UNIVERSIDADE DO VALE DO RIO DOS SINOS
CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO EM
COMPUTAÇÃO APLICADA

Métodos para Contenção de Poluição em Redes P2P

por

JULIANO FREITAS DA SILVA

Dissertação submetida a avaliação como requisito parcial para a obtenção do grau de Mestre em Computação Aplicada

Orientador: Prof. Dr. Marinho Pilla Barcellos

São Leopoldo, Janeiro de 2007

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

da Silva, Juliano Freitas

Métodos para Contenção de Poluição em Redes P2P / por Juliano Freitas da Silva. — São Leopoldo: Ciências Exatas e Tecnologicas da UNISINOS, 2007.

71 f.: il.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos. Ciências Exatas e Tecnológicas Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, São Leopoldo, BR–RS, 2007. Orientador: Barcellos, Marinho Pilla.

1. Poluição. 2. Contenção. 3. Peer-to-Peer. 4. Segurança. I. Barcellos, Marinho Pilla. II. Título.

À minha esposa
A meus pais

Agradecimentos

A Deus pelo dom da Vida.
À Fabi por seu amor incondicional.
A meus pais por seu zelo e entendimento.
A meus irmãos Cinara e Roger pela amizade.
Ao Marinho pela sua dedicação, exemplo e ensinamentos.
Ao Luciano Gaspary por toda a ajuda e incentivo.
Ao Marlom pela parceria de sempre.
Aos bolsistas Rodolfo, Guilherme, Lucas e Rafael pela ajuda.
Ao Letieri pelo suporte matemático.
Aos colegas de mestrado pelo companheirismo.
Aos amigos, colegas de trabalho e familiares pelo apoio e compreensão.

Resumo

Apesar de ser uma das principais aplicações da Internet na atualidade, o compartilhamento de arquivos P2P tem sido fortemente prejudicado por ataques de poluição de conteúdo. Esta dissertação propõe e analisa uma classe de métodos de contenção de poluição cujo princípio básico é a limitação do número instantâneo de downloads de acordo com a reputação de versões. Inicialmente, o método é proposto e avaliado em termos de um ambiente idealizado, mostrando sua eficiência na contenção de poluição e baixa sobrecarga induzida quando o título não é poluído. A seguir, valendo-se de modelos clássicos para projeto de redes P2P, são propostos e comparados métodos de contenção distribuída.

Palavras-chave: Poluição, Contenção, Peer-to-Peer, Segurança.

TITLE: "CONTENTION POLLUTION METHODS IN P2P NETWORKS"

Abstract

Despite currently one of the main Internet applications, P2P file sharing has been hampered by content pollution attacks. This work proposes and analyzes a class of contention methods to reduce the dissemination of polluted content whose basic principle is to limit the amount of instantaneous downloads according to its reputation. The method is firstly proposed and evaluated in terms of an idealized environment. The evaluation shows the efficiency of the contention method and the low overhead induced when the content is not polluted. Then, inspired by classic P2P designs, we propose and compare distributed contention methods.

Keywords: Pollution, Contention, P2P, Security.

Sumário

Lista de Abreviaturas

CAN Content Addressable Network

CPU Central Processing Unit

DHT Distributed Hash Table

DoS Denial of Service

ESM End System Multicast

GCED Global, Centralizada, Estruturada, Dependente

GDNA Global, Descentralizada, Não-estruturada, Autônoma

IP Internet Protocol

MB Megabyte

MSN Microsoft Network

NAT Network Address Translation

P2P Peer-to-Peer

RTT Round Trip Time

SCED Segmentada, Centralizada, Estruturada, Dependente

SCND Segmentada, Centralizada, Não-estruturada, Dependente

Lista de Figuras

Capítulo 1

Introdução

Nos últimos anos, sistemas Peer-to-Peer (P2P) têm obtido grande notoriedade, sendo o compartilhamento de arquivos a aplicação de maior visibilidade. Segundo [Kumar et al., 2006], estima-se que mais de 8 milhões de usuários estejam conectados simultaneamente às redes P2P FastTrack/Kazaa, Gnutella, eDonkey e eMule. Em 2005, 80% do tráfego em backbones IP referia-se a aplicações P2P [Barbera et al., 2005]. Em uma rede P2P, conteúdo é disponibilizado na forma de *títulos*, que podem ser músicas, vídeos, documentos ou programas. Podem existir diferentes *versões* de um mesmo título, sendo elas diferenciadas por um identificador único, normalmente gerado com base no *hash* do seu conteúdo. Pelo processo natural de download, são disseminadas *cópias* de uma mesma versão em diversos pares da rede P2P.

Poluição de conteúdo (também denominada "envenenamento" [Christin et al., 2005]) é atualmente uma das maiores ameaças a sistemas P2P de compartilhamento de arquivos. O ataque tem como objetivo impedir a obtenção de determinado título, a partir da disponibilização massiva de versões incorretas na rede P2P. Usuários, incapazes de diferenciar as versões íntegras das comprometidas, efetuam download das versões poluídas. Conforme demonstrado por [Liang et al., 2005a], o ataque tem sido realizado em larga escala. Naquele estudo foi demonstrado que mais de 50% dos arquivos populares de música na rede FastTrack/Kazaa se encontravam poluídos, atingindo 80% em determinados títulos. Dentro da perspectiva de que usuários efetuam repetidos downloads até encontrar a versão correta, poluição impacta negativamente a produtividade dos usuários, bem como a carga na Internet.

O ataque de poluição é somente possível devido às características das redes P2P atuais, tais como autonomia de pares, identidades fracas, baixo acoplamento, ausência de controle de admissão e anonimato. Uma série de benefícios é, entretanto,

resultante dessas características, como ausência de ponto central de falhas e de ataque devido a descentralização e baixo acoplamento dessas redes. Propostas de solução ao problema apresentado, portanto, devem implicar minimamente a modificação de tais características – um grande desafio a ser tratado, que tem motivado fortemente o interesse científico.

Este trabalho propõe e analisa um método de contenção de poluição baseado na determinação de limites ao número instantâneo de downloads. O limite é estabelecido de acordo com a reputação da versão, dinamicamente obtida a partir de votos emitidos pelos participantes após a avaliação do conteúdo. A taxa permitida de download a uma versão é diretamente proporcional à sua reputação. Inicialmente, o método é proposto em um ambiente ideal, centralizado, e então estendido para um ambiente distribuído, onde diferentes alternativas são descritas e analisadas. Para demonstrar a efetividade do método proposto, avalia-se a penalidade inserida na disseminação de versões corretas, bem como a eficiência do método em reduzir a disseminação de versões poluídas. Sabe-se, entretanto, que contenção de poluição é um problema complexo e desafiador; o presente trabalho representa um passo na busca de contramedidas eficazes para o seu tratamento. A contribuição deste trabalho é significativa ao delinear um esboço de solução, comparar alternativas, estabelecer os requisitos de projeto e critérios de avaliação.

Para atingir aos objetivos citados, inicialmente são revisados os fundamentos sobre redes P2P e segurança no Capítulo 2. O Capítulo 3 define o problema específico a ser tratado, aprofundando a discussão sobre ataques de poluição em redes P2P e discutindo as principais limitações de sistemas de reputação no combate à poluição. O Capítulo 4 apresenta a modelagem e a avaliação do método de contenção proposto em um ambiente ideal. O Capítulo 5 trata do problema em ambientes distribuídos, identificando aspectos relevantes, apresentando modelos básicos de solução e estabelecendo uma comparação entre os mesmos. O Capítulo 6 finaliza o trabalho, apresentando conclusões e trabalhos futuros.

Capítulo 2

Sistemas Peer-to-Peer e Desafios de Segurança

Nos últimos anos, sistemas Peer-to-Peer (P2P) têm despertado o interesse do mundo acadêmico e da indústria, principalmente pelo seu potencial em oferecer o substrato necessário à criação de compartilhamento de dados em larga escala, distribuição de conteúdo e *multicast* em nível de aplicação [Lua et al., 2005]. Sistemas P2P referem-se a aplicações onde os nodos (pares) efetuam o papel de cliente e servidor. Essas arquiteturas são, em geral, caracterizadas pelo compartilhamento direto de recursos computacionais entre os participantes. Isso possibilita que as aplicações possam beneficiar-se diretamente de características como escalabilidade e robustez inerentes às redes P2P. Junto a essas possibilidades, entretanto, são também impostos novos desafios. Em especial, em um ambiente tão heterogêneo, questões referentes à segurança devem ser cuidadosamente consideradas. Para a ampla disseminação de sistemas P2P nas mais diversas áreas, é necessário que a segurança dos participantes, bem como das instituições que os hospedam, seja garantida.

Neste capítulo, primeiro é apresentado um embasamento sobre sistemas P2P, visando esclarecer a terminologia adotada. Em seguida, serão abordados os principais aspectos de segurança, possíveis ataques e possibilidades de solução. Os aspectos de segurança mais relevantes a este trabalho serão posteriormente aprofundados no Capítulo 3.

