Impacto das políticas de replicação na disponibilidade de documentos em redes P2P sob ataques DoS

Michele Nogueira Lima[1] **, Dorgival Olavo Guedes Neto**[1]

[1] Departamento de Ciência da Computação - DCC
Universidade Federal de Minas Gerais
Belo Horizonte, MG

{michele,dorgival}@dcc.ufmg.br

Abstract. *During the last decades the Internet and the Web have become very important for people and companies. However, the Internet is vulnerable to malicious attacks such as denial of service (DoS) attacks that are difficult to detect and to prevent. That has led to studies on attack-tolerant systems, usually based on P2P networks. This work studies the behavior of one of these systems in relation to the different content replication strategies used by the network nodes. The target system is simulated using loads and topologies realistic for the proposed application and three replication strategies are considered: uniform, proporcional and square-root. The results indicate that in face of events that may cause the failure of a large number of nodes (between 30 % and 80 %), proportional replication offers results on the order of 10 % better than the other strategies. This result differs from previous results in the literature which did not consider events causing major node loss, pointing to the need to consider such events when content availability must be garanteed against large scale attacks.*

Resumo. *Durante as últimas décadas a Internet e a Web tornaram-se bastante utilizadas por pessoas e empresas. Entretanto, a Internet é vulnerável a ataques maliciosos como ataques de negação de serviço (DoS) que são difíceis de detectar e evitar. Isso tem levado ao desenvolvimento de sistemas tolerantes a ataques, usualmente baseados em redes P2P. Este trabalho avalia o comportamento de um desses sistemas em função de diferentes políticas de replicação de conteúdo entre os nós da rede. O sistema é simulado considerando-se uma topologia e carga realistas para a aplicação proposta e três técnicas de replicação são analisadas: uniforme, proporcional e square-root. Os resultados indicam que na presença de eventos que causem a falha de uma fração significativa dos nós (entre 30 % e 80 %) a replicação proporcional oferece resultados da ordem de 10 % superiores às demais políticas. Esse resultado difere de resultados anteriores da literatura obtidos sem a consideração de eventos de indisponibilidade de vários nós, indicando a necessidade de se considerar esse tipo de evento em casos onde a disponibilidade do conteúdo deve ser garantida contra ataques em larga escala.*

Palavras-chave: negação de serviço, servidores web, tolerância a ataques, redes *peer-to-peer*;

1. Introdução

Durante as últimas décadas a Internet e a *World Wide Web* (Web ou WWW) tornaram-se bastante utilizadas por pessoas e empresas, passando a ser ferramentas essenciais na disseminação de informações. Atualmente, empresas como a *CNN.com* e eventos como as Olimpíadas ou a Copa do Mundo de Futebol possuem uma presença marcante na Internet. Entretanto, a rede mundial de computadores é bastante vulnerável a ataques maliciosos [de Vivo et al., 1999]. Esses ataques comprometem o funcionamento da Internet através da interrupção de seus serviços, acarretando prejuízos para pessoas e empresas.

Os tipos mais comuns de ataques maliciosos, devido à dificuldade de defesa, são os ataques de negação de serviço – *Denial-of-Service* (DoS). Eles têm como objetivo inviabilizar o acesso a um serviço da Internet através do esgotamento dos recursos do servidor (memória, CPU, largura de banda dos canais de acesso ou espaço em disco), provocando uma sobrecarga no mesmo ou na rede [Dean et al., 2001, Gil and Poletto, 2001, Mohiuddin et al., 2002, Peng et al., 2003]. Um dos problemas da defesa contra tais ataques está no fato deles muitas vezes serem bastante semelhantes a eventos de *flash crowd*[1] legítimos, os quais devem ser tolerados pelos servidores [Jung et al., 2002]. Atualmente, a maioria dos métodos utilizados para tratar de DoS como sistemas de gerência de rede pró-ativos e sistemas de detecção de intrusos são inadequados, ineficientes ou muito restritivos [Schäfer, 2002]. Assim sendo, uma possibilidade cada vez mais considerada para se lidar com esse problema é o desenvolvimento de técnicas que permitam aos servidores tolerar ataques sem deixar de atender seus usuários, mesmo que ao custo de uma certa redução da qualidade do serviço.

Em específico para o serviço WWW, atualmente a maior parte das soluções propostas a fim de tornar este serviço tolerante tem como princípio básico de operação a utilização de um número de máquinas da rede que operam como "servidores Web substitutos" para o caso de um servidor ser retirado do sistema devido a um ataque. Duas questões importantes a serem tratadas nesse caso são o direcionamento das requisições para esses servidores substitutos e a distribuição e replicação do conteúdo do servidor original. Atualmente, as principais técnicas utilizadas nos sistemas de tolerância a ataques para a distribuição de requisições são a migração de conexões e as redes *Peer-to-Peer* (P2P), sendo esta última a que tem apresentado melhores resultados.

As redes *Peer-to-Peer* são redes virtuais que oferecem uma estrutura descentralizada para acesso aos recursos das máquinas participantes. Os sistemas de tolerância a ataques as utilizam como forma de dispersar as requisições feitas a um servidor sob ataque, uma vez que essas requisições podem ser redirecionadas para qualquer membro da rede virtual. A vantagem no uso dessas redes quando comparado a outras abordagens, como sistemas de arquivos paralelos/distribuídos tolerantes a falhas, consiste na flexibilidade e na simplicidade de manter o funcionamento do sistema mesmo quando uma grande porcentagem de nós tornam-se falhos.

