Sobre Calotes e Múltiplas Personalidades no BitTorrent

Felipe Pontes, Francisco Brasileiro, Nazareno Andrade

Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Laboratório de Sistemas Distribuídos
Campina Grande, PB, Brasil

{felipe,fubica,nazareno}@dsc.ufcg.edu.br

Resumo. *A geração de identidades e a associação destas às entidades de um enxame (swarm) BitTorrent, um dos sistemas de distribuição de conteúdo mais populares do momento, é feita normalmente de forma autônoma. Além disso, o mecanismo de incentivo do BitTorrent usa uma escolha aleatória na descoberta de novos parceiros. Essas duas características tornam o sistema vulnerável a um ataque sybil realizado por um caloteiro. Neste artigo, nós definimos um modelo analítico que pode ser usado para analisar o impacto que um ataque desse tipo pode ter sobre o mecanismo de incentivo do BitTorrent. Nossa análise inicial utiliza duas métricas para avaliar tal impacto. A primeira é o número de identidades que um caloteiro precisa criar para que o seu tempo de download seja melhor ou igual ao tempo de download dos nós que cooperam com o sistema. A segunda métrica avalia quão mais rápido um caloteiro faz download de um arquivo, em comparação com um nó que coopera com o sistema, à medida que o número de identidades do caloteiro cresce. Nossos resultados mostram que o número de identidades necessárias para ter sucesso em um ataque é pequeno e cresce sub-linearmente com o número de nós no sistema. Por exemplo, para um sistema com cerca de* 50.000 *nós, menos* 100 *identidades diferentes são necessárias para que a utilidade do atacante seja maior que aquela de um colaborador.*

Abstract. *BitTorrent, one of the most popular content distribution protocols nowadays, has an identification generation scheme that is completely autonomous. Furthermore, BitTorrent uses a random mechanism to discover new peers. This leaves the system vulnerable to a sybil attack, by which an entity associates multiple identifications to itself in an attempt to fool the other peers that execute the agreed protocol and increase its utility. In this paper we present an analytical model that can be used to evaluate the impact of such an attack. Our initial analysis uses two metrics. The first one is the number of different identifications that an attacker must have in order to experience download times equal or smaller to that experienced by the other peers that collaborate resources to the system. The other metric assesses how faster an attacker downloads a file, compared to collaborators, as we increase the number of identifications that the attacker has. Our results show that the number of different identities required is small and grows in a sub-linear way with the size of the system. For instance, in a system with around* 50.000 *peers, less than* 100 *identities are necessary to make the utility of the attacker better than that of a collaborating peer.*

1. Introdução

Com o crescimento e aperfeiçoamento da Internet observou-se a possibilidade de se compartilhar recursos entre usuários da rede com interesses afins. Aproveitando essa oportunidade, vários sistemas foram desenvolvidos com base na arquitetura entre-pares (*peer-to-peer*). Nela, usuários compartilham seus recursos diretamente com seus pares. Entretanto, tais sistemas devem possuir mecanismos de incentivo à colaboração, como forma de evitar que uma porção significativa dos usuários sejam "caronas" (*free-riders*) ou "caloteiros"[1], e utilizem os recursos providos no sistema sem colaborar com a comunidade [Adar and Huberman 2000].

Em sistemas entre-pares a associação de identidades a entidades do sistema é normalmente feita de forma autônoma pela própria entidade. Dessa forma, se por um lado elimina-se a necessidade de um componente central que assegure identificações únicas para entidades, por outro lado a geração de novas identidades para uma entidade pode ser feita com um custo muito baixo, o que torna o sistema vulnerável a um ataque *sybil*[2] [Douceur 2002]. Em um ataque *sybil* uma entidade maliciosa cria identidades suficientes para constituir uma grande fração da comunidade e enganar o sistema. De modo geral, sistemas que possuem um controle fraco do esquema de criação e associação de identidades são suscetíveis a ataques *sybil*.

Ataques *sybil* podem se materializar de diferentes formas. Por exemplo, em um sistema de votação *online* suscetível a esses ataques, um usuário malicioso com mais de uma identificação pode votar mais de uma vez e comprometer o resultado final da votação. Um ataque *sybil* também pode ser utilizado por um usuário que queira aumentar sua pontuação em sistemas de classificação de páginas Web [Page et al. 1998]. Nesse caso, um atacante pode criar várias páginas e utilizá-las para recomendar uma outra página. Um exemplo mais conhecido do ataque é a distribuição indiscriminada de mensagens eletrônicas (*spam*). Nessa modalidade do ataque, um atacante cria várias contas de correio eletrônico, na maioria das vezes inativas, para enviar mensagens indesejadas, dificultando a ação de filtros que usem o endereço do remetente como parâmetro de filtragem. Por fim, o ataque pode também ser realizado em sistemas entre-pares de compartilhamento de recursos, nos quais um atacante tenta contornar os mecanismos de incentivo à colaboração e conseguir tanta utilidade quanto possível, sem contribuir com quaisquer recursos para o sistema, ou seja, dando um "calote" nos seus pares. Em particular, alguns sistemas possuem características aleatórias, as quais podem ser utilizadas para realizar desempates entre entidades que estejam em igualdade de condições e decidir quem terá preferência na utilização de recursos. Isso potencializa o ataque, criando cenários ideais para o atacante.

A geração de identidades no BitTorrent [Cohen 2003], um dos sistemas de distribuição de conteúdo mais populares do momento, é feita normalmente de forma autônoma. Além disso, o mecanismo de incentivo do BitTorrent usa uma escolha aleatória na descoberta de novos parceiros, como será detalhada na Seção 2. Essas duas características tornam o sistema vulnerável a um ataque *sybil* realizado por um caloteiro. Neste artigo, nós definimos um modelo analítico que pode ser usado para analisar o impacto que um ataque desse tipo pode ter sobre o mecanismo de incentivo do BitTorrent. Em

[1] Neste artigo nós daremos preferência ao termo caloteiro, uma vez que o comportamento que vamos avaliar é executado por um usuário que tem intenção explícita de burlar o protocolo; esse não é necessariamente o comportamento de um "carona".