2.1 Características Principais de Sistemas Peer-to-Peer

Não existe consenso na literatura sobre o conceito exato de sistemas P2P. A definição inicial refere-se a sistemas totalmente distribuídos nos quais os pares apresentam as mesmas características e funcionalidades. Essa direção, entretanto, exclui vários sistemas atuais aceitos como P2P. Segundo [Theotokis and Spinellis, 2004] as duas características primordiais de sistemas P2P são: (a) compartilhamento direto de recursos computacionais entre os participantes, sem a necessidade de servidores centralizados (apesar do uso de entidades centrais ser admitido em tarefas específicas e pontuais); (b) capacidade de auto-organização em função de conectividade variável e população transiente de pares, adquirindo resistência a falhas de enlaces e de computadores.

Outro fato a ser mencionado é que redes P2P baseiam-se na colaboração, em princípio voluntária, de seus participantes. Com base nessas características em [Barcellos and Gaspary, 2006] é estabelecida a seguinte definição de sistemas P2P:

> "*Redes Peer-to-Peer (P2P) são sistemas distribuídos consistindo de pares interconectados capazes de se auto-organizar em redes de sobreposição (*overlay*s) com o objetivo de compartilhar recursos tais como conteúdo (música, vídeos, documentos, etc.), ciclos de CPU, armazenamento e largura de banda, capazes de se adaptar a populações transientes de pares enquanto mantendo conectividade aceitável e desempenho, sem necessitar da intermediação ou apoio de uma entidade central*".

Objetivando estabelecer a terminologia a ser utilizada nesse estudo, assume-se que uma rede de sobreposição P2P é composta por pares conectados entre si em nível de aplicação. Cada par possui um identificador único na rede P2P. Pares podem armazenar e fornecer objetos a outros que realizam as requisições. Considerando a operação correta do protocolo, pares cooperam na execução de buscas por objetos, download de conteúdo e/ou fornecimento de serviços a outros pares. Chaves identificam unicamente um objeto ou serviço na rede P2P.

2.2 Classificação de Aplicações Peer-to-Peer

Sistemas P2P têm sido empregados em diversas categorias de aplicações. Acredita-se, inclusive, que outras aplicações também poderiam beneficiar-se dessa

tecnologia. Abaixo são apresentadas as principais categorias de aplicações P2P existentes.

- **Compartilhamento de arquivos (*file sharing*)**. São as aplicações P2P mais populares, também conhecidas como distribuição de conteúdo. Seu funcionamento se baseia na publicação de arquivos cujo conteúdo permanece imutável, sendo disseminado para quaisquer usuários participantes da rede P2P. Normalmente, não há restrições quanto à distribuição ou leitura do conteúdo disponibilizado. São exemplos, o Napster [Napster, 2007], Gnutella [Gnutella, 2007], Kazaa [KaZaA, 2007] e BitTorrent [BitTorrent, 2007].

- **Armazenamento de Arquivos em Rede**. A principal diferença dessa categoria em relação à anterior é o fato de que os arquivos em sistemas de armazenamento podem ser modificados. Dessa forma, as alterações devem considerar a replicação e serem propagadas para todas as cópias existentes. Restrições de acesso podem ser consideradas nestes sistemas. Como exemplo, são citados PAST [Druschel and Rowstron, 2001], OceanStore [OceanStore, 2007], Ivy [Chen et al., 2002] e JetFile [JetFile, 2007].

- **Transmissão de Dados**. Também conhecidos como *overlay musticast*, formam uma infra-estrutura de comunicação baseada em *multicast* em nível de aplicação. Possibilitam que conteúdo seja distribuído a um número potencialmente grande de pares dispersos geograficamente. Normalmente essa tecnologia é empregada para a transmissão de eventos ao vivo (*Live Streaming*). Pode-se citar como exemplo o ESM [ESM, 2007].

- **Computação Distribuída**. Aplicações dessa categoria visam prover processamento intensivo, normalmente criando infra-estruturas de grade. Em determinados sistemas pode, inclusive, haver controle centralizado, no formato mestre-escravo. São exemplos o OurGrid [Andrade et al., 2003], SETI@Home [Seti@Home, 2007] e Genome@Home [Genome@home, 2007].

- **Colaboração e Comunicação**. Provêem o suporte necessário para comunicação direta e, normalmente, em tempo real entre os participantes da rede P2P. As aplicações existentes possibilitam comunicação através de voz, mensagens de texto, vídeo e transmissão direta de arquivos. São exemplos, o Skype [Skype, 2007], ICQ [ICQ, 2007], Jabber [Jabber, 2007], MSN Messenger [MSN, 2007] e Google Talk [GoogleTalk, 2007].

2.3 Organização da Rede de Sobreposição

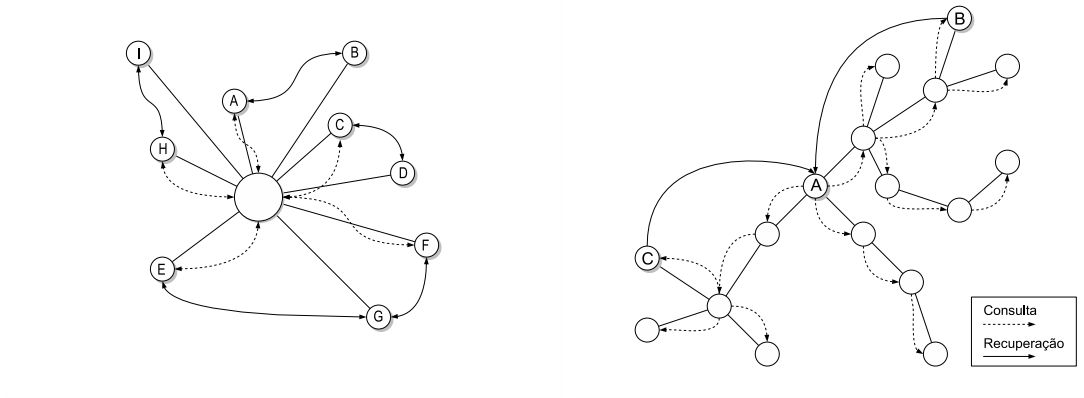
Redes de sobreposição P2P podem ser organizadas sob duas modalidades principais: não estruturadas ou estruturadas. A organização define o formato de operação, influenciando diretamente diversos aspectos como segurança, robustez e desempenho. Abaixo são apresentadas as principais características de cada categoria.

- **Redes não estruturadas**. Nos sistemas com rede não estruturada, a topologia é determinada de forma *ad hoc*. A organização é criada de maneira aleatória, resultante da entrada e saída de pares do sistema; portanto, não existe qualquer regra para o posicionamento de pares, objetos e/ou serviços na rede.

- **Redes estruturadas**. A topologia é definida por regras de alocação de chaves a pares, associando um objeto ou serviço a um par (ou conjunto de pares) de forma determinística e conhecida globalmente na rede P2P. Em boa parte dos casos são utilizadas tabelas *hash* distribuídas ou DHTs (*Distributed Hash Tables*).

Na literatura essa terminologia refere-se à principal classificação P2P: sistemas P2P estruturados e não estruturados. Uma dificuldade inicial associada a sistemas P2P não estruturados refere-se à localização de objetos. Os primeiros métodos de busca realizavam a inundação da rede P2P, claramente ineficiente em termos de número de mensagens, tráfego e tempo para a localização de objetos. Isso motivou a pesquisa e desenvolvimento de novas formas de busca em sistemas não estruturados, com a criação de métodos como caminhada aleatória [Gkantsidis et al., 2004] e índices de roteamento [Tsoumakos and Roussopoulos, 2003]. Em contrapartida, sistemas não estruturados adaptam-se diretamente ao modelo de operação P2P, sem necessitar qualquer reorganização em função de entrada e saída de pares, potencialmente alta nessas redes. Como exemplo de sistemas P2P não estruturados pode-se citar Gnutella, Napster, BitTorrent (ilustrados na figura 2.1), FastTrack/Kazaa, BitTorrent, eDonkey 2000 e Freenet.

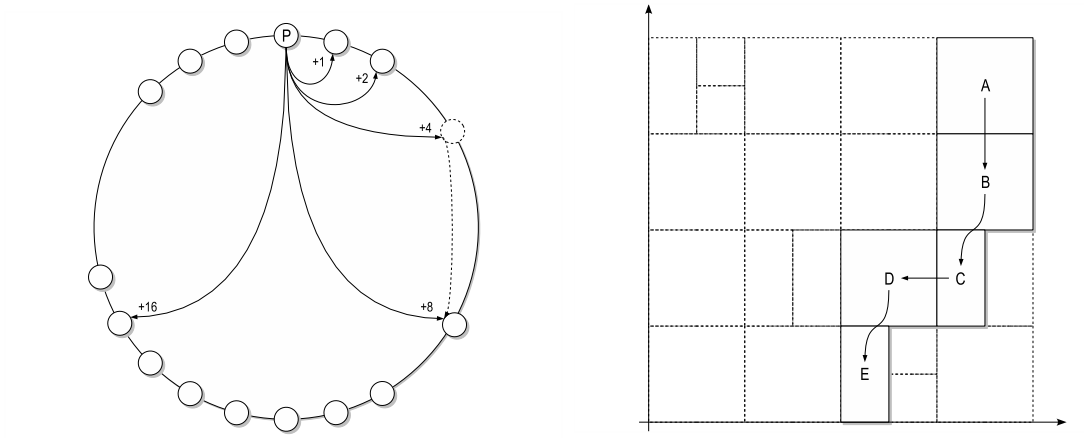
Sistemas P2P estruturados, tipicamente baseados em DHT, possibilitam que os objetos sejam encontrados em um pequeno número de passos. Entretanto, é necessária uma correspondência perfeita entre o termo solicitado na busca e a chave do objeto. Com isso, o par requisitante necessita conhecer antecipadamente a chave do objeto procurado. Alguns autores argumentam que a organização da rede com a saída e entrada de pares é difícil de ser mantida. São exemplos: Chord



(a) Napster (b) Gnutella

FIGURA 2.1 – Exemplos de redes P2P não estruturadas

[Stoica et al., 2003], CAN [Ratnasamy et al., 2001], Tapestry [Zhao et al., 2004], Pastry [Bjurefors et al., 2004] e Kademlia [Maymounkov and Mazieres, 2002]. Os sistemas P2P Chord e CAN são ilustrados na figura 2.2.



