books?hl=pt-BR&lr=&id=DWXfwclewjwC&oi=fnd&pg=PP1&dq=Content+Networking+-+Architecture,+Protocols,+and+Practice&ots=0KTpw4i3vV&sig=rMDbWoX5pqikF6Ninxbq-Mzd_DE#v=onepage&q&f=false> acessado em 02 de Setembro de 2011.

Karger, D.; Sherman, A.; Berkheimer, A.; Bogstad, B.; Dhanidina, R.; Iwamoto, K.; Kim, B.; Matkins, L.; Yerushalmi, Y. - **Web Caching With Consistent Hashing**. Disponível em: <HTTP://www.cs.brown.edu/courses/csci2950-u/f10/papers/chash99www.pdf> acessado em 16 de Setembro de 2011.

HTTP://www.usatoday.com/tech/news/2006-07-16-YouTube-views_x.htm acessado em 21 de Outubro de 2011.

HTTP://www.internetworldstats.com/stats.htm acessado em 21 de Outubro de 2011.

Sherman,A.; Lisiecki, A., P.; Berkheimer, A.; Wein, J.; Akamai Technologies; Columbia University; Polytechnic University - **ACMS: The Akamai Configuration Management System.** Disponível em: <HTTP://www.akamai.com/dl/technical_publications/ACMSTheAkamaiConfiguration ManagementSystem.pdf> acessado em 13 de novembro de 2011.

Su,A.; Choffnes, R., D.; Kuzmanovic, A.; Bustamante, E., F. - **Drafting Behind Akamai: Inferring Network Conditions Based on CDN Redirections.** Disponível em: <HTTP://networks.cs.northwestern.edu/publications/ton-Akamai.pdf> acessado em 13 de novembro de 2011.

7) GLOSSÁRIO

Anycast: método de transmissão de dados do tipo "um para um" onde o pacote é direcionado para o destino mais próximo.

Backbones: equipamento responsável por interligar várias redes geograficamente dispersas.

Canônico: normativo, que estabelece uma regra ou relação entre duas coisas.

Hash: sequência de dados que identifica uma palavra ou texto que não possa ser divulgado publicamente, garantindo assim a sua integridade.

Hop counts: unidade de medida para a contagem de por quantos roteadores um pacote de dados trafegou durante a comunicação entre dois computadores.

Latência: unidade de medida referente ao atraso da resposta de uma solicitação de comunicação.

Multiplexação: divisão de um sinal de origem em vários novos sinais que são encaminhados a caminhos diferentes no meio de comunicação.

Proxy: equipamento intermediário de filtragem de dados na comunicação entre dois computadores.

Réplica: duplicação.

Root: sinônimo de principal, começo de algo, primeiro.

Roteador: equipamento responsável por interligar redes de dados diferentes.

Round trip: unidade de medida para a contagem do tempo levado ao completar-se a comunicação entre dois computadores.

Streaming: conteúdo *multimídia*, video, som.

Webcast: transmissão de video na *internet*.

.