[2] Esse termo é uma referência à personagem do livro homônimo, o qual relata o estudo de um distúrbio psicológico de uma paciente que apresentava múltiplas personalidades [Schreiber 1973].

particular, nós queremos avaliar qual é o número mínimo de identidades que precisam ser geradas por um nó caloteiro de forma que o tempo de download dele seja menor ou igual ao tempo de download dos outros nós que cooperam com o sistema.

O restante do artigo está organizado da seguinte forma. A próxima seção descreve o funcionamento básico do BitTorrent, abordando seu mecanismo de incentivo à colaboração e como um atacante poderia explorar características do mecanismo para não precisar contribuir com o sistema. Na Seção 3 é apresentado um modelo analítico para representar o ataque de um nó caloteiro com múltiplas identidades sobre um enxame do BitTorrent, enquanto que a Seção 4 trás uma análise desse modelo. A Seção 5 discute os principais trabalhos relacionados com a nossa pesquisa. Finalmente, a Seção 6 conclui o artigo apresentando nossas considerações finais e uma indicação dos trabalhos futuros a serem realizados.

2. O Protocolo do BitTorrent

Quando um arquivo é disponibilizado através do protocolo HTTP (*HyperText Transfer Protocol*), todo o custo de distribuição desse arquivo é assumido pela máquina hospedeira. O protocolo BitTorrent permite que grandes volumes de conteúdo digital possam ser distribuídos de forma eficiente usando uma sistema entre-pares. Com o BitTorrent, quando vários nós estão baixando o mesmo arquivo ao mesmo tempo, eles enviam pedaços do arquivo uns para os outros. Isso faz com que o custo da distribuição do arquivo seja compartilhado entre os pares envolvidos na sua distribuição. Esse conjunto de nós participando da distribuição do conteúdo é denominado de "enxame" (*swarm*) [Cohen 2003].

Para um arquivo ser distribuído, um nó que possua uma cópia completa, denominado de "semeador" (*seeder*), cria um arquivo de metadados (identificado pela terminação *.torrent*) e disponibiliza-o para acesso público, geralmente em sítios *Web*. O arquivo *.torrent* possui informações necessárias à distribuição do arquivo. Uma das informações contidas nesse arquivo é o endereço de um ou mais nós, denominados de *trackers*, que implementam um serviço de *rendez-vous*, permitindo que um nó que se junte ao enxame possa encontrar outros pares naquele enxame[3]. Além do endereço do *tracker*, o arquivo *.torrent* também guarda informações relacionadas à divisão do arquivo em pedaços menores (chamados peças).

Um nó que deseja fazer o download do arquivo, denominado de "sugador" (*leecher*), deverá obter o arquivo *.torrent* e se conectar ao *tracker* para obter uma lista de pares que possuem peças daquele arquivo. Com a lista, o nó sugador poderá se conectar diretamente aos seus pares e iniciar o compartilhamento. Nós sugadores, ao concluírem o download do arquivo, podem permanecer no enxame por algum tempo, atuando como novos nós semeadores. Os nós que fazem parte de um enxame contactam o *tracker* de tempos em tempos tentando descobrir outros pares que se juntaram ao enxame. Cada nó tem um número máximo de conexões permitidas (tipicamente 55), das quais uma parte é usada para que o nó tente se conectar com seus pares, enquanto que a outra é usada para permitir que outros nós se conectem com o nó. Se por um lado um nó pode, potencialmente, receber dados (i.e. fazer download) de todas as suas conexões ativas, por outro lado ele utiliza apenas um pequeno número dessas conexões, chamadas de "não

[3]Para tornar o texto mais fluido, sem perda de generalidade, nós vamos assumir a partir desse ponto que existe um único *tracker* por enxame.

sufocadas" (*unchoked*), para fazer upload para os seus pares (tipicamente 5). A Figura 1 ilustra as conexões de um nó em um enxame BitTorrent.

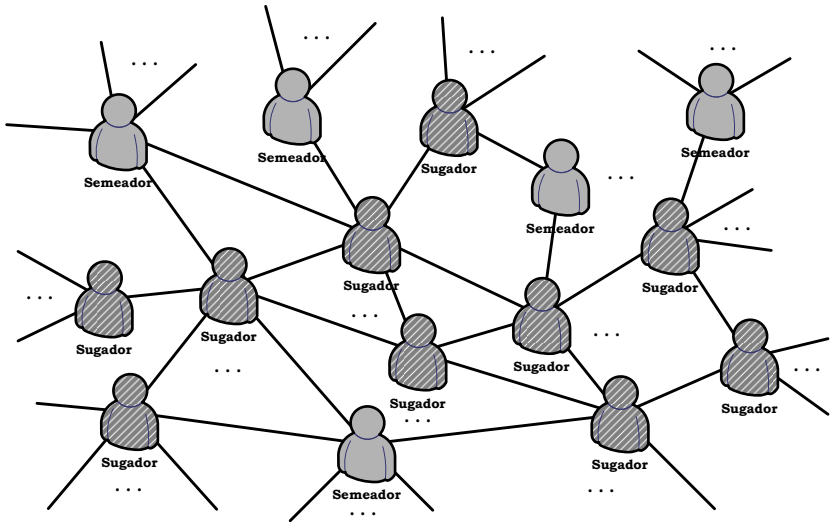

Figura 1. Conexões entre os *pares*.