(a) Chord (b) CAN

FIGURA 2.2 – Exemplos de redes P2P estruturadas

2.4 Segurança em Sistemas Peer-to-Peer

Esta seção aborda os principais requisitos de segurança de sistemas P2P, bem como os métodos existentes para o seu tratamento. Questões gerais sobre segurança em sistemas distribuídos e redes de computadores não serão diretamente tratadas, em função de já existir vasta literatura a respeito, além de não ser o foco específico

desse estudo. Nesse contexto, serão discutidos seguintes aspectos de segurança pertinentes a sistemas P2P:

- **disponibilidade**: garantia de utilização de determinado serviço, quando necessário;
- **autenticidade**: assegura que alguém (ou algo) é de fato quem (ou o que) afirma ser;
- **confidencialidade**: garante que somente as partes envolvidas na comunicação serão capazes de processar as mensagens enviadas;
- **integridade**: protege os dados contra corrupção maliciosa ou acidental;
- **autorização**: capacidade de restrição de acesso aos recursos baseando-se em informações sobre o requisitante, o recurso a ser acessado e os detalhes específicos da requisição;
- **reputação**: permite estimar o grau de confiança nas demais entidades de um sistema;
- **anonimidade**: garante que as identidades reais no sistema permanecerão desconhecidas;
- **negabilidade**: assegura que uma entidade não possa ser responsabilizada pelos dados que detém;
- **não-repúdio**: evita que uma entidade negue responsabilidade nas ações que foram executadas.

Verifica-se claramente que alguns requisitos são base para outros (autenticidade e autorização) e, em contrapartida, outros têm funções antagônicas (autenticidade e anonimidade). Considera-se que cada aplicação P2P possui seus requisitos próprios de segurança e que deve ter métodos de segurança instanciados conforme a sua necessidade.

Segundo [Barcellos and Gaspary, 2006], estudos sobre segurança de redes P2P normalmente definem modelos de ataque, que se baseiam nas particularidades do ambiente e nos poderes do atacante. Em suma, um atacante irá conectar a rede P2P um ou mais pares maliciosos sob o seu controle. Na concretização do ataque, esses pares não necessitarão seguir corretamente os protocolos estabelecidos, quanto ao roteamento e à manutenção do topologia. Esses pares poderão também agir

em conluio para afetar o funcionamento da rede P2P. É importante considerar que muitas vezes um ataque pode ocorrer alheio à vontade do usuário, como por exemplo, através da contaminação do software P2P. Essas questões, bem como os requisitos de segurança pertinentes a sistemas P2P, serão discutidas a seguir.

Disponibilidade. Este requisito é medido em função da parcela de tempo que um determinado objeto ou serviço está disponível para acesso. No compartilhamento de arquivos se refere ao sucesso das operações de leitura e escrita de dados. Em sistemas P2P para a computação distribuída é a garantia de que os pares estão acessíveis para realizar o processamento esperado.

A principal contramedida para ataques de disponibilidade refere-se à replicação. Os principais ataques à disponibilidade estão baseados em negação de serviço (DoS) e ataques de roteamento. Negação de serviço pode ser realizada no nível de rede ou no nível da rede de sobreposição. Já ataques de roteamento referem-se a anomalias no encaminhamento das mensagens ou o seu simples descarte, ambos causados por pares maliciosos para desviar as mensagens do seu destino final. Normalmente ataques de roteamento são dependentes do tipo de rede P2P: estruturada e não estruturada.

Autenticidade. Possibilita que pares, objetos ou serviços possam comprovar que sua identidade está correta. Uma série de ataques é possível nesse contexto, dentre eles destaca-se *Sybil* [Douceur, 2002]. *Sybil* refere-se ao ataque onde um único par falsifica a existência de múltiplas identidades no sistema. [Douceur, 2002] afirma ainda que em um sistema P2P de larga escala não é possível impedir esse problema de forma distribuída. A única alternativa realmente eficaz para tratamento desse ataque é a utilização de uma autoridade certificadora de confiança ou de um mecanismo centralizado de autenticação (o que é claramente desaconselhável em um sistema P2P de larga escala).

Em sistemas P2P estruturados o fato de um par poder escolher o seu identificador ocasiona vulnerabilidades ao sistema. Um atacante, por exemplo, poderia escolher o seu identificador para apropriar-se de um conjunto de chaves com o objetivo de comprometê-las, com o encaminhamento de informações de localização incorretas. Para o tratamento desse ataque os sistemas P2P devem atribuir identificadores aleatórios aos pares.

Integridade. Este quesito é diretamente aplicável em sistemas P2P de compartilhamento de conteúdo, onde a integridade garante que a adulteração de arquivos possa ser identificada. Objetos podem ser alterados no local em que estão armazenados ou em trânsito. Criptografia baseada na geração do *hash* de conteúdo pode ser utilizada para detectar se um dado objeto está ou não corrompido. Em geral, como violar métodos de criptografia é computacionalmente caro, atacantes tendem a gerar versões falsas de determinados arquivos e disseminá-las na rede P2P. Esse ataque é denominado *poluição de arquivos* e será discutido em detalhes no Capítulo 3.

Reputação e Confiança. Quando pares não colaboram ou não executam corretamente o protocolo especificado é necessário responsabilizá-los de alguma forma, para inibir este tipo de comportamento. Conforme [Marti and Garcia-Molina, 2006] pares com comportamento incorreto podem ser divididos em egoístas ou maliciosos. Pares egoístas (também chamados de pares-carona ou "*free-riders*") são empregados para obter o máximo possível do sistema, contribuindo o com o mínimo possível de recursos. Pares maliciosos, em geral, objetivam prejudicar determinados pares ou mesmo a rede P2P como um todo.

Em geral, podem ser empregados mecanismos para responsabilizar os pares por suas ações: sistemas de gerência de reputação e confiança e métodos baseados em micro-pagamentos. Micro-pagamentos baseiam-se em oferecer remuneração explícita a pares que têm os seus serviços utilizados. Sistemas de reputação e confiança baseiam-se em manter escores que refletem o comportamento ou estado de determinado par, recurso ou objeto. A partir desse mapeamento, pares optam por relacionar-se com outros que possuam boa reputação e por obter conteúdo com reputação adequada. Mecanismos de gerência de reputação serão tratados no Capítulo 3.

Autorização ou Controle de Acesso. Mecanismos de autorização para sistemas P2P ainda possuem baixa adoção, conforme [Barcellos and Gaspary, 2006]. Isso se deve principalmente pelos requisitos a serem satisfeitos por soluções de controle de acesso para a sua adequação a sistemas P2P, tipicamente comprometendo a sua escalabilidade (normalmente mecanismos de autorização se baseiam em serviços centrais) e anonimidade.

Autorização é um requisito importante para a adoção mais ampla de P2P, por exemplo em ambientes corporativos. Mesmo em sistemas de compartilhamento de arquivos, podem ser utilizados métodos para a autorização de downloads conforme as garantias necessárias a determinado ambiente. Em [Tran et al., 2005] é proposto um sistema de controle de acesso baseado em reputação. Para acesso aos objetos é necessário que os pares possuam escore de confiança acima de determinados níveis.

Anonimidade e Negabilidade. Em redes P2P a anonimidade garante não-identificação: (a) do autor ou do responsável pela publicação de determinado objeto ou serviço; (b) da identidade do par que está armazenando um objeto ou serviço; (c) da identidade e do conteúdo do objeto; e (d) dos detalhes da requisição para recuperar determinado objeto [Dingledine et al., 2001]. Soluções para prover anonimidade são em geral motivadas por necessidade de discurso livre, principalmente em países com fortes restrições à liberdade de expressão.

As soluções propostas até o momento baseiam-se em uma técnica conhecida como *mix de Chaum* [Chaum, 1981], onde dois pares que desejam se comunicar utilizam um conjunto arbitrário de *relays*, até que um deles encaminha a mensagem ao destinatário. São exemplos dessa classe de soluções roteamento cebola (*onion routing*) [Goldschlag et al., 1999] e Tor [Dingledine et al., 2004]. Já negabilidade pode ser entendida como um componente da anonimidade, onde os pares não podem ser responsabilizados pelo conteúdo que armazenam.