Uma das questões principais na estruturação de uma rede virtual P2P é a forma de distribuição e replicação da informação entre os nós da rede. Isso se deve ao fato de que a existência de cópias dos objetos em mais de um nó aumenta as chances do sistema

[1]Eventos de *flash crowds* são definidos como um aumento inesperado no número de requisições por um documento de um sítio Web podendo sobrecarregar o servidor Web.

ser capaz de atender a todas as requisições mesmo que ocorra a queda de alguns nós. As pesquisas nessa área têm considerado principalmente as redes P2P para compartilhamento de arquivos e não levam em conta o problema da retirada simultânea de vários nós da rede normalmente associada a um ataque DoS, nem as distribuições de popularidade associadas a objetos da WWW.

Este trabalho avalia o comportamento das políticas de replicação propostas na literatura para redes P2P quando confrontadas com o problema de distribuir o conteúdo de sítios WWW e frente a ataques DoS que podem desabilitar uma parte dos nós da rede. O principal foco das avaliações consiste na medição da disponibilidade do serviço e do conteúdo Web diante de ataques DoS, haja visto que a idéia do sistema proposto na seção 3 é tornar sistemas Web tolerantes a ataques, mesmo que em alguns momentos existam aumentos na carga gerada e na latência.

O restante deste artigo está organizado da seguinte forma: a seção a seguir discute os principais trabalhos relacionados. A seção 3 apresenta a arquitetura do sistema de tolerância a ataques proposto e sua operação geral. A seção 4 discute a metodologia utilizada no estudo, incluindo as premissas principais do modelo desenvolvido e sua simulação. A seção 5 apresenta os resultados obtidos e finalmente a seção 6 apresenta algumas conclusões e discute possíveis trabalhos futuros.

2. Trabalhos relacionados

Um dos primeiros exemplos de trabalhos na linha de sistemas distribuídos para tolerância a ataques DoS na Internet é apresentado por de Kong e Ghosal (*pseudo-serving* [Kong and Ghosal, 1997, Kong and Ghosal, 1999]). Neste trabalho uma estrutura complexa baseada em alterações nos programas navegadores (*browsers*) fazia a distribuição de requisições entre clientes que já haviam requisitado determinado conteúdo. A técnica utilizada nesse caso é a de migração de conexões: quando um servidor atinge uma certa capacidade limite de processamento ou de número de conexões, as conexões subseqüentes e algumas já existentes são migradas para outra máquina que possa atender às requisições. Sua principal desvantagem é a complexidade para se manter os estados das conexões migradas. O sistema NETSEC também utiliza esse princípio, porém se baseia em uma rede de servidores replicados, ao invés de depender do conteúdo acessado e colocado na *cache* por clientes [Sangpachatanaruk et al., 2003].

A maior parte dos trabalhos mais recentes, entretanto, utiliza fortemente os princípios de redes P2P para simplificar o processo de distribuição de requisições, valendo-se das características intrínsecas daquelas redes. CoopNet, por exemplo, é uma proposta de uma rede de clientes em princípio semelhante ao esquema de *pseudoserving*, entretanto utilizando protocolos P2P [Padmanabhan and Sripanidkulchai, 2002].

Os membros de uma rede *Peer-to-Peer* podem ser desde servidores com maior capacidade até simples computadores domésticos, todos cooperando entre si. As redes possuem uma estrutura descentralizada e são classificadas em dois grupos em função da forma como organizam as informações entre os nós. Caso definam uma estrutura organizada para distribuição do conteúdo entre os nós são denominadas redes estruturadas, caso contrário, se permitem que objetos sejam distribuídos de qualquer forma entre os nós, são denominadas não-estruturadas [Graham, 2001]. Para que um membro da rede

possa responder a uma requisição ele precisa localizar uma cópia dos dados relacionados à requisição. O comportamento do processo de busca depende fortemente da forma como documentos são distribuídos e replicados na rede.

No sistema PROOFS [Stavrou et al., 2002], uma rede P2P não-estruturada — como a rede Gnutella — é utilizada em conjunto com um protocolo de construção da rede virtual para aliviar os efeitos de eventos de *flash crowd* sobre servidores web. O protocolo de construção da rede realiza periodicamente uma troca aleatória dos vizinhos pertencentes aos nós participantes da mesma, aumentando a probabilidade de um objeto ser encontrado. O sistema *Backslash* [Stading et al., 2002] é outro trabalho que considera o problema de aliviar efeitos de eventos de *flash crowd* sobre servidores web. Este sistema também utiliza uma rede P2P, porém esta é formada apenas por outros servidores web cooperando entre si. Esse tipo de cooperação em certos casos pode representar um ganho de desempenho em função da maior disponibilidade para todos os participantes [Villela and Rubenstein, 2003].

As principais formas de replicação de conteúdo em redes P2P são a replicação uniforme, proporcional e "raiz quadrada" (*square-root*). Na replicação uniforme, cada objeto que se pretende disponibilizar é replicado em um número constante de nós da rede, independente de sua popularidade[2]. Já nas duas outras distribuições, os objetos são replicados com base em sua popularidade em relação aos demais objetos. Diversos trabalhos já estudaram essas técnicas no contexto de redes P2P para compartilhamento de arquivos sem considerar o efeito de ataques DoS. Em particular, Cohen *et al.* [Cohen and Shenker, 2002, Cohen et al., 2002] estudaram em detalhes as três distribuições mencionadas e provaram que em redes P2P em condições estáveis (sem a ocorrência de ataques DoS) a distribuição *square-root* é ótima em termos de redução dos tempos de busca. Outro trabalho que estuda essas distribuições é o de Wang, Pai e Peterson, porém o enfoque é o de redes de distribuição de conteúdo (CDNs) [Wang et al., 2002]. Entretanto, esses trabalhos não consideram o problema da queda de nós em decorrência de um ataque DoS massivo, que é o objetivo de nosso trabalho.