Apesar do protocolo BitTorrent não evitar que nós caloteiros possam utilizar o sistema, existe um mecanismo para incentivar um nó a colaborar com o enxame. Esse mecanismo é implementado através da forma como um sugador escolhe entre as suas conexões ativas, aquelas que não serão sufocadas. Essa escolha é baseada em uma política *tit-for-tat*, na qual o nó prioriza a realização de upload para os pares que lhe oferecem uma melhor taxa de download. Dessa forma, os nós com maiores taxas de upload (maiores contribuintes) irão ter maiores taxas de download. Portanto, colaborar com o sistema é normalmente a estratégia mais interessante para um nó.

Como um sugador não possui uma visão completa do sistema, o mecanismo de incentivo requer que o nó tente, de tempos em tempos, descobrir se existem conexões que poderiam lhe ofertar melhores taxas de download do que aquelas que ele não está sufocando durante um certo intervalo de tempo (tipicamente esses períodos são de 10 segundos, e a informação utilizada se refere a um intervalo de 30 segundos). Para tal, o sugador sufoca uma parcela das conexões não sufocadas que estão lhe dedicando as menores taxas de upload e escolhe aleatoriamente o mesmo número de conexões sufocadas (*choked*) que passarão a não ser mais sufocadas. Esse mecanismo é chamado de *optimistic unchoking* e, tipicamente, uma única conexão por vez é envolvida nessa operação.

No caso dos nós semeadores, o mecanismo de *optimistic unchoking* é um pouco diferente. Un nó conectado a um semeador não faz upload de conteúdo para este. Desse modo, ao invés de considerar a taxa de upload recebida em suas conexões, um nó semeador considera a taxa de download que os outros nós estão conseguindo dele, priorizando aqueles com maiores taxas. Essa estratégia faz com que semeadores difundam o conteúdo de forma mais eficiente.

Antes de se juntar a um enxame, cada nó é responsável por gerar aleatoriamente sua própria identidade, uma *string* de 20 *bytes*. O esquema de geração de identidades independente e de baixo custo, combinado com a característica aleatória do algoritmo *optimistic unchoking* provê um cenário ideal para um nó caloteiro realizar um ataque

sybil em enxames BitTorrent. Um caloteiro pode inundar a lista de pares distribuída pelo *tracker* com suas identidades, aumentando a probabilidade de aparecer na lista de pares de outros nós que consultem aquele *tracker*. Isso também aumenta a probabilidade de uma das identidades do nó caloteiro ser escolhida aleatoriamente através do algoritmo *optimistic unchoking*. A contribuição deste artigo é avaliar o impacto que tal ataque pode ter no desempenho do sistema.

3. Modelando Ataques *Sybil* no BitTorrent

O modelo apresentado abaixo considera um instante de tempo particular na vida de um enxame quando existem L nós sugadores, S nós semeadores e um nó caloteiro com Σ personalidades (cada uma representada por uma identidade diferente) fazendo parte do enxame. O modelo permite calcular a utilidade de um nó sugador e do caloteiro. A utilidade de um nó em um enxame é representada pela taxa média de download agregada do nó. Esta por sua vez é dada pela soma da taxa de upload que cada um dos pares com os quais ele está conectado lhe dedica. Nossa modelagem do ataque *sybil* parte do princípio de que um ataque é vantajoso quando a taxa de download agregada de um atacante é maior que aquela de um sugador. Caso isso aconteça, é mais vantajoso não contribuir com o sistema.

A taxa de upload que um nó n_i dedica a outro nó n_j com o qual está conectado, em um determinado instante de tempo, depende da taxa máxima de upload de n_i, do papel que n_i está desempenhando (sugador, semeador ou caloteiro), do número de outros pares com os quais n_i está conectado, dos papeis que esses outros pares estão desempenhando, e, dependendo do papel de n_i, da taxa de upload ou de download de seus pares. Sejam c_a, c_u e c_o, respectivamente, o número de conexões ativas que o nó tem, o número de conexões não sufocadas e o número de conexões sujeitas ao mecanismo de *optimistic unchoking*. No que segue, nós fazemos as seguintes hipóteses simplificadoras: i) os semeadores, os sugadores e o caloteiro têm sempre o mesmo valor para c_a, c_u e c_o; ii) $c_a \geq c_u$; iii) todos os nós semeadores têm os mesmos recursos de processamento e de rede e a taxa máxima de upload de cada semeador é U_s^{max}; e, iv) todos os nós sugadores e o nó caloteiro têm os mesmos recursos de processamento e de rede e a taxa máxima de upload de um sugador é U_l^{max}.

O número de semeadores, sugadores e caloteiros conectados a um nó define uma *configuração* para o nó. Dada uma configuração qualquer $\mathcal{C}_n$, sejam $\mathcal{C}_n^l$, $\mathcal{C}_n^s$ e $\mathcal{C}_n^\sigma$, respectivamente, o número de sugadores, semeadores e personalidades do caloteiro conectados ao nó n na configuração $\mathcal{C}_n$. De maneira geral, $0 \leq \mathcal{C}_n^l \leq c_a$, $0 \leq \mathcal{C}_n^s \leq c_a$, $0 \leq \mathcal{C}_n^\sigma \leq c_a$ e $\mathcal{C}_n^l + \mathcal{C}_n^s + \mathcal{C}_n^\sigma = c_a$. Nós assumimos que uma personalidade do nó caloteiro consegue identificar as outras personalidades do nó caloteiro e, dessa forma, nós caloteiros não se conectam uns com os outros, ou seja, quando n é uma personalidade do caloteiro, $\mathcal{C}_n^\sigma = 0$. De forma análoga, os semeadores também conseguem identificar outros semeadores (eles têm todas as peças do arquivo e essa informação é trocada quando dois nós tentam se conectar); assim, quando n é um semeador, $\mathcal{C}_n^s = 0$.