Capítulo 3

Trabalhos Relacionados

Este capítulo apresenta a revisão bibliográfica dos principais trabalhos relacionados à poluição de arquivos. Nele são discutidas as propostas de solução existentes. Além disso, são também estudados sistemas de gerência de reputação, já que há uma forte tendência expressa na literatura em utilizá-los no tratamento de poluição de arquivos. Em ambientes nos quais se tem poucas garantias, como é o caso de redes P2P, sistemas de reputação são ferramentas de segurança importantes. Sabe-se, entretanto, que os mesmos tornam possíveis novas classes de ataque. Dessa forma, avalia-se o uso de tais sistemas, bem como analisa-se a sua adequação (ou técnicas propostas pelos mesmos) como contramedidas ao problema de poluição de arquivos em redes P2P.

3.1 Poluição de Arquivos em Redes Peer-to-Peer

Poluição de arquivos é atualmente uma das maiores ameaças a sistemas P2P de compartilhamento de conteúdo. Conforme demonstrado por [Liang et al., 2005a] esse ataque tem sido realizado em larga escala de forma a impedir a livre distribuição de conteúdo em redes P2P. Segundo esse mesmo trabalho, a indústria fonográfica tem financiado tais ataques em função dos prejuízos ocasionados pela distribuição de músicas de forma ilícita.

Independente das questões ligadas à distribuição de conteúdo protegido, poluição de arquivos constitui-se em um problema relevante e que necessita ser tratado. Esse tipo de ataque pode também ser realizado em redes P2P de compartilhamento de conteúdo legal, como é o caso do QTRAX [QTrax, 2006]. Recentemente, grandes empresas têm divulgado a liberação de conteúdo a partir de tecnologias P2P, como é o caso da BBC [BBCP2P, 2007]. Além disso, sistemas

P2P têm potencial de uso muito mais amplo do que a simples distribuição de conteúdo protegido por leis de direitos autorais. Por exemplo, em países com política autoritária esses sistemas podem ser utilizados para apoiar a liberdade de expressão e ataques de poluição poderiam ser realizados como forma de censura. Em suma, ataques de poluição impactam negativamente o uso das redes P2P, o que motiva a pesquisa de métodos para o seu tratamento.

3.1.1 Caracterização

Os termos apresentados nesta seção baseiam-se no modelo básico de distribuição de conteúdo, presente nos trabalhos [Liang et al., 2006] e [Kumar et al., 2006]. Apesar de abordados genericamente, os conceitos são válidos para a grande maioria dos sistemas P2P de compartilhamento de arquivos.

Um determinado arquivo na rede P2P, tipicamente som, vídeo, documento ou programa é um *título*. Um título é disponibilizado na rede P2P em diferentes *versões*. Por exemplo, no caso de músicas, versões podem ser criadas por diferentes codificadores (*encoders*). Verifica-se que arquivos populares nas redes P2P são oferecidos sob um número bastante alto de versões. Cada versão possui um identificador único, normalmente criado a partir do *hash* do seu conteúdo.

Em função do download de versões pelos usuários, são espalhadas *cópias* de uma mesma versão em diversos pares da rede P2P. Cada par envia informações sobre suas cópias quando solicitado, normalmente em resposta às consultas de outros pares. A Figura 3.1 ilustra um exemplo de uma consulta realizada na rede eDonkey. Nota-se na figura o número disponível de cópias de cada versão (coluna Disponibilidade), o que informa o número de pares que disponibilizavam cada versão no momento da consulta.

Neste contexto, poluição de conteúdo (também chamada de "envenenamento") refere-se a distribuir um número massivo de arquivos incorretos na rede P2P, dificultando a localização e obtenção de arquivos íntegros [Christin et al., 2005]. Boa parte dos usuários escolhe as versões a serem obtidas pelo número de cópias disponível na rede P2P, o que intuitivamente se constitui em uma garantia de qualidade e de desempenho na obtenção do título. Um atacante pode valer-se de *Sybil* [Douceur, 2002] para expor um alto número de cópias de determinada versão, o que aumenta a eficiência do ataque a um custo relativamente baixo.

É importante atentar para o fato de que pares corretos também podem propagar conteúdo corrompido involuntariamente. Conforme [Costa et al., 2006] muitos usuários tendem a não inspecionar rapidamente os arquivos obtidos e devido a isso não apagam imediatamente o conteúdo poluído. Mesmo durante o download

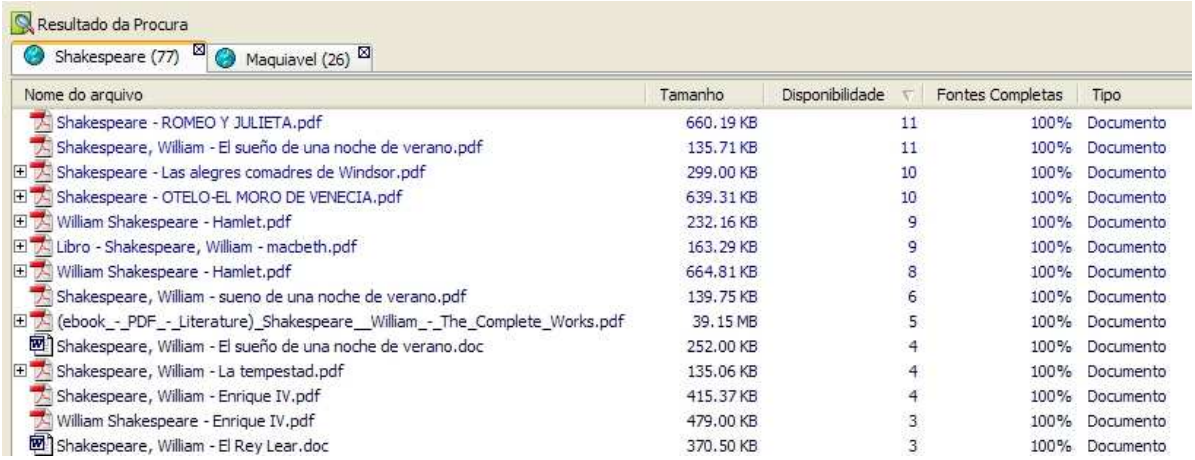


FIGURA 3.1 – Exemplo de consulta na rede eDonkey

de uma versão poluída, normalmente o par já está também disponibilizando este conteúdo a outros participantes.

Segundo [Liang et al., 2005a] existem dois tipos principais de poluição em redes P2P: **poluição de conteúdo** e **poluição de metadados**. A modalidade mais comum de ataque é a poluição de conteúdo, na qual são disponibilizados arquivos com conteúdo incorreto, corrompido ou mesmo embaralhado. Poluição de metadados, por sua vez, refere-se ao oferecimento de conteúdo semanticamente incorreto. Por exemplo, ao ser disponibilizado um arquivo de música com o nome do artista ou do título incorreto, está sendo realizado um ataque de poluição de metadados. O arquivo de música obtido é válido, porém não possui o conteúdo que o requisitante desejava obter.

Em [Costa et al., 2006] é apresentado um outro tipo de poluição, chamado de **poluição por chave**. Esse ataque baseia-se na distribuição na rede de uma versão poluída com o mesmo identificador de uma versão correta. Isso é possível, porque em alguns sistemas P2P os identificadores são gerados por uma função *hash* aplicada em apenas partes do arquivo. Alterar partes do arquivo que não são consideradas na geração do *hash* compromete a integridade sem alterar o identificador do arquivo. Tipicamente, o download é obtido de diferentes pares na rede. Se apenas um bloco recebido for originário de uma cópia corrompida, o arquivo inteiro estará comprometido. Por exemplo, o algoritmo *uuhash* considera apenas determinadas partes do arquivo na geração de seu identificador, tornando a rede FastTrack vulnerável a esse ataque.

3.1.2 Pesquisas Recentes na Área

Após a apresentação dos aspectos fundamentais sobre ataques de poluição em redes P2P, são discutidas as principais contribuições nessa área. Em [Christin et al., 2005], são expostas três estratégias utilizadas na realização de ataques de poluição: *injeção randômica de poluição*, *injeção de poluição baseada em replicação* e *injeção randômica com replicação*. Injeção randômica de poluição simula a distribuição de conteúdo acidentalmente poluído, ou seja, o poluidor encaminha conteúdo gerado aleatoriamente aos requisitantes. Esse método cria inúmeras versões diferentes de determinado título na rede. A motivação para isso é dificultar que o usuário localize as versões íntegras, pelo grande número de versões incorretas. Injeção de poluição baseada em replicação, por sua vez, consiste na inserção de múltiplas cópias de um mesmo arquivo poluído na rede. Essa técnica tem grande efeito, pois os usuários tendem a selecionar os arquivos a serem obtidos a partir do número de cópias disponíveis. A terceira e última estratégia, denominada injeção randômica com replicação, é a combinação das técnicas anteriores: inserir massivamente versões poluídas com várias cópias na rede P2P. Um atacante para isso tenderá a utilizar os recursos de que disponha ou mesmo *Sybil* para simular a existência de inúmeras entidades. Os autores em [Christin et al., 2005] afirmam que até então não são conhecidos métodos capazes de resolver completamente os ataques de poluição baseados em replicação. Possivelmente um sistema de reputação mais elaborado, que considere a reputação dos arquivos de forma mais robusta, possa ser aplicado no tratamento do ataque.