3. Arquitetura do sistema

O sistema de tolerância a ataques DoS considerado neste trabalho utiliza características das redes P2P não-estruturadas para descentralizar todas as requisições Web. O uso das redes não-estruturadas é justificado pela facilidade de formação da rede, pela independência entre o posicionamento dos dados e a topologia da rede e pela liberdade com que os membros da rede podem utilizar as mensagens de busca. Estas características aumentam a flexibilidade do sistema de tolerância, uma vez que uma rede estruturada é por natureza menos resistente a falhas que comprometam a estrutura de sua organização interna.

A rede virtual proposta para o sistema de tolerância é composta apenas por servidores web. No projeto considerado, ao contrário do sistema NETSEC mencionado anteriormente, os servidores não são réplicas do servidor central, o que exigiria um investimento

[2]Um documento muito popular é aquele que está dentre os mais requisitados e que possui uma grande probabilidade de ser requisitado novamente.

muito alto na implementação do sistema. Ao invés disso, o sistema pode ser composto por um grupo de servidores originalmente independentes que podem ser agrupados para formar uma cooperativa, de forma que todos os servidores se disponham a servir requisições por documentos de qualquer servidor do conjunto. O conteúdo de cada servidor será então distribuído e replicado para todos os servidores, que cederão parte do seu espaço de armazenamento e capacidade de processamento para a cooperativa. Requisições para qualquer dos servidores podem ser distribuídas para qualquer dos servidores do grupo utilizando-se uma versão adaptada de um servidor DNS (*Domain Name System*) ou por reescrita de URLs, de forma semelhante ao sistema *Backslash* [Stading et al., 2002].

Nessa seção a arquitetura do sistema de tolerância a ataques DoS e seu funcionamento são descritos. Para simplificar o entendimento, a discussão do sistema foi dividida em três partes: o problema de redirecionamento das requisições, a questão de localização dos recursos procurados (busca) e a análise das técnicas de replicação de conteúdo, objetivo maior deste trabalho. Na discussão que se segue, o sistema de tolerância a ataques DoS é em alguns momentos referenciado apenas por **sistema**, e os servidores participantes do sistema são normalmente denominados **nós**, como em uma rede P2P.

3.1. Redirecionamento das requisições

Cada nó do sistema é inicialmente um típico servidor web, possuindo as funcionalidades relacionadas à hospedagem de sítios web e à comunicação com os clientes utilizando o modelo tradicional navegador (*browser*)-servidor, como mostrado na figura 1-a. Além dessas funcionalidades tradicionais, são adicionados aos nós a capacidade de cooperarem entre si e de redirecionarem suas requisições para outros nós, descentralizando, desta forma, as requisições. Por exemplo, na figura 1-b, a requisição (1) é direcionada pelo navegador para um servidor web qualquer no sistema, que por sua vez pode ter que encaminhar uma consulta para um outro servidor a fim de obter o objeto requisitado, ocorrendo o mesmo com as requisições (2) e (3).

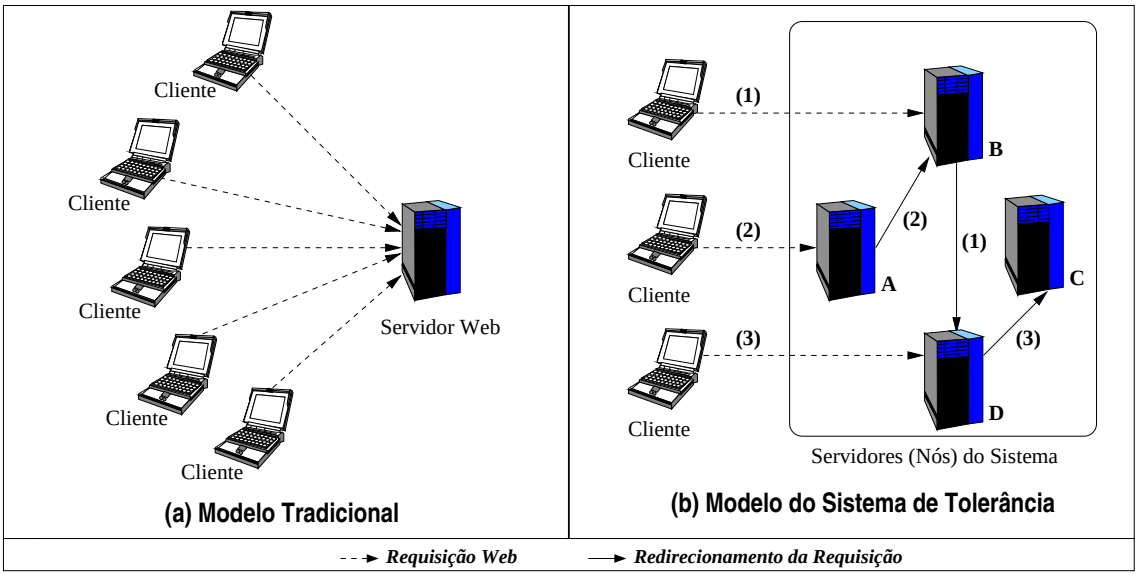

Figura 1: Modelo WWW tradicional x Sistema tolerante a ataques

Os redirecionamentos das requisições podem ser feitos utilizando reescrita de

URLs e/ou através do uso de um servidor especial de DNS. Esse processo combinado é discutido em detalhes para o caso do sistema *Backslash* [Stading et al., 2002]. Neste sistema de tolerância, os clientes são direcionados em um primeiro acesso pelo DNS, que monitora a disponibilidade dos servidores. A reescrita de URL auxilia apenas na redistribuição de carga entre os servidores. Para evitar que clientes fiquem presos indefinidamente a servidores sob ataque, técnicas de limitação do tempo de vida (TTL) das respostas do DNS são utilizadas. Essas técnicas, suas qualidades e limitações são descritas no artigo de Balachander et al. [Krishamurthy et al., 2003].