A taxa de upload que um nó sugador n_l dedica a cada um dos nós com os quais está conectado vai depender da configuração de n_l, ou seja, do número de semeadores, sugadores e personalidades do nó caloteiro com os quais esteja conectado. Por exemplo, se n_l está conectado a um outro nó sugador $n_{l'}$ e todos os outros pares de n_l são

semeadores, então toda a banda de upload de n_l é dedicada a $n_{l'}$. Em geral, como será apresentado a seguir, há uma taxa média de upload particular a ser dedicada a cada nó conectado a n_l para cada uma das *configurações* de papeis possíveis para as conexões de n_l.

Vamos primeiro definir como se calcula a fatia da banda de upload $U_{l\leftarrow l,\mathcal{C}_{n_l}}$ que um nó sugador n_l dedica ao conjunto dos outros nós sugadores em uma configuração $\mathcal{C}_{n_l}$. Obviamente, se $\mathcal{C}^l_{n_l} = 0$, trivialmente $U_{l\leftarrow l,\mathcal{C}_{n_l}} = 0$. A seguir consideramos as configurações nas quais n_l está conectado a pelo menos um nó sugador, ou seja, $1 \leq \mathcal{C}^l_{n_l} \leq c_a$. Dado que os nós sugadores são idênticos, qualquer nó sugador conectado a n_l vai receber deste a mesma taxa média de download. Além disso, essas taxas são recebidas a partir das conexões não sufocadas de n_l. Essas, por sua vez são disputadas apenas por sugadores e personalidades do caloteiro. Quando o número de sugadores da configuração é maior ou igual ao número de conexões não sufocadas de n_l ($\mathcal{C}^l_{n_l} \geq c_u$), os $\mathcal{C}^l_{n_l}$ nós sugadores dividem equanimemente todas essas conexões. Porém, devido ao mecanismo de *optimistic unchoking*, algumas dessas conexões (c_o) são divididas também com as personalidades do caloteiro. Desse modo, a fatia de banda de upload que n_l dedica a todos os nós sugadores na configuração $\mathcal{C}_{n_l}$ é dada pela equação 1.

$$U_{l\leftarrow l,\mathcal{C}_{n_l}} = \left(c_u - c_o + \frac{\mathcal{C}^l_{n_l}}{\mathcal{C}^l_{n_l} + \mathcal{C}^\sigma_{n_l}} \cdot c_o\right) \cdot \left(\frac{U^{max}_l}{c_u}\right) \tag{1}$$

Quando o número de sugadores é menor que o número de conexões não sufocadas ($\mathcal{C}^l < c_u$), então cada sugador receberá a banda que n_l dedica a cada uma de suas conexões não sufocadas. Nesse caso, a fatia de banda de upload que n_l dedica a todos os nós sugadores na configuração $\mathcal{C}_{n_l}$ é dada pela equação 2.

$$U_{l\leftarrow l,\mathcal{C}_{n_l}} = \mathcal{C}^l_{n_l} \cdot \frac{U^{max}_l}{\min(c_u, \mathcal{C}^l_{n_l} + \mathcal{C}^\sigma_{n_l})} \tag{2}$$

Por outro lado, a fatia da banda de upload obtida pelas personalidades do nó caloteiro conectado ao nó sugador n_l em uma configuração $\mathcal{C}_{n_l}$, $U_{\sigma\leftarrow l,\mathcal{C}_{n_l}}$, é dada pela diferença entre a taxa máxima de upload de n_l e a fatia da banda de upload que é fornecida para os $\mathcal{C}^l_{n_l}$ sugadores conectados a n_l na configuração $\mathcal{C}_{n_l}$, como indicado na equação 3.

$$U_{\sigma\leftarrow l,\mathcal{C}_{n_l}} = U^{max}_l - U_{l\leftarrow l,\mathcal{C}_{n_l}} \tag{3}$$

Para que a banda oferecida por uma conexão com um sugador seja de fato útil, é preciso que este nó tenha consigo as peças do arquivo que interessam ao nó para o qual está fazendo upload. Qiu e Srikant [Qiu and Srikant 2004] estudaram essa questão e concluíram que a probabilidade disso acontecer é constante e aproximadamente igual a $1 - \left(\frac{\log \mathcal{N}}{\mathcal{N}}\right)$, onde $\mathcal{N}$ é o número de peças nas quais um arquivo foi dividido para distribuição em um enxame.

Note que a taxa de upload obtida por um nó sugador e por uma personalidade do caloteiro em duas configurações distintas que tenham o mesmo número de semeadores, sugadores e personalidades do caloteiro é a mesma. Dessa forma, sejam $\mathcal{C}_{n_l}[1], \mathcal{C}_{n_l}[2], ..., \mathcal{C}_{n_l}[k]$ as k configurações possíveis para um nó sugador n_l do enxame,

tal que para quaisquer duas configurações $\mathcal{C}_{n_l}[i]$, $\mathcal{C}_{n_l}[j]$, $i \neq j$, $\mathcal{C}^l_{n_l}[i] \neq \mathcal{C}^l_{n_l}[j] \vee \mathcal{C}^s_{n_l}[i] \neq \mathcal{C}^s_{n_l}[j] \vee \mathcal{C}^\sigma_{n_l}[i] \neq \mathcal{C}^\sigma_{n_l}[j]$. Se $p(\mathcal{C}_{n_l}[i])$ é a probabilidade da configuração $\mathcal{C}_{n_l}[i]$ acontecer, então, as fatias médias de banda de upload que um nó n_l dedica ao conjunto de sugadores e ao conjunto de personalidades do caloteiro, são dadas, respectivamente, pelas equações 4 e 5.