Uma análise formal sobre poluição é realizada por [Kumar et al., 2006]. Nesse trabalho, a disseminação de poluição em redes P2P é estudada a partir da modelagem de fluidos. As diferentes técnicas de disseminação de poluição são estudadas e analisadas frente aos diferentes comportamentos de seleção de arquivos dos usuários. A disseminação de poluição é estudada a partir de duas estratégias de seleção principais: seleção orientada ao número de cópias e orientada ao número de versões. Na primeira, o usuário escolhe a versão priorizando as que possuem maior número de cópias disponível. Já na seleção orientada à versão, os usuários escolhem determinada versão sem levar em consideração o seu número de cópias existente.

Basicamente, na seleção orientada ao número de cópias, quanto maior o número inicial de cópias poluídas, maior tenderá a ser a taxa de disseminação de poluição no ambiente. Uma conclusão indireta, mas fundamental, é que será vantajoso a um atacante disseminar arquivos poluídos antes do lançamento e conseqüente popularização de um título, garantindo um alto número inicial de cópias poluídas.

Em redes onde os usuários tendem a efetuar o download sem considerar o número de cópias, quanto maior o número de versões poluídas, maior a probabilidade de um usuário escolher uma versão poluída. O artigo demonstra também que com pares carona e abandonamento[1], a obtenção de versões corretas é prejudicada ainda mais. Uma das principais contribuições deste estudo é o modelo analítico oferecido que pode ser utilizado por estudos posteriores na área.

Em [Costa et al., 2006] é realizado um estudo que compara a disseminação de poluição por chave e poluição de conteúdo. É também avaliada a mediação de um moderador para a contenção de ataques de poluição. Inicialmente, a disseminação de poluição é agravada pelo fato de os usuários não apagarem imediatamente conteúdo poluído após o seu download. Com isso, é exposta a necessidade de métodos de incentivo para que usuários inspecionem e apaguem rapidamente os arquivos quando o seu conteúdo é incorreto. Como conclusão desse estudo tem-se que os ataques de poluição por chave disseminam conteúdo corrompido mais rapidamente que a poluição de conteúdo. Mesmo quando os usuários possuem um alto nível de incentivo para apagar conteúdo poluído, a poluição de chave se mostra eficiente, pois o download de apenas uma parte de um arquivo poluído compromete todo o download realizado. Outro ponto ressaltado por esse trabalho é que a presença de um moderador é uma alternativa para a contenção de poluição, entretanto, refere-se a uma abordagem não escalável. Para eficácia desse método é também necessário que os usuários possuam incentivos para reportar ao moderador os arquivos corrompidos que foram obtidos a partir da rede.

3.1.3 Propostas de Solução

Poluição é um problema atual que tem prejudicado fortemente as redes P2P de compartilhamento de conteúdo. Apesar de ser uma questão crítica, atualmente, o tratamento de poluição em redes P2P pode ainda ser considerado um problema em aberto. A maior parte dos estudos realizados até o momento foca-se no mapeamento do ataque e na avaliação dos seus impactos. São expostas direções possíveis a serem seguidas para a solução dessa classe de ataques, constituindo-se de métodos abstratos e muitas vezes sem avaliação experimental ou analítica. De qualquer forma, deve-se ressaltar o mérito dos trabalhos desenvolvidos, pois criam a base necessária às propostas de solução, que ainda são poucas na literatura.

Em [Liang et al., 2005b] é proposta a criação de uma lista-negra gerada pela avaliação de metadados em redes P2P. A técnica fundamenta-se na premissa de que

[1]Abandonamento refere-se à desistência de obtenção de determinado título, após sucessivos downloads incorretos na rede P2P.

em um ataque de poluição serão empregadas máquinas em redes IP próximas (não necessariamente na mesma sub-rede, porém em sub-redes numericamente próximas). Com isso, será possível detectar os blocos utilizados por atacantes. O método de detecção baseia-se em duas fases: (a) obtenção de metadados a partir de consultas à rede P2P; (b) consolidação das informações para a detecção dos poluidores. Na fase de obtenção de dados o sistema coleta, para cada título necessário, o número de versões e cópias existentes na rede. Para cada cópia são obtidas informações de identificação sobre o par que as hospeda (IP, porta e nome de usuário). Essas informações são mantidas em um banco de dados.

A segunda fase do processo consolida as informações do banco de dados gerado. Basicamente, os endereços IP que possuem alta densidade de versões e/ou cópias de determinado título são considerados poluidores. Em um segundo momento, os mesmos são consolidados de forma a definir o intervalo de endereços IP mantidos por poluidores. Esses endereços são diretamente adicionados a uma lista-negra.

A principal limitação desse trabalho refere-se ao fato de que a detecção dos pares poluidores ocorre independentemente do protocolo de consulta, sendo realizada por determinados pares. Com isso, será necessário que antes de se realizar a consulta, já tenha sido executado o mecanismo de detecção e criação da lista-negra. O método ocasiona também grande sobrecarga de download de metadados à rede P2P. Questões como a cooperação dos pares para a troca dessas informações não são consideradas, por exemplo. Além disso, um atacante poderá freqüentemente alterar a sua configuração (versões, cópias ou questões de topologia) para confundir o método.

[Kumar et al., 2006] defende a criação de mecanismos de lista-negra como alternativa ao tratamento de poluição. É apenas citado que, para serem eficazes, listas-negras devem considerar o intercâmbio de informações entre pares. Ou seja, pares que tiveram experiências ruins com outros participantes os adicionam na sua lista-negra. Quando consultados, irão informar essas experiências aos outros participantes. Esse método, entretanto, é contornado por um atacante, ao efetuar um *Whitewashing*, ou seja, sair e entrar novamente na rede com um novo identificador, que não está mapeado na lista-negra.

[Liang et al., 2006] segue a mesma linha e propõe um método abstrato de controle de poluição baseado na criação de uma lista-negra global, utilizando como base um sistema de reputação de pares. Inicialmente, é importante ressaltar que esse trabalho apresenta a proposta tanto para contenção de poluição de conteúdo quanto para poluição de índices [2]. Esse trabalho baseia-se na reputação de endereços

[2]Poluição de índices refere-se ao ataque onde um par malicioso informa índices falsos às

IP, agrupando-os em função da sub-rede na qual estão inseridos. Com isso, cada sub-rede terá uma reputação que reflete o comportamento dos seus pares na rede. Essa abordagem é motivada pelo fato de que um atacante irá empregar vários equipamentos para realizar ataques de poluição, e eles normalmente pertencem a uma mesma sub-rede. Esse método também visa dificultar a aplicação de *Sybil*, na medida em que as várias identificações de um mesmo par normalmente estarão na mesma sub-rede. A partir das informações de lista-negra, os pares estabelecem limites mínimos de reputação para decidir sobre a interação com os parceiros. As sub-redes que possuírem reputação abaixo de um determinado limiar serão adicionadas à lista-negra de um determinado participante. Para obter uma visão mais aprimorada, os pares trocam as informações que possuem.

Há limitações no método proposto por [Liang et al., 2006]. A primeira é que não está sendo considerado o uso de NAT. A partir de grandes provedores, poderão existir acessos de usuários corretos e de poluidores, que estarão na mesma sub-rede. Outra é que forjar endereços IP é uma técnica bastante conhecida e difundida de ataques a sistemas computacionais e que poderia ser facilmente empregada. Por fim, o método não oferece qualquer incentivo para que os usuários informem o seu testemunho de relação a outros pares. Para a obtenção de resultados de melhor qualidade, o ideal seria considerar as experiências obtidas pela maior quantidade possível de pares.

A análise desses trabalhos demonstra a ausência de mecanismos eficazes para o tratamento de poluição em redes de compartilhamento de arquivos. Pelo fato de poluição ser um problema recente, muitos dos trabalhos estudados apresentam direções de resolução que necessitam ainda de amadurecimento ou mesmo reformulação, sendo ainda necessários esforços da comunidade científica no tratamento do problema.

3.2 Gerência de Reputação em Sistemas Peer-to-Peer

Sistemas Peer-to-Peer, em grande parte baseados em anonimidade (em verdade, pseudo-anonimidade, pois determinadas informações reais dos participantes podem ser obtidas), tornam-se ambientes propícios para ações maliciosas de seus participantes – como é o caso de ataques de poluição. Na literatura, boa parte das direções de solução propostas baseia-se na utilização de sistemas de gerência de reputação, de forma isolada ou como base a outros mecanismos (como é o caso

consultas realizadas na rede P2P. Por exemplo, índices que referenciam-se objetos inexistentes. Vide [Liang et al., 2006]

de listas-negras). Dessa forma, sistemas de gerência de reputação são considerados trabalhos relacionados à essa pesquisa. Assim, essa seção objetiva avaliar sistemas de reputação propostos pela literatura. Ao final, esses sistemas são comparados, bem como é discutida a sua aplicação no tratamento de ataques de poluição em redes P2P.

Segundo [Chang et al., 2005], *confiança*, em termos computacionais, pode ser definida como a crença que um agente tem na boa vontade de outro em prover a qualidade de serviço esperada, em um dado contexto e em um determinado período. Sistemas de gerência de reputação, especificamente, possibilitam que os participantes obtenham indícios sobre a integridade de outras entidades, baseando-se no histórico das transações já realizadas com as mesmas. Em suma, sistemas de reputação constituem-se em uma das principais formas para prover confiança em redes P2P [Yu et al., 2004].