A localização de documentos baseada na reescrita de DNS é iniciada com o tratamento da requisição de DNS pelo sistema, que devolve para o cliente o endereço de um servidor qualquer dentro do conjunto de servidores. O cliente em seguida estabelece uma conexão HTTP para o servidor indicado e envia a requisição por uma URL. O servidor pode ou não possuir uma cópia do objeto solicitado. No primeiro caso o processo se passa como em um sistema normal, com o servidor respondendo diretamente ao cliente. Por outro lado, caso o servidor não possua o objeto solicitado, ele deverá realizar uma pesquisa entre os demais servidores do sistema em busca do documento. Essa pesquisa seguirá o formato das consultas em redes P2P. Uma vez tendo obtido uma resposta positiva de algum outro servidor que possua o objeto, o servidor originalmente contactado pelo cliente retorna para ele a informação solicitada. Caso a consulta na rede P2P de servidores não seja bem sucedida o servidor contactado retorna um código de erro HTTP para o cliente. A desvantagem desse método é o aumento da latência nas consultas, uma vez que a recuperação de um objeto pode acarretar uma busca em uma rede P2P não estruturada. Por outro lado, a natureza completamente descentralizada da rede torna-a bastante resistente a ataques, uma vez que não há nenhum ponto centralizador. Desde que a informação esteja disponível em algum nó ativo, ela pode ser encontrada por uma consulta na rede P2P. Alguns clientes podem observar a queda do servidor com o qual a conexão HTTP foi estabelecida, caso ele seja atacado durante o processamento de uma requisição, porém outros clientes podem ser protegidos pelo sistema de DNS, que passa a não utilizar os servidores desabilitados.

3.2. Localização de recursos

Uma vez que um servidor recebe uma requisição do navegador do cliente solicitando um documento de um dos sítios que fazem parte do sistema, ele passa a funcionar como um *gateway* entre o cliente HTTP e o conjunto de servidores que colaboram entre si. Esse conjunto de servidores também é chamado de parte de espelhamento do sistema devido ao fato dos servidores fazerem cópias ou “espelhos” de seus conteúdos em outros servidores do conjunto colaborativo.

O espelhamento dos conteúdos dos sítios no sistema é implementado utilizando uma rede *peer-to-peer* não-estruturada, nesse caso a rede Gnutella [Limeware, 2003]. Dessa forma a parte colaborativa do sistema é tratada como uma rede virtual utilizando as operações da rede Gnutella de inserção e exclusão de um nó e a operação de busca por um objeto.

As buscas por documentos nas redes Gnutella são realizadas através de um processo de difusão de mensagens denominado *flooding*, como mostra a figura 2-a. Quando um nó da rede recebe uma mensagem de busca por um objeto, por exemplo, o nó (4) da

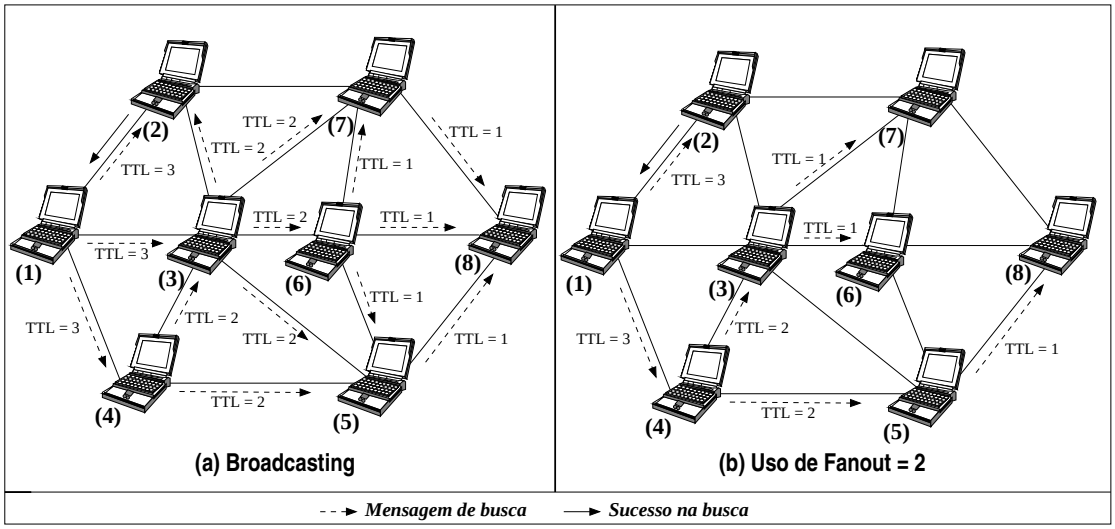

Figura 2: Localização de Documentos

figura 2-a, se o nó não consegue respondê-la diretamente, envia uma consulta a todos os nós conectados a ele, com exceção do nó que enviou a consulta original. Os nós subseqüentes seguirão o mesmo procedimento até que a mensagem possa ser retirada da rede ao ser recebida por todos os nós. Esse mecanismo é reconhecidamente pouco escalável para redes muito grandes. Para evitar a sobrecarga da rede, dois mecanismos já discutidos na literatura são utilizados [Lv et al., 2002]: a definição de um limite na propagação de consultas, denominado TTL (*time to live*), e a limitação no número de vizinhos a quem um nó repassa uma consulta recebida (*fanout*). O TTL especifica a distância máxima em termos de nós que uma mensagem pode atravessar a partir da origem antes de ser descartada e o *fanout* força um nó a escolher apenas alguns vizinhos para repassar a mensagem de busca. Como mostra a figura 2-b, uma mensagem de busca é inicializada com um determinado valor para o TTL, no caso do nó 1, TTL=3, sendo decrementada a cada nó atravessado da rede. Se o TTL de uma mensagem chegar ao valor zero, esta mensagem é retirada da rede.