$$U_{l \leftarrow l} = \left(1 - \frac{\log \mathcal{N}}{\mathcal{N}}\right) \cdot \sum_{i=1}^{k} p(\mathcal{C}_{n_l}[i]) \cdot U_{l \leftarrow l, \mathcal{C}_{n_l}[i]} \tag{4}$$

$$U_{\sigma \leftarrow l} = \left(1 - \frac{\log \mathcal{N}}{\mathcal{N}}\right) \cdot \sum_{i=1}^{k} p(\mathcal{C}_{n_l}[i]) \cdot U_{\sigma \leftarrow l, \mathcal{C}_{n_l}[i]} \tag{5}$$

Como definido anteriormente, $p(\mathcal{C}_{n_l}[i])$ é a probabilidade da configuração $\mathcal{C}_{n_l}[i]$ ocorrer. As conexões que um nó estabelece dependem das identidades que ele obtém do *tracker*. Este, por sua vez, repassa identidades fazendo uma escolha aleatória entre as identidades que ele conhece. Dessa forma, para as configurações $\mathcal{C}_{n_l}[i]$ que podem ocorrer para um nó sugador n_l, $p(\mathcal{C}_{n_l}[i])$ é dada por:

$$p(\mathcal{C}_{n_l}[i]) = \frac{\binom{L-1}{\mathcal{C}^l_{n_l}[i]} \cdot \binom{S}{\mathcal{C}^s_{n_l}[i]} \cdot \binom{\Sigma}{\mathcal{C}^\sigma_{n_l}[i]}}{\binom{L-1+S+\Sigma}{c_a}} \tag{6}$$

Vamos agora definir como se calcula a fatia de upload $U_{l \leftarrow s, \mathcal{C}_{n_s}}$ que um nó semeador n_s dedica a um nó sugador em uma configuração $\mathcal{C}_{n_s}$. O algoritmo de *optimistic unchoking* do semeador leva em consideração as taxas de download dos pares, priorizando aqueles com taxas mais elevadas. Quando o caloteiro usa mais de uma personalidade isso diminui a sua taxa de download com os nós semeadores com os quais está conectado, e dessa forma as personalidades do caloteiro serão preteridas pelos semeadores sempre que houverem sugadores suficientes para ocupar todas as conexões não sufocadas do semeador. Neste caso, uma personalidade do caloteiro só recebe upload de um semeador quando ela é escolhida pelo algoritmo de *optimistic unchoking*. Desse modo, a fatia de upload recebida de n_s pelo conjunto de nós sugadores na configuração $\mathcal{C}_{n_s}$ é dada pela equação 7.

$$U_{l \leftarrow s, \mathcal{C}_{n_s}} = \left(c_u - c_o + \frac{\mathcal{C}^l_{n_s}}{\mathcal{C}^l_{n_s} + \mathcal{C}^\sigma_{n_s}} \cdot c_o\right) \cdot \left(\frac{U^{max}_s}{c_u}\right) \tag{7}$$

Por outro lado, a fatia de upload obtida pelo conjunto de personalidades do nó caloteiro conectado ao nó semeador n_s em uma configuração $\mathcal{C}_{n_s}$, $U_{\sigma \leftarrow s, \mathcal{C}_{n_s}}$, é dada pela diferença entre a taxa máxima de upload de n_s e a fatia de upload que é fornecida para os $\mathcal{C}^l_{n_s}$ sugadores conectados a n_s na configuração $\mathcal{C}_{n_s}$, como explicitado abaixo:

$$U_{\sigma \leftarrow s, \mathcal{C}_{n_s}} = U^{max}_s - U_{l \leftarrow s, \mathcal{C}_{n_s}} \tag{8}$$

Novamente, a fatia de upload obtida de um nó semeador n_s pelo conjunto de nós sugadores e pelo conjunto de personalidades do caloteiro em duas configurações distintas

que tenham o mesmo número de sugadores e personalidades do caloteiro é a mesma. Dessa forma, sejam $\mathcal{C}_{n_s}[1], \mathcal{C}_{n_s}[2], ..., \mathcal{C}_{n_s}[k']$ as k' configurações possíveis para um nó semeador n_s do enxame, tal que para quaisquer duas configurações $\mathcal{C}_{n_s}[i]$, $\mathcal{C}_{n_s}[j]$, $i \neq j$, $\mathcal{C}^l_{n_s}[i] \neq \mathcal{C}^l_{n_s}[j] \wedge \mathcal{C}^s_{n_s}[i] = \mathcal{C}^s_{n_s}[j] = 0 \wedge \mathcal{C}^\sigma_{n_s}[i] \neq \mathcal{C}^\sigma_{n_s}[j]$. Se $p(\mathcal{C}_{n_s}[i])$ é a probabilidade da configuração $\mathcal{C}_{n_s}[i]$ acontecer, então, as fatias médias da banda de upload que o conjunto de nós sugadores e o conjunto de personalidades do nó caloteiro conseguem de n_s, são dadas, respectivamente, pelas equações 9 e 10.

$$U_{l \leftarrow s} = \sum_{i=1}^{k'} p(\mathcal{C}_{n_s}[i]) \cdot U_{l \leftarrow s, \mathcal{C}_{n_s}[i]} \tag{9}$$

$$U_{\sigma \leftarrow s} = \sum_{i=1}^{k'} p(\mathcal{C}_{n_s}[i]) \cdot U_{\sigma \leftarrow s, \mathcal{C}_{n_s}[i]} \tag{10}$$