Conforme [Marti and Garcia-Molina, 2006], sistemas de reputação em redes P2P são, em geral, compostos por três funções principais: coleta de informações, ranqueamento e ações de resposta. Coleta de informações refere-se ao método de obtenção dos votos na rede. A coleta de informações é base para o mecanismo de ranqueamento, no qual as informações obtidas são consolidadas, gerando a reputação sobre o elemento avaliado – o que fornecerá subsídios à decisão sobre interagir ou não com o mesmo. O último passo no fluxo de reputação é a execução das ações de resposta, para, assim, privilegiar a operação dos pares com boa reputação e marginalizar os que não possuem. Basicamente, as ações de resposta podem resultar em incentivos ou em punições aos pares.

Segundo [Barcellos and Gaspary, 2006] são possíveis cinco estratégias de reputação em sistemas P2P: otimistas, pessimistas, centralizadas, investigativas e transitivas. Estratégias otimistas assumem que pares são confiáveis até que se tenha comprovação do contrário. Isso beneficia diretamente pares estranhos[3], na medida em que os mesmos tendem a ser normalmente selecionados para as transações. Estratégias otimistas são recomendáveis onde os benefícios de cooperação são altos ou os custos e riscos de uma possível traição são baixos [O'Hara et al., 2004]. Estratégias pessimistas de reputação propõem exatamente o contrário: assumir que todos os pares não são confiáveis até que o provem ser. Claramente, verifica-se que pares estranhos tenderão a não construir uma reputação rapidamente, na medida em que não são priorizados nas transações [Marti and Garcia-Molina, 2006].

Estratégias centralizadas consideram a existência de uma entidade central

[3]Nodos que entraram na rede recentemente, sob os quais o sistema de reputação não possui informações.

como mantenedora da reputação dos pares, o que conflita diretamente com a natureza descentralizada de redes P2P, além de consistir em um ponto central de falhas e ataque. Estratégias investigativas consistem em consultar os outros participantes para obter informações sobre um par e complementar a sua reputação local (se houver alguma). Por fim, estratégias transitivas consideram que a confiança pode ser medida em cadeias de relacionamentos. Ou seja, se um par A confia em B, B confia em C, logo A irá confiar em C. Essa estratégia baseia-se no fenômeno "*Small World*" [Milgram, 1967], no qual um par pode atingir qualquer outro em um pequeno número de saltos. Com isso, o método é escalável, à medida em que tais cadeias poderiam ser utilizadas para localizar rapidamente informações de reputação de outros participantes.

Em geral, a maior parte dos trabalhos considera apenas a reputação da identidade dos pares participantes. Conforme [Marti and Garcia-Molina, 2006], em muitos sistemas P2P as identidades são "fracas" e estão sujeitas a ataques (vide Seção 2.4). Alguns trabalhos, em especial [Walsh and Sirer, 2005] e [Damiani et al., 2002], propõem manter e considerar a reputação dos arquivos na rede. Essa técnica é também conhecida como *reputação de objetos*. Nesse sentido, a reputação está associada aos identificadores dos arquivos, únicos na rede P2P. Também argumenta-se que a reputação de objetos tende a ser mais resistente a ataques à autenticidade, na medida em que não seria vantajoso a um par sair da rede e reentrar (*Whitewashing*), já que isso não apaga a reputação do seu conteúdo.

Nas próximas subseções são apresentados sistemas de gerência de reputação em redes P2P, elicitando seus aspectos fundamentais. Ao final da seção é apresentada uma tabela comparativa entre os mesmos.

3.2.1 XREP

Em [Damiani et al., 2002] é apresentado um sistema de reputação para aplicações de compartilhamento de arquivos, denominado XREP. O sistema baseia-se em combinar a reputação de pares ("*servents*", termo que se refere à função de cliente e servidor) e de arquivos (objetos). Reputação de pares considera o seu identificador, ao passo que reputação de objetos baseia-se no *digest* do conteúdo. A reputação é gerenciada cooperativamente a partir de um algoritmo de sondagem distribuído, que reflete a opinião da comunidade sobre os downloads já realizados a partir de determinado par.

O protocolo é proposto como uma extensão do Gnutella. Tipicamente, no protocolo original, a seleção considera qualidade de oferta de conteúdo (velocidade de conexão e número de pares que possuem o arquivo desejado). A proposta

se baseia em aprimorar a seleção, considerando a reputação do objeto e do par onde o mesmo reside. Cada participante mantém um repositório de suas experiências com os outros pares. Este repositório armazena dois valores possíveis, indicando satisfação ou insatisfação em suas interações com outras entidades (pares e objetos). Posteriormente, quando consultado, um par irá encaminhar as informações solicitadas, mantidas neste repositório. O protocolo de sondagem é descrito abaixo.

1. **Localização do objeto**: o par que deseja realizar um download emite um broadcast (*Query*) com as informações sobre os objetos que deseja localizar. Os pares que possuírem objetos que satisfaçam à consulta responderão à mensagem inicial (*QueryHit*). A extensão ao Gnutella consiste em adicionar o *digest* de cada recurso junto a essa mensagem de resposta.

2. **Seleção de objetos e pares**: o requisitante seleciona o objeto conforme a sua preferência. Em seguida, ele questiona os outros participantes sobre as suas opiniões sobre o objeto e sobre o par no qual reside. Para proteger a integridade e a confidencialidade das respostas, junto à mensagem inicial é enviada uma chave pública sob a qual as mensagens devem ser cifradas. No momento em que cada par recebe a mensagem de consulta, verifica o seu repositório e encaminha ao requisitante as informações que possui.

3. **Avaliação dos votos**. O requisitante descarta os votos falsos, que não foram criptografados com a chave correta. Em seguida, ele agrupa os votos, baseando-se na sub-rede dos pares que os emitiram (tentativa de resolver *Sybil*). Por fim, o requisitante escolhe aleatoriamente um conjunto de pares de cada grupo e solicita a confirmação do seu voto, de forma a garantir que o mesmo não foi forjado.

4. **Escolha do par**: o par com a maior reputação é escolhido. Para não criar um gargalo em determinados participantes, o requisitante contacta o par escolhido para verificar se ele possui capacidade para atender à requisição. Caso não possua, o par com a reputação subseqüente é escolhido e o processo repetido sucessivamente até localizar um par capaz de atender a requisição.

5. **Download do objeto**. Após o download, o par atualiza o seu repositório com a experiência com o objeto e par selecionados.

3.2.2 EigenTrust

EigenTrust [Kamvar et al., 2003] é um sistema de gerência de reputação no qual a cada par é associado um valor de confiança global, que reflete as experiências dos participantes na rede. Os autores elencam cinco premissas principais de projeto:

- **auto-regulação**: o princípio de funcionamento deve ser mantido pelos pares participantes, sem a existência de uma entidade central;
- **anonimidade**: informações de reputação são associadas a uma pseudo-identidade, resguardando a identidade real do participante;
- **não garantir lucro a pares estranhos**: objetivando desencorajar *whitewashing*.
- **baixa sobrecarga**: considerando poder computacional, infraestrutura, armazenamento e complexidade das mensagens.
- **robustez a ataques em conluio**: com isso garante-se que os pares não possam prejudicar-se ou beneficiar-se de informações de reputação incorretas.

Basicamente, a reputação global de um par é obtida a partir das referências encaminhadas pelos outros participantes, pesados com a própria reputação dos mesmos. Essa característica faz com que pares maliciosos, normalmente, com baixa reputação, não tenham as suas opiniões fortemente consideradas.

O sistema se baseia na existência de gerentes de reputação: pares responsáveis por manter a reputação de outros participantes. Antes de interagir com um par selecionado, o requisitante solicita ao gerente informações sobre o mesmo. Existem vários gerentes no sistema, sendo que cada gerente é responsável por manter as informações de reputação de determinado par. Este mapeamento é realizado via DHT, onde é utilizado o *hash* do identificador do par para localizar o seu gerente responsável. Por questões de segurança e tolerância a falhas, é necessária a existência de mais de um gerente de reputação a cada par. A estratégia adotada para resolver este problema baseia-se em utilizar diferentes funções *hash* para mapear o mesmo identificador a diferentes pares. Além disso, um gerente de reputação não pode escolher o par ao qual irá computar a reputação, garantindo que pares maliciosos não possam beneficiar-se mutuamente ou prejudicar determinado participante.

Um gerente de reputação realiza uma inundação da rede para computar a reputação do par ao qual é responsável. Outra peculiaridade do método é que as informações de votos não são armazenadas de forma absoluta, mas sim, a sua

normalização. Dessa forma, os votos dos pares estão definidos no intervalo $[0, 1]$. O trabalho do gerente de reputação é consolidar essas informações e encaminhar a reputação do par, quando for consultado. Além disso, no download, a seleção do par ocorrerá de forma semi-aleatória, considerando a sua reputação. Isso garante que os pares com maior reputação não sejam sobrecarregados, bem como faz com que os outros pares também tenham condições de elevar a sua reputação ao agir corretamente.