Quando um nó possui o objeto procurado, este envia ao nó que originou a busca uma notificação de que o objeto foi encontrado. Na figura 2, o nó (1) procura por um objeto e inicia a busca; no nó (2) o objeto é encontrado e uma mensagem de sucesso de busca é enviada ao nó inicial.

3.3. Replicação de conteúdo

Os nós participantes do sistema de tolerância a ataques possuem um espaço de armazenamento que é dedicado a manter cópias de documentos servidos originariamente por outros servidores do sistema. Considerando os custos e a capacidade dos discos atuais, determina-se que o espaço total no sistema destinado à réplica de conteúdo de um sítio é pelo menos igual ao tamanho total desse sítio Web. O espaço para réplica pode ser populado (preenchido) inicialmente segundo uma política fixa de replicação (por exemplo, replicação uniforme), ou pode ser gerenciado como uma *cache*, armazenando réplicas de objetos populares que tenham sido requisitados ao nó recentemente, ou ainda pode ser tratado segundo uma política mista, que inclua *caching* e réplicas distribuídas de forma pró-ativa entre servidores.

No sistema considerado utiliza-se uma política mista. Inicialmente, todos os documentos de todos os servidores possuem a mesma quantidade de réplicas espalhadas aleatoriamente pelos nós do sistema. As réplicas subseqüentes adicionadas a cada servidor são resultados de buscas realizadas e dependerão da política de replicação utilizada. Neste trabalho, como discutido anteriormente, avaliamos o uso de três técnicas de replicação: uniforme, proporcional e *square-root*.

Na replicação uniforme, os documentos são igualmente replicados por todos os nós, desconsiderando diferenças de popularidade entre objetos. Nesse caso, cada objeto de um servidor é copiado um número fixo de vezes para alguns nós do sistema escolhidos aleatoriamente e essa distribuição só é alterada no caso de alteração do conteúdo do servidor. Cada consulta atendida implica na cópia da informação do nó onde o objeto foi encontrado diretamente para o cliente, sem que nenhuma nova cópia seja feita por qualquer nó dentro do sistema.

Na técnica de replicação proporcional, a replicação de um objeto é proporcional à probabilidade do mesmo ser requisitado. Isso resulta na existência de um maior número de cópias para os objetos mais populares. Para se conseguir esse tipo de replicação, quando um objeto é encontrado durante uma busca, uma cópia desse objeto é armazenada no nó do sistema que originou a busca, isto é, o nó que recebeu a requisição HTTP do cliente. Com o passar do tempo, objetos populares surgirão em diversos nós do sistema e poderão mesmo vir a desalojar objetos pouco populares que estavam inicialmente replicados em alguns nós. Desconsiderando-se falhas (por exemplo, durante um ataque DoS), mesmo objetos pouco populares terão garantidamente pelo menos uma cópia no sistema: o objeto original em seu servidor de origem.

Finalmente, na replicação *square-root*, a quantidade de cópias de um objeto é proporcional à raiz quadrada da probabilidade de um objeto ser requisitado. Isso é conseguido replicando-se cada objeto localizado em uma busca, não apenas no nó que inicia a requisição, mas em todos os nós do caminho percorrido pela consulta dentro da rede P2P. Apesar desse procedimento parecer mais agressivo que a replicação proporcional (que é linear em função da popularidade), é provado que o resultado final é uma distribuição de réplicas que obedece à distribuição baseada na raiz quadrada da popularidade [Cohen and Shenker, 2002].

Quando o espaço reservado para *cache* é totalmente preenchido e novas cópias precisam ser feitas, uma política de substituição de cópias é aplicada. No sistema, a política adotada foi a LRU (*Least Recently Used*): cópias com maior tempo sem uso são substituídas por novas réplicas.

4. Metodologia

Para se avaliar a eficácia das três políticas de replicação simulamos uma rede P2P sujeita a uma carga de requisições com características encontradas em servidores WWW usuais. A principal métrica utilizada foi a porcentagem de buscas bem sucedidas observadas pelos clientes para as diversas configurações. Métricas que indiquem a eficiência do sistema em termos do tempo de acesso aos documentos ou o custo de atualização não são consideradas.

O simulador DISCOVERY [Kelaskar et al., 2002, Matossian, 2003], implemen-

tado na linguagem Java, foi utilizado como base para as simulações. Em sua versão original ele implementava apenas uma rede Gnutella com replicação uniforme de objetos. O simulador foi alterado para utilizar topologias de redes baseadas em leis de potência (*power laws*) criadas com o gerador de topologias BRITE [Faloutsos et al., 1999, BRITE, 2003]. Além disso, o simulador foi alterado também para incluir as técnicas de replicação proporcional e *square-root*. As buscas executadas pelo simulador, que inicialmente seguiam uma distribuição uniforme, foram alteradas para seguir uma distribuição de Zipf [Crovella et al., 1996] a fim de garantir que a carga considerada seguisse o perfil de carga normalmente observado em servidores WWW.