Por outro lado, a probabilidade de uma configuração $\mathcal{C}_{n_s}[i]$ ocorrer é dada por:

$$p(\mathcal{C}_{n_s}[i]) = \frac{\binom{L}{\mathcal{C}^l_{n_s}[i]} \cdot \binom{\Sigma}{\mathcal{C}^\sigma_{n_s}[i]}}{\binom{L+\Sigma}{c_a}} \tag{11}$$

Finalmente, para calcular a taxa de download agregada média do caloteiro, basta somar a taxa de download agregada que ele consegue em todas as configurações possíveis, ponderando essas taxas pela probabilidade da configuração ocorrer. Ou seja, D_σ é dado pela equação 12.

$$D_\sigma = \sum_{i=1}^{k''} \frac{\binom{L}{\mathcal{C}^l_{n_\sigma}[i]} \cdot \binom{S}{\mathcal{C}^s_{n_\sigma}[i]}}{\binom{L+S}{c_a}} \cdot \left(U_{\sigma \leftarrow l} \cdot \mathcal{C}^l_{n_\sigma}[i] + U_{\sigma \leftarrow s} \cdot \mathcal{C}^s_{n_\sigma}[i] \right) \tag{12}$$

A taxa de download agregada média obtida por um nó sugador qualquer é dado pela diferença entre a banda total de upload disponível sistema e a taxa de download agregada média do caloteiro, como mostra a equação 13.

$$D_l = \frac{U_l * L + U_s * S - D_\sigma}{L} \tag{13}$$

4. Análise

Nesta subseção apresenta-se uma análise de resultados provenientes do modelo introduzido na Seção 3. Nela, nós analisamos o impacto de um ataque *sybil* em sistemas BitTorrent, investigando a sua viabilidade e o benefício que um caloteiro pode conseguir através desse ataque. A viabilidade pode ser expressa em termos do número de identidades necessárias para que um caloteiro consiga um melhor tempo de download que um sugador. Já o benefício é uma medida do quão melhor o tempo de download de um caloteiro pode ser em relação ao de um sugador.

Nossa análise assume sempre os mesmos valores para c_a, c_u e c_o, quais sejam: $c_a = 55$, $c_u = 5$ e $c_o = 1$. Esses são valores comumente utilizados por implementações de

clientes BitTorrent. Também assumimos para todos os resultados apresentados abaixo que $U_l = U_s = 17$. Esses valores para a taxa média de upload são derivados de observações de enxames feitas por Bellissimo *et al.* [Bellissimo et al. 2004].

A Figura 2 mostra o comportamento da quantidade de identidades em função da relação entre o número de sugadores e de semeadores presentes em um enxame com 1.000 nós para diferentes tamanhos de arquivos. Pode-se observar que com exceção do arquivo de tamanho $1Mb$, o qual possui o menor número de peças (cada peça tendo tamanho $256kb$ [Cohen 2003]), todos os outros possuem comportamento semelhante para o número de identidades necessárias. Tal fato nos leva a concluir que a quantidade de identidades necessárias para um caloteiro ter um melhor tempo de download que um sugador independe da relação entre o número de sugadores e de semeadores presentes no enxame, como também, na maior parte dos casos, do tamanho do arquivo sendo disponibilizado. Isso se deriva do fato de que para enxames com um número de peças suficientemente grande, a parcela de contribuição de sugadores se aproximará da parcela de contribuição de semeadores por conta do aumento na probabilidade deles terem uma peça que interesse a outros nós, fazendo com que, do ponto de vista das personalidades dos caloteiros, praticamente não se consiga fazer uma distinção entre eles (veja as equações 1 e 7). O resultado diferente encontrado para o arquivo com $1Mb$ significa que, sendo a probabilidade de um sugador possuir uma peça que interesse a um outro nó menor que nos outros casos, por conta do menor número de peças do arquivo, a diferença entre sugadores e semeadores fica um pouco mais destacada, pois os sugadores irão colaborar menos, fazendo com que quando aumenta-se a relação entre sugadores e semeadores, tenha-se um aumento na quantidade de identidades necessárias a um caloteiro. Em qualquer caso, o número de identidades necessárias para que a utilidade do caloteiro seja maior que a de nós colaboradores em um enxame contendo 1.000 nós é pequeno (pouco mais de 1% do número de nós).

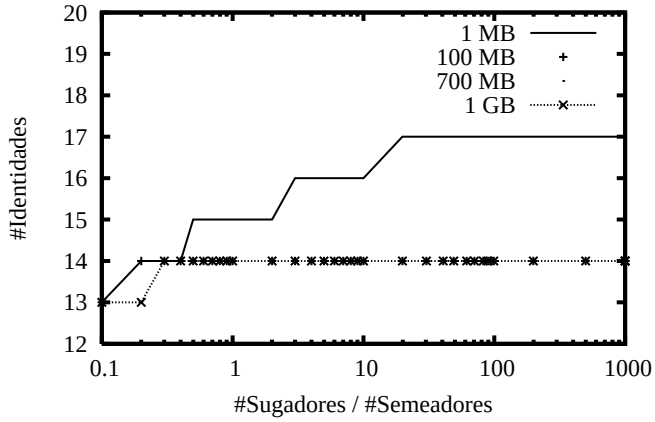

Figura 2. Aumentando a relação entre o número de sugadores e semeadores para arquivos de tamanhos diferentes.

Vamos agora avaliar o impacto do tamanho do enxame no número de identidades necessárias para que a utilidade do caloteiro seja maior do que aquela de um sugador. A Figura 3 mostra o número de identidades requeridas quando aumenta-se a população de um enxame, para um arquivo de $700Mb$ (esse é um tamanho típico, segundo as observações de enxames por Bellissimo), com o número de sugadores igual ao número

de semeadores. O gráfico mostra que a quantidade de identidades também aumenta sublinearmente com o tamanho do enxame, ou seja, quanto menor o enxame, mais barato é para um caloteiro atacá-lo. Isso pode ser explicado pelo fato de que aumentando-se a quantidade de nós em um enxame aumenta-se também a concorrência pelos recursos. No entanto, o número de identidades necessárias é bem pequeno com relação ao tamanho da população de um enxame. Para um enxame com aproximadamente 50.000 nós, são necessárias menos de 100 identidades.