3.2.3 FuzzyTrust

Em [Song et al., 2005] é apresentado um sistema de gerência reputação para redes P2P de comércio eletrônico baseado em lógica difusa (*Fuzzy*). O FuzzyTrust estabelece métodos para garantir confiança mútua entre pares desconhecidos em transações P2P. O sistema se beneficia das vantagens da lógica difusa, capaz de lidar com a imprecisão pertinente a esses sistemas. O sistema também utiliza uma DHT para manter a reputação dos pares, de forma a garantir um mecanismo eficiente de consulta e obtenção de informações de reputação. Cada par mantém duas tabelas para registro de transações e para a manutenção do escore local de cada par com quem interagiu. Para inferir a reputação global, cada par consulta as tabelas de escore dos outros pares da rede P2P.

Os princípios de funcionamento do FuzzyTrust foram definidos a partir de uma extensiva análise de dados de transações do eBay [eBay, 2007]. A principal conclusão obtida refere-se ao fato de que as transações seguem Leis de Poder: poucos usuários realizam a maior parte das transações, ao passo que muitos realizam poucas transações. Outra conclusão importante é que a maior parte das transações envolve valores baixos e apenas uma pequena parte constitui-se em transações de grande valor monetário. Este estudo originou três princípios de projeto, os quais são expressos a seguir.

- O consumo de banda de rede para a troca de informações de reputação para os pares que realizam a maior parte das transações (*hotspots*) será extremamente alto. Dessa forma, o sistema de reputação deveria considerar transações não balanceadas entre os usuários.

- O sistema de reputação não deve aplicar o mesmo ciclo de avaliação a todos os pares da rede, tendo em vista o seu grau diferenciado de utilização da rede P2P. Por exemplo, os pares responsáveis pela maior parte das transações devem possuir informações de reputação atualizadas mais freqüentemente que os demais usuários.

- As transações que envolvem maiores valores devem ser avaliadas mais freqüentemente que as demais.

Conforme já comentado, o sistema baseia-se em dois grandes processos: a manutenção de reputação local e a agregação global de reputação. Na manutenção local os pares executam inferências difusas nos parâmetros locais para gerar os escores locais. São exemplos de parâmetros a serem considerados: método e tempo de pagamento, tempo de entrega e qualidade da mercadoria. A última fase consiste na obtenção e agregação dos escores de reputação local de outros pares, obtendo a reputação global de determinado participante. São determinados três pesos de agregação: a reputação do par, a data da transação e a quantidade envolvida na transação. Essas três variáveis poderiam ser agregadas de várias maneiras, sendo que no protótipo implementado, as mesmas foram base para cinco políticas de agregação:

1. se o valor da transação é alto e a transação é recente, então o seu peso na reputação global será alto;
2. se o valor da transação é baixo ou a transação é muito antiga, então o peso de agregação será baixo;
3. se um par possui uma boa reputação e o valor da transação é alto, o peso de agregação será muito alto;
4. se a reputação de um par é boa e o valor é baixo, então o peso de agregação será médio;
5. se um par possui má reputação, então o peso de agregação será muito pequeno.

Os autores estabeleceram um estudo comparativo junto ao EigenTrust, no qual informam que ambos os sistemas tem tempos de obtenção de reputação global semelhantes, porém, com o FuzzyTrust possuindo menor sobrecarga de número de mensagens. Para trabalhos futuros, os autores pretendem considerar aspectos como anonimidade e armazenamento seguro de reputações globais.

3.2.4 TrustGuard

O trabalho [Srivatsa et al., 2005] objetiva tratar as vulnerabilidades inerentes a sistemas de reputação: ataques de traição; *shilling*, onde pares maliciosos submetem votos falsos e fazem conluio para aumentar sua própria reputação; e envio de votos sobre transações não existentes. Neste sentido, o TrustGuard apresenta as seguintes contribuições: (a) capacidade de manipular e considerar oscilações de reputação,

ocasionadas por pares maliciosos; (b) introdução de um mecanismo de admissão, que garante que somente votos de transações verdadeiras sejam considerados; e (c) proposta de mecanismos de filtragem de votos falsos baseados na credibilidade dos pares que os encaminharam.

Cada par possui três componentes principais: máquina de avaliação de confiança, gerente de transação e serviço de armazenamento. Quando um par M desejar efetuar transações com um outro par N, inicialmente utiliza sua máquina de avaliação de confiança. Esse componente coleta os votos da rede e os agrega para obter o escore de confiança sobre o par desejado. O gerente de transação, de posse dos valores gerados pela máquina de avaliação de confiança, decide se a transação deverá ser executada ou não. A máquina de avaliação de confiança também é responsável por gerar e intercambiar as provas de transação e, após isso, executar efetivamente a transação. Por fim, os pares encaminham os votos sobre a transação a gerentes designados para o armazenamento desses valores. Tais gerentes, a partir de seu serviço de armazenamento, executam testes de veracidade sobre os votos recebidos, para após isso armazená-los.

As principais contribuições deste trabalho residem em três sub-componentes: monitor de oscilações e filtro de votos falsos, máquina de avaliação de confiança e detector de transações falsas (parte do Serviço de Armazenamento). O monitor de oscilações considera o histórico de interações de um par e as suas flutuações na estimativa de confiança. Basicamente, a estratégia é considerar que as transações recentes tenham maior peso. Com isso, quando um par correto passar a apresentar comportamento malicioso, terá suas ações recentes fortemente consideradas na estimativa de sua reputação.

Conforme já citado, um par malicioso poderá valer-se do sistema de gerência de reputação para enviar votos desonestos a respeito de um determinado participante. Para resolver esse problema, o filtro de votos falsos atribui um valor de credibilidade ao voto recebido e o considera com relação a esse peso. O valor de credibilidade está diretamente relacionado à reputação do par emissor.

De forma a evitar que um par encaminhe votos sobre transações que nunca ocorreram ou mesmo de si próprio, o TrustGuard propõe que sejam trocadas provas de transação entre os participantes. Essas provas deverão ser fornecidas junto ao voto a ser encaminhado pelo par aos mantenedores de reputação. Esse mecanismo evita que um par submeta votos com quem não realizou transações, entretanto, não evita que sejam enviados votos incorretos com quem interagiu.

3.2.5 Credence

Em [Walsh and Sirer, 2005] é proposta a criação de um sistema de reputação de objetos (arquivos) chamado Credence, tendo como objetivo específico endereçar o problema de poluição em redes P2P. Esse trabalho motiva o seu desenvolvimento no fato de os pares não possuírem métodos para obter, antecipadamente, informações sobre a integridade de determinado conteúdo. Adicionalmente, muitas vezes os pares relacionam-se com participantes desconhecidos. Se pares interagissem somente com poucos componentes, métodos de reputação de identidades seriam efetivos.

O Credence habilita os pares a obter informações que atestem a autenticidade dos objetos, ou seja, o grau em que os dados condizem com a descrição informada. Inicialmente, é utilizado um método de votação, onde os usuários contribuem positiva ou negativamente sobre os objetos da rede P2P. Este trabalho é baseado principalmente no fato de que as relações com objetos são finais, dependendo unicamente de suas características imutáveis. Neste contexto, avaliações com grande variação podem ser consideradas indicadores de comportamento malicioso.

No Credence, cada objeto é composto por descritor e dados. O descritor contém metadados para facilitar a localização do objeto. A premissa de funcionamento é que os pares deverão julgar a autenticidade de cada descritor antes de realizar o download do objeto. São utilizados chaves e certificados para assegurar a autenticidade e unicidade dos votos.

Na fase de avaliação, os votos são recebidos e agregados pelo par que deseja efetuar o download de determinado objeto. Basicamente, o requisitante encaminha uma mensagem de verificação de determinado objeto à rede P2P. Os pares que possuírem informações sobre o objeto em questão encaminharão a sua opinião ao requisitante. Cada voto é pesado pelo requisitante conforme a sua relação com o emissor. O escore de relacionamento é obtido pela correlação dos históricos de votos, que pode ser positiva ou negativa, conforme a tendência em votar identicamente. Com isso, um requisitante irá considerar os votos emitidos por pares com os quais possua correlação positiva e desconsiderar os com os quais possua correlação negativa. O cálculo do coeficiente de correlação é ilustrado em [Walsh and Sirer, 2006].

3.2.6 Comparação entre Sistemas de Gerência de Reputação

Esta seção finaliza o capítulo, enfatizando a análise e comparação dos sistemas de reputação estudados. A comparação é estruturada conforme os principais componentes de sistemas de reputação, anteriormente estudados. A Tabela 3.1

estabelece um quadro comparativo entre os sistemas XREP, EigenTrust, FuzzyTrust, TrustGuard e Credence.