As simulações consideram um universo de até 1000 nós participando de uma rede P2P não-estruturada que tem características da rede Gnutella. Esses nós representam os servidores do sistema que colaboram entre si. Por simplicidade, consideramos o conteúdo total de apenas um servidor, que é inicialmente distribuído igualmente por todos os servidores do sistema com cada servidor recebendo um arquivo daquele sítio. Apesar de ser uma simplificação, podemos considerar que a simulação apresenta o comportamento visto pelos clientes de um sítio pertencente ao sistema.

Três conjuntos principais de simulações foram realizados, sendo cada conjunto relacionado à aplicação de uma das técnicas de replicação no sistema de tolerância. A tabela 1 apresenta os parâmetros e valores usados dentro de cada conjunto. Os valores destes parâmetros foram variados a fim de verificar o uso das técnicas de replicação diante de diferentes números de vizinhos, diferentes *fanouts* e diferentes porcentagens de falhas de nós devido a ataques. Os nós que sofrem ataques nas simulações são escolhidos de forma aleatória.

Parâmetros	*Valores*	*Descrição*
Vizinhos (*N*)	10, 50	Número máximo de conexões a partir de cada nó da rede P2P
Fanout (*F*)	2, 5, 10	Número máximo de vizinhos para os quais mensagens de busca serão repassadas
Porcentagem de falhas (*P*)	30 - 80	Porcentagem dos nós escolhidos aleatoriamente para sofrer ataques DoS durante uma simulação

Tabela 1: Parâmetros e valores utilizados nas simulações

Em cada simulação é realizado um número total de 1100 iterações (requisições), sendo 10% destas usadas na inicialização da rede, 45% representam a realização de buscas no sistema durante sua operação com comportamento normal (isto é, sem a ocorrência de ataques DoS) e o restante (45%) é reservado à realização de buscas sobre o sistema após a ocorrência do ataque.

A principal métrica utilizada para medir a eficiência do sistema é a *média das porcentagens de sucesso de buscas* para várias simulações (MPS). Em geral, *porcentagem de sucesso* (PS) indica o número de requisições atendidas com sucesso durante um período da simulação dividido pelo total de requisições efetuadas durante o mesmo período (equação 1).

$$\mathrm{PS} = \frac{\text{buscas realizadas com sucesso}}{\text{total de requisições}} \tag{1}$$

Um período de simulação pode ser o intervalo em que o sistema opera atendendo

às requisições sem sofrer ataques ou o intervalo no qual as requisições ocorrem após a falha de vários nós da rede devido ao ataque DoS. Para cada combinação de valores dos parâmetros foram realizadas 20 simulações e a média da porcentagem de sucessos de buscas foi calculada.

5. Resultados

Inicialmente, os valores dos parâmetros *número de vizinhos* e *fanout* foram avaliados para se verificar a sensibilidade do comportamento da rede à sua variação. A figura 3 apresenta um gráfico com a média da porcentagem de sucesso para as três políticas de replicação de objetos (uniforme, proporcional e *square-root*) em função do número de vizinhos (10 e 50) e do *fanout* (5 e 10) para um caso onde 50% dos nós do sistema foram desabilitados por um ataque DoS. Pode-se observar que o aumento do número de vizinhos de 10 para 50 tem um impacto pequeno no desempenho do sistema em relação à mudança da política de replicação. Isso tem o aspecto positivo de permitir que um nó armazene estado para manutenção da topologia da rede, considerando menos vizinhos, se necessário, sem uma degradação muito grande de desempenho. Além disso, nota-se também que o ganho na porcentagem de sucesso ao aumentar o *fanout* de 5 para 10 vizinhos também é limitado, podendo vir a ser descartado posteriormente caso o aumento do *fanout* venha a gerar um número muito grande de mensagens.

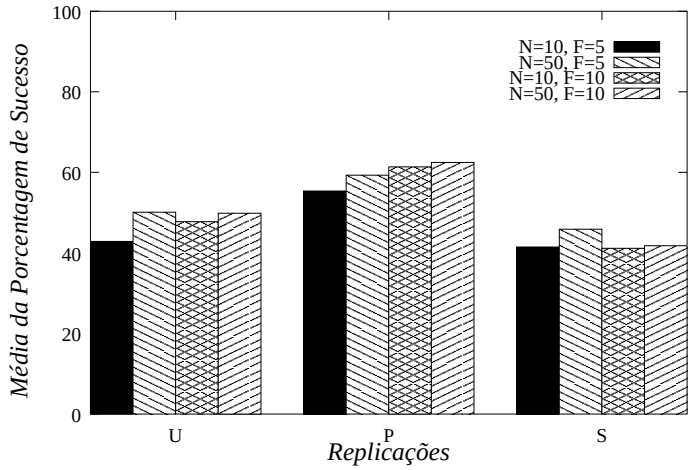

Figura 3: Impacto do número de vizinhos e do *fanout* no sucesso da busca

Para observar melhor a dependência do sistema em relação ao *fanout* foi realizada uma segunda bateria de testes cujos resultados são apresentados na figura 4, que apresenta um gráfico comparando o desempenho do sistema (em termos da porcentagem de sucessos) variando-se o *fanout* entre 2, 5 e 10 para as três políticas de replicação, novamente considerando-se uma porcentagem de nós desativados igual a 50%. Percebe-se claramente que o aumento de *fanout* de 5 para 10 traz ganhos limitados para o sistema.