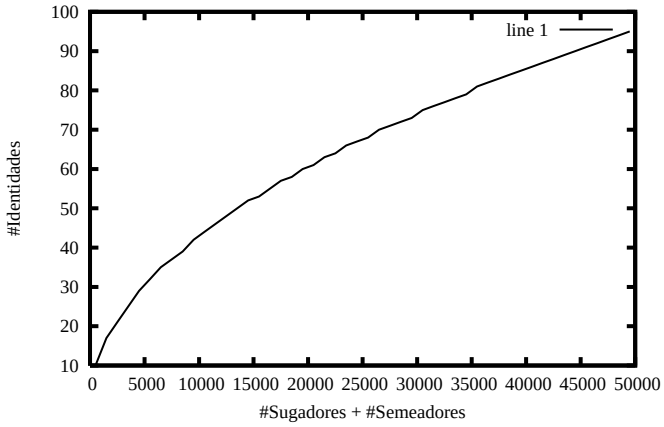

Figura 3. Aumentando a população de um enxame.

Utilizando a mesma quantidade de nós da análise anterior e observando um enxame particular com a mesma quantidade de sugadores e semeadores, já que a relação entre essas quantidades não influencia no número de identidades, a Figura 4 apresenta quão melhor pode ser a utilidade do caloteiro em relação àquela de um sugador. Pode-se observar que o benefício de um caloteiro aumenta bastante rápido à medida que aumenta-se o número de suas identidades.

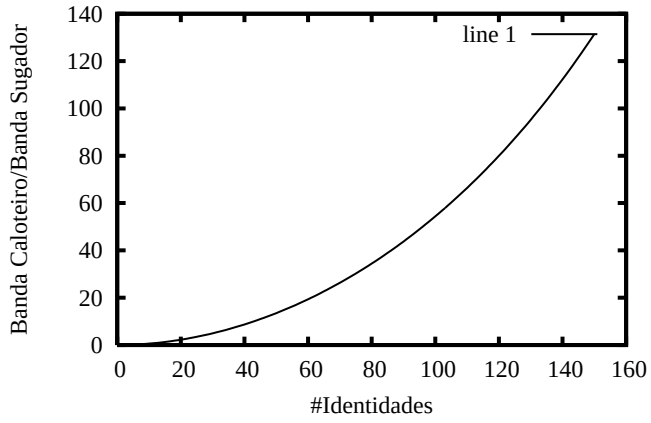

Figura 4. Aumentando o número de personalidades de um caloteiro aumenta-se seu benefício.

5. Trabalhos Relacionados

Alguns trabalhos foram realizados com o intuito de modelar analiticamente sistemas BitTorrent. No entanto, diferentemente deste trabalho (e até onde nós sabemos), nenhum deles visou representar um ataque *sybil*. De maneira geral, foram propostos modelos

para investigar de que maneira a participação de nós semeadores e sugadores no sistema interfere no desempenho do mesmo. Guo *et al.*, com base no modelo de fluído proposto em [Qiu and Srikant 2004], propuseram alterações para melhorar o desempenho de sistemas BitTorrent através da participação simultânea dos nós em múltiplos enxames [Guo et al. 2005]. Já no trabalho de Andrade *et al.*, também com base no modelo de fluído proposto em [Qiu and Srikant 2004], foi analisado como a inserção de uma entidade servidora pode ser útil para melhorar a qualidade de serviço na distribuição do conteúdo em sistemas BitTorrent [Andrade et al. 2007].

Jun e Ahamad afirmaram, com base em experimentos, que o mecanismo de incentivo à colaboração do BitTorrent não pune caloteiros e nem premia os sugadores adequadamente [Jun and Ahamad 2005]. Entretanto, Andrade *e*t al. descreveram aspectos do BitTorrent e experimentos que mostram que o seu uso resulta, de maneira geral, em aumento do comportamento colaborativo [Andrade et al. 2005]. Confirmando esse fato, e contrariando Jun e Ahamad, Legout *et al.* avaliaram também experimentalmente o mecanismo e concluíram que ele é robusto a caloteiros [Legout et al. 2006].

Liogkas *et al* apresentaram estratégias para um nó malicioso conseguir mais vantagem que nós colaboradores no BitTorrent [Liogkas et al. 2006]. Uma estratégia para burlar o sistema seria a de só fazer download de semeadores. Quando um nó se conecta a outro no BitTorrent eles enviam a informação um para o outro de quais peças do arquivo possuem. Um semeador pode ser identificado porque ele anuncia que possui todas as peças. Outra estratégia seria a de só fazer download dos nós mais rápidos no sistema. Quando um nó recebe uma peça do arquivo ele anuncia para todos os outros nós. Observando as mensagens de recebimento das peças pode-se calcular as taxas de download dos outros nós, determinando quem são os mais rápidos. Por fim, uma outra estratégia é a de um nó malicioso mentir ao anunciar quais pedaços do arquivo possui. Dessa maneira mais nós ficarão interessados nos pedaços do arquivo que ele possui e ele poderá se conectar a mais nós aumentando sua taxa de download. Não é feita uma análise do impacto de um ataque *sybil* como fazemos neste artigo.