As figuras 3 e 4 indicam que a técnica de replicação proporcional apresenta melhores resultados em relação que às duas outras técnicas quando 50% dos nós da rede são desabilitados por um ataque DoS. A figura 5 apresenta uma análise mais detalhada do comportamento do sistema considerando-se diferentes porcentagens de falhas de nós na rede (30%, 40%, 50%, 60% ,70% e 80%). Em vista das análise anteriores, essas simulações

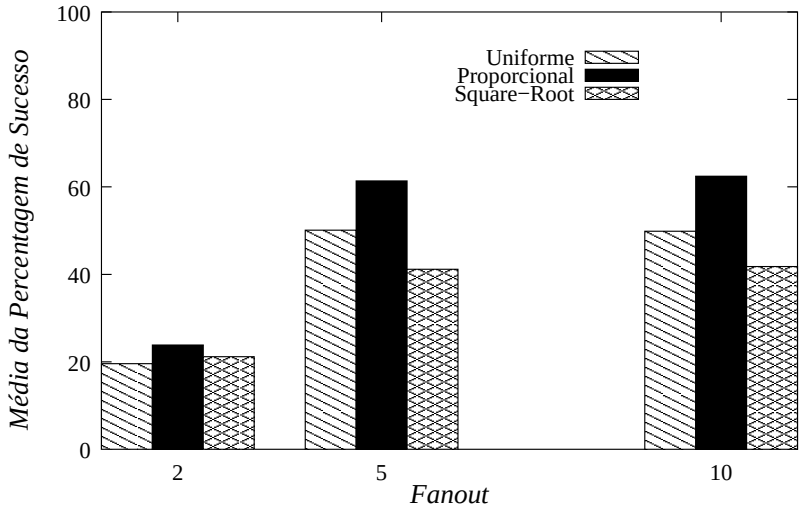

Figura 4: Sucesso de busca em função do *fanout*

foram realizadas com *fanout* e número de vizinhos igual a 10. A figura confirma que a replicação proporcional apresenta melhor desempenho em todas aquelas situações de falha, mantendo um desempenho aproximadamente 10% superior às demais políticas. O desempenho das políticas uniforme e *square-root* se invertem quando a porcentagem de falhas ultrapassa pouco mais de 50%, sendo a distribuição *square-root* mais eficiente em situações mais extremas. Apesar de não possuirmos ainda dados detalhados que permitam uma análise definitiva, isso parece ser devido ao fato de a distribuição *square-root* garantir uma pequena vantagem para os objetos mais populares, que acabam garantindo mais sucessos sob um regime de grande deterioração. Enquanto a rede ainda não apresenta uma degradação muito alta, o maior número de réplicas de objetos menos populares parece beneficiar mais à distribuição uniforme do que o aumento relativamente pequeno da presença de objetos mais populares para a distribuição *square-root*.

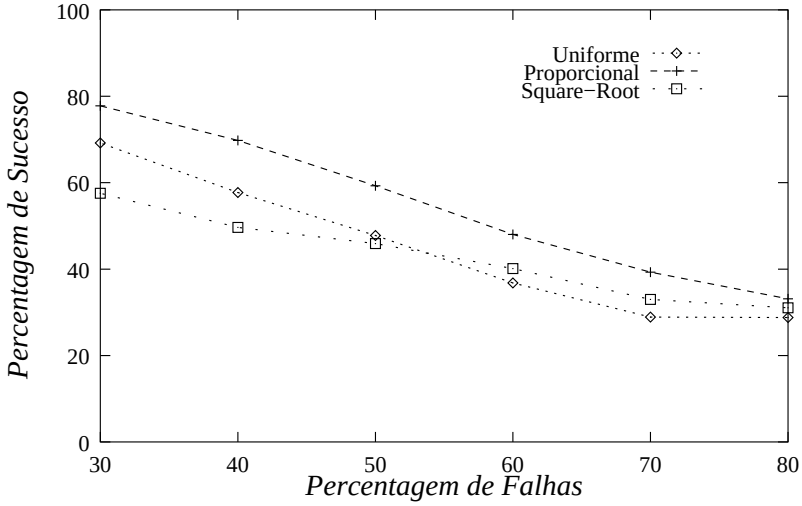

Figura 5: Sucesso de busca para diferentes porcentagem de falhas

É interessante destacar que o resultado apresentado neste trabalho diferencia-se significativamente do resultado apresentado no trabalho de Cohen *et*

al. [Cohen and Shenker, 2002, Cohen et al., 2002]. Aquele trabalho faz um estudo das mesmas três técnicas de replicação em redes P2P não-estruturadas e apresenta como resultado que a técnica de replicação *square-root* apresenta nitidamente melhor desempenho, o que poderia parecer se contrapor ao resultado aqui apresentado. Entretando, aquele trabalho não considera a aplicação das políticas de replicação em presença de um ataque DoS ou outro evento que pudesse causar a queda de um grande número de nós da rede. Nesse caso, nosso trabalho indica que diante de eventos inesperados, como ataques DoS, a política de replicação proporcional apresenta melhores resultados.

6. Conclusões e trabalhos futuros

Este trabalho apresenta uma análise das principais políticas de replicação de objetos em redes *peer-to-peer* face a eventos catastróficos como a queda de uma grande fração dos nós da rede devido a um ataque DoS. Os resultados indicam que sistemas de tolerância a ataques DoS como o descrito neste trabalho devem utilizar uma técnica de replicação proporcional à popularidade dos objetos envolvidos. A análise foi realizada utilizando-se um simulador de redes não-estruturadas do tipo Gnutella estendido para incluir algumas características particulares importantes, como a consideração de uma topologia de rede que siga a leis de potência (*power laws*) e uma distribuição de popularidade para os objetos consultados que corresponda ao observado em servidores WWW.