Recentemente, Konrath *et al.* analisaram duas estratégias para um ataque *sybil* tentar destruir um enxame BitTorrent [Konrath et al. 2007]. A primeira delas consiste em utilizar um grande número de identidades para burlar o protocolo de escolha de peça a ser requisitada. Mentindo ao anunciar quais peças possui, um atacante pode artificialmente tornar uma peça mais ou menos rara, fazendo com que as peças que são realmente raras tendam a desaparecer do enxame e impedindo que os outros nós completem o download com sucesso. A segunda estratégia utilizada no trabalho consiste em inundar o enxame com personalidades de caloteiros para que os nós se conectem predominantemente com eles e não consigam baixar peças do arquivo. Resultados de simulações mostraram que a partir de um determinado valor, quanto maior o número de atacantes com relação ao número de nós colaboradores, maior é a taxa de falha nos downloads. Diferente do explorado por *Konrath et al*, este trabalho investiga o impacto de um ataque *sybil* em sistemas BitTorrent quando o atacante não está interessado em destruir o sistema, mas sim em ganhar mais utilidade que um nó colaborador.

6. Conclusão

A partir do exposto neste artigo podemos concluir que um ataque *sybil* pode ser prejudicial a sistemas BitTorrent mesmo com a geração de poucas personalidades. Tal ataque

independe da relação entre o número de sugadores e de semeadores em um enxame. Por outro lado, quanto maior a população do enxame, maior o número de personalidades necessárias para que um caloteiro consiga um melhor tempo de download que um sugador. Isso nos leva a concluir que é melhor para um caloteiro esperar o momento em que o enxame não está mais tão concorrido para atacá-lo. Além disso, o número de personalidades necessárias para ter sucesso em um ataque cresce de maneira sub-linear com o tamanho do enxame. Precisa-se de menos que 100 personalidades para atacar um enxame com aproximadamente 50.000 nós, por exemplo.

O benefício de um caloteiro quando da realização de um ataque *sybil* em um sistema BitTorrent cresce muito rapidamente quando aumenta-se o número de suas personalidades. Isso confirma a suposição inicial de que quanto mais personalidades um caloteiro possuir mais fácil ele conseguirá um melhor tempo de download que um sugador.

Apesar do resultado favorável encontrado para um caloteiro, na prática, um ataque pode não ser tão bem sucedido, pois o grande número de conexões que devem ser estabelecidas entre o caloteiro e os outros pares em um enxame pode reduzir o desempenho do cliente fazendo com que ele não consiga uma boa taxa de download. Em uma máquina de um usuário doméstico limita-se em média a 8.000 a quantidade de conexões que podem ser realizadas (esse é o limite comum do número de máximo de arquivos que podem ser abertos ao mesmo tempo). Se cada personalidade de um caloteiro estabelecer 55 conexões, em média o número máximo de personalidades que um caloteiro conseguirá produzir será de aproximadamente 150.

Além disso, os resultados apresentados são válidos apenas se considerarmos as hipóteses simplificadoras introduzidas na Seção 3. Como trabalhos futuros nós vamos realizar um maior detalhamento do estudo através da realização de simulações e de experimentos em enxames. Pretendemos nas simulações tentar confirmar as suposições levantadas no modelo e nos experimentos confirmar na prática os resultados do modelo e de simulações, podendo analisar o desempenho do cliente quando do estabelecimento de um grande número de conexões.

Agradecimentos

Agradecemos a Miranda Mowbray pelas discussões que tivemos durante o desenvolvimento desse trabalho. Este trabalho foi desenvolvido em colaboração com a HP Brasil P&D.

Referências

Adar, E. and Huberman, B. A. (2000). Free riding on gnutella. *First Monday*.

Andrade, N., Mowbray, M., Lima, A., Wagner, G., and Ripeanu, M. (2005). Influences on cooperation in bittorrent communities. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 111–115, New York, NY, USA. ACM Press.

Andrade, N., Santana, J., Brasileiro, F., and Cirne, W. (2007). On the Efficiency and Cost of Introducing QoS in BitTorrent. In *Seventh International Workshop on Global and Peer-to-Peer Computing - GP2PC*.

Bellissimo, A., Shenoy, P., and Levine, B. N. (2004). Exploring the Use of BitTorrent as the Basis for a Large Trace Repository. Technical report, University of Massachusetts.

Cohen, B. (2003). Incentives Build Robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA.

Douceur, J. R. (2002). The Sybil attack. *Lect. Note. Comput. Sci.*, 2429:251–260.

Guo, L., Chen, S., Xiao, Z., Tan, E., Ding, X., and Zhang, X. (2005). Analyzing Torrent Evolution and Performance of BitTorrent-like File Sharing Systems. Technical report, Ohio State University, George Mason University and AT&T Labs-Research.

Jun, S. and Ahamad, M. (2005). Incentives in bittorrent induce free riding. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 116–121, New York, NY, USA. ACM Press.

Konrath, M. A., Barcellos, M. P., Silva, J. F., Gaspary, L. P., and Dreher, R. (2007). Atacando um enxame com um bando de mentirosos: vulnerabilidades em BitTorrent. In *Anais do 25º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'07)*, Belém, Pará, Brasil. Sociedade Brasileira de Computação.

Legout, A., Urvoy-Keller, G., and Michiardi, P. (2006). Rarest first and choke algorithms are enough. In *IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement*, pages 203–216, New York, NY, USA. ACM Press.

Liogkas, N., Nelson, R., Kohler, E., and Zhang, L. (2006). Exploiting bittorrent for fun (but not profit). In *5th International Workshop on Peer-to-Peer Systems (IPTPS 2006)*.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web. In *7th International World Wide Web Conference*, pages 161–172.

Qiu, D. and Srikant, R. (2004). Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 367–378, New York, NY, USA. ACM Press.

Schreiber, F. R. (1973). *Sybil*. Warner Books.