Outros resultados incluem a observação da pouca variação em termos de desempenho ao se considerar otimizações do protocolo Gnutella como a limitação do número de vizinhos e do número de vizinhos consultados a cada requisição (*fanout*). Apesar de valores maiores para essas grandezas levarem a ganhos positivos, a fração mais significativa desses ganhos se materializa já com valores relativamente pequenos (10) para ambos os casos.

Como trabalhos futuros mencionamos a inclusão de outras variáveis na análise do sistema como, por exemplo, o número de mensagens trocadas (já em andamento), e a simulação simultânea de carga para diversos servidores, bem como a utilização de carga real obtida de *logs* de servidores reais. Uma outra questão que está sendo considerada é a validação desses resultados em uma rede real, com uma implementação desse sistema na plataforma PlanetLab [Planet Lab, 2003].

Referências

BRITE (2003). Boston University Representative Internet Topology generator. [Online] Disponível em: http://www.cs.bu.edu/brite/. Acesso: Novembro 2003.

Cohen, E. et al. (2002). Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th International Conference on Supercomputing*, pages 84–95.

Cohen, E. and Shenker, S. (2002). Replication strategies in unstructured peer-to-peer networks. In *Proceedings of the ACM SIGCOMM'02 Conference*.

Crovella, M. et al. (1996). Characterizing reference locality in the www. In *Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems (PDIS '96)*, page 92.

de Vivo, M. et al. (1999). Internet vulnerabilities related to TCP/IP and T/TCP. *SIGCOMM Computer Communication Review*, 29(1):81–85.

Dean, D., Franklin, M., and Stubblefield, A. (2001). An algebraic approach to IP traceback. In *Proceedings of Network and Distributed System Security Symposium (NDSS)*.

Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the Internet topology. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 251–262.

Gil, T. M. and Poletto, M. (2001). Multops: A data-structure for bandwidth attack detection. In *Proceedings of the 10th USENIX Security Symposium*, pages 23–39.

Graham, R. L. (2001). Peer-to-Peer: Toward a Definition. [Online] Disponível em: http://www.ida.liu.se/conferences/p2p/p2p2001/p2pwhatis.html. Acesso: Outubro.

Jung, J., Krishnamurthy, B., and Rabinovich, M. (2002). Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites. In *Proceedings of the International World Wide Web Conference*, pages 252–262.

Kelaskar, M., Matossian, V., Mehra, P., Paul, D., Vaidhyanathan, A., and Parashar, M. (2002). A study of discovery mechanisms for peer-to-peer applications. In *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid Workshop on Global and Peer-to-Peer on Large Scale Distributed Systems*, pages 444–445.

Kong, K. and Ghosal, D. (1997). Pseudo-serving: A user-responsible paradigm for Internet access. In *Proceedings of the 6th International World Wide Web Conference*.

Kong, K. and Ghosal, D. (1999). Mitigating server-side congestion in the Internet through pseudoserving. *IEEE/ACM Transactions on Networking*, 7(4):530–543.

Krishamurthy, B., Liston, R., and Rabinovich, M. (2003). Dew: DNS-enhanced web for faster content delivery. In *Proceedings of the Twelfth International World Wide Web Conference*.

Limeware (2003). Limeware Internet Website. [Online] Disponível em: http://www.limeware.com. Acesso: Outubro 2003.

Lv, Q., Ratnasamy, S., and Shenker, S. (2002). Can heterogeneity make gnutella scalable? In *Eletronic Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS)*.

Matossian, V. (2003). Javadocs of the P2P Discovery Project. [Online] Disponível em: http://www.caip.rutgers.edu/ vincentm/Discovery/javadocs/. Acesso: Novembro 2003.

Mohiuddin, S. et al. (2002). Defending against a large scale denial-of-service attack. In *Proceedings of the 2002 IEEE*, pages 30–37.

Padmanabhan, V. N. and Sripanidkulchai, K. (2002). The case for cooperative networking. In *Proceedings of the Peer-to-Peer Systems: First International Workshop, (IPTPS) 2002*, pages 178–190.

Peng, T., Leckie, C., and Ramamohanarao, K. (2003). Protection from distributed denial of service attack using history-based IP filtering. In *Proceedings of the IEEE International Conference on Communications (ICC 2003)*.

Planet Lab (2003). Planet Lab. [Online] Disponível em: http://www.planet-lab.org/. Acesso: Dezembro 2003.

Sangpachatanaruk, C. et al. (2003). Design and analysis of a replicated elusive server scheme for mitigating denial of service attacks. Para ser publicado em: Journal of Systems and Software, Elsevier.

Schäfer, G. (2002). Research challenges in security for next generation mobile networks. In *Proceedings of the Workshop on Pioneering Advanced Mobile Privacy and Security (PAMPAS)*.

Stading, T., Maniatis, P., and Baker, M. (2002). Peer-to-peer caching schemes to address flash crowds. In *Proceedings of the 1st International Peer-to-Peer Systems Workshop (IPTPS 2002)*.

Stavrou, A., Rubenstein, D., and Sahu, S. (2002). A lightweight, robust P2P system to handle flash crowds. In *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP'02)*, page 226.

Villela, D. and Rubenstein, D. (2003). A queueing analysis of server sharing collectives for content distribution. In *Proceedings of the Eleventh IEEE International Workshop on Quality of Service (IWQoS 2003)*.

Wang, L., Pai, V., and Peterson, L. (2002). The effectiveness of request redirection on CDN robustness. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*.