TABELA 3.1 – Comparação entre os sistemas de gerência de reputação estudados

Sistema de Reputação	Coleta de Informações	Escore e Ranqueamento	Seleção e Ações de Resposta
XREP	Pares consultam diretamente a rede para obter a reputação de determinado elemento. Coleta de votos a partir de inundação.	Capacidade de descartar os votos incorretos. Agrupa os votos em clusters para dificultar Sybil. É mantida a reputação de objetos e de pares.	O par com a melhor reputação é escolhido. O requisitante inicialmente contata o par escolhido para verificar se ele possui capacidade para atender a requisição. O objeto é selecionado conforme a preferência do usuário, em função de sua reputação.
EigenTrust	Existência de gerentes de reputação, que devem ser consultados para a obtenção da reputação de determinado par. Coleta de votos a partir de inundação.	Os votos são pesados segundo a reputação do emissor. É mantida apenas reputação de pares.	Seleção ocorre de forma semi-aleatória, considerando a reputação de pares. Isso garante que os pares com maiores reputações não sejam sobrecarregados.
FuzzyTrust	Existência de gerentes de reputação, que devem ser consultados para a obtenção da reputação de determinado par. Coleta de votos a partir de inundação.	Os votos são agregados globalmente de forma ponderada segundo três variáveis: reputação do par emissor, data da transação e qualidade da transação. É mantida apenas reputação de pares.	Seleção ocorre de forma semi-aleatória, considerando a reputação de pares. Isso garante que os pares com maiores reputações não sejam sobrecarregados.
TrustGuard	Existência de gerentes de reputação, que devem ser consultados para a obtenção da reputação de determinado par. Coleta de votos a partir de inundação.	Valoriza as transações recentes na avaliação dos votos. Garante que somente votos de transações verdadeiras sejam consideradas. Filtragem de votos falsos, baseado na credibilidade dos pares que os encaminharam. É mantida apenas reputação de pares.	A seleção ocorre sob a solicitação do usuário. O sistema de reputação, após isso, irá estimar se o par é confiável para que o usuário decida se a transação deverá ou não ser executada.
Credence	Pares consultam diretamente a rede para obter a reputação de determinado elemento. Coleta de votos a partir de inundação.	Os votos são pesados segundo a relação do par que deseja efetuar o download e o par que emitiu o voto. Sistema destina-se a manter a reputação de objetos.	O usuário escolhe determinado objeto para ser obtido. O sistema de reputação é utilizado para estimar a autenticidade do objeto, em função dos votos emitidos.

Inicialmente, a coleta de informações na maioria dos sistemas estudados é baseada em inundação. Esses métodos são reconhecidamente ineficientes, além de não considerar as informações de pares que não estejam presentes na rede no momento da consulta. Em [Yu et al., 2004] é proposto um método alternativo, baseado na construção de árvores de confiança, nas quais os pares enviam as

consultas a apenas um subconjunto de seus vizinhos. Apesar da redução da sobrecarga, esse método ainda não considera a opinião dos pares ausentes no momento da consulta. Além disso, os mecanismos de reputação também deveriam possuir métodos eficientes de incentivo, para motivar os pares a encaminhar seus votos, quando consultados.

Com relação ao ranqueamento, verifica-se que a maior parte dos sistemas de reputação baseia-se na reputação de pares. Em compartilhamento de arquivos, considerar a reputação de objetos agrega segurança à integridade de conteúdo na rede P2P. Apenas o XREP e o Credence, dentre a bibliografia consultada, consideram a reputação de arquivos.

Em geral, a maioria dos sistemas de reputação modernos consolida os votos recebidos pesando-os com a reputação dos nodos que os emitiram. Isso se mostra uma técnica bastante adequada em sistemas P2P, permitindo desconsiderar as opiniões de pares maliciosos que poderiam valer-se do sistema de reputação para, em conluio, comprometer a rede. Basicamente, o FuzzyTrust e o TrustGuard valorizam as informações de transações recentes como método de controle ao problema do traidor, ataque em que um par acumula uma boa reputação e depois explora a mesma para agir de maneira maliciosa. Com isso, pares que mantinham um comportamento correto, ao passar a executar transações incorretas, tem a sua reputação rapidamente reduzida.

A seleção normalmente se dá com base na reputação de pares, o que pode criar gargalos no acesso aos pares com as maiores reputações. Para minimizar essa questão, a técnica normalmente utilizada consiste em adicionar certo grau de aleatoriedade à escolha de pares com reputação acima de determinado limiar. A ação de resposta mais difundida refere-se a garantir que pares ou arquivos com baixa reputação sejam evitados nas transações.

Quando se considera a aplicação dos métodos atuais no combate à poluição de conteúdo em redes P2P, verificam-se algumas limitações. Poucos sistemas de reputação são orientados para resolver essa classe de ataques. Uma série de estudos têm sido conduzidos no sentido de mapear e delimitar o problema de poluição em redes P2P. É esperado assim que métodos de contenção estejam alinhados a tais trabalhos, o que não ocorre com os sistemas de reputação existentes. Uma possibilidade citada na literatura recente, refere-se a utilizar informações de reputação como base a mecanismos mais robustos, como é o caso de lista-negra.

O único sistema de reputação estudado que objetiva tratar especificamente ataques de poluição é o Credence. Esse sistema propõe endereçar este problema a partir de um sistema de reputação de objetos (versões), que possibilita ao usuário

obter informações sobre a integridade do conteúdo que deseja obter. O Credence, entretanto, não é adequado à utilização em ambientes com alta taxa de poluição, onde a localização de versões corretas é bastante dificultada. Até a detecção das versões íntegras pelo mecanismo de reputação, grande quantidade de conteúdo poluído já pode ter sido disseminada na rede.

No método proposto nesta dissertação o fundamento principal é liberar a taxa de disseminação de novas versões de forma proporcional à confiança na integridade de seu conteúdo. Isso torna possível conter poluição enquanto o conteúdo ainda não foi ou está sendo obtido por muitos participantes. Limitar a taxa de disseminação, além de propiciar segurança aos usuários, reduz a eficiência do ataque, pois segundo [Costa et al., 2006] conteúdo incorreto obtido da rede não é rapidamente analisado e apagado por usuários, tornando-o disponível a outros pares na rede. Acredita-se, assim, que o método de contenção proposto neste trabalho é mais efetivo e robusto do que o controle imposto por sistemas de reputação tradicionais.

Capítulo 4

Contenção de Poluição em Ambiente Ideal

Conforme demonstrado no capítulo anterior, poluição de arquivos é um problema relevante e que necessita ser tratado. Métodos de contenção eficazes devem ser capazes de conter rapidamente a disseminação de arquivos poluídos na rede P2P. Contramedidas já propostas ao ataque baseiam-se em sistemas de reputação (de pares e de versões) ou na criação de listas-negras. Essas alternativas, entretanto, não são suficientes para tratar ataques realizados em larga escala. Por exemplo, enquanto um título é recente, um poluidor pode gerar indefinidamente novas versões e além disso pode freqüentemente modificar a sua identificação para comprometer sistemas de reputação ou de lista-negra. A partir de um número alto de versões na rede, reputar as versões corretas pode levar um tempo arbitrariamente alto. Até que isso ocorra, a disseminação de versões poluídas estará ocorrendo em grande quantidade.

A proposta deste trabalho é analisar a criação de métodos capazes de limitar a disseminação de poluição, tornando possível a sua detecção e conseqüente contenção. O rumo a ser seguido refere-se a analisar contenção de poluição a partir da definição de limites ao número instantâneo de downloads em uma rede P2P. Este limite é estabelecido de acordo com a reputação da versão, dinamicamente obtida a partir de votos na rede P2P. Com isso, é possível estabelecer o número permitido de downloads diretamente proporcional à reputação do arquivo. O objetivo principal do método de contenção proposto é *minimizar a disseminação de versões poluídas na rede P2P*. Entretanto, a imposição de limites de downloads deve ser realizada de forma controlada e comedida para não prejudicar significativamente a disseminação de versões íntegras na rede. Dessa forma, um dos principais requisitos do sistema é *penalizar minimamente a disseminação de versões íntegras na rede P2P*.

Este capítulo apresenta o funcionamento do método de contenção proposto em um ambiente simplificado, dito ideal. No próximo capítulo, o método será estendido a um ambiente distribuído, onde são analisadas e comparadas as suas diferentes formas de implementação.

4.1 Modelagem

Esta seção apresenta a proposta de contenção de poluição em um ambiente idealizado: memória compartilhada, infinitos processadores e pares que respeitam o protocolo estipulado. Sem perda de generalidade, a modelagem é feita com base em uma única versão de um título. Pares disponibilizam cópias de uma versão, que exceto no momento inicial, são obtidas por download de outros pares. Há um número infinito de pares desejando participar do compartilhamento de uma versão, efetuando seu download e então disponibilizando sua cópia. O crescimento da rede é limitado pela capacidade dos pares (número máximo de uploads simultâneos que um par pode realizar), denotado como "grau de disseminação" ou δ. Um "semeador" (*seeder*) é um par que possui uma cópia da versão e a disponibiliza a outros pares, os "sugadores" (*leechers*).

O funcionamento básico da rede resume-se ao processo natural de download: um sugador contacta um semeador e solicita o download de uma versão. Ao final do download, caso a cópia recebida seja íntegra, o sugador transforma-se em semeador e passa a disponibilizar até δ uploads de cada vez (caso contrário, para efeito de modelo, o sugador não se torna semeador e exclui a cópia obtida).

Com o método de contenção, ao final do download o par avalia a integridade da versão e emite um voto, positivo ou negativo, sobre a mesma. Um voto positivo atesta a sua integridade. Os totais de votos positivos e negativos são mantidos em r e s, respectivamente, e usados para obter o escore de reputação da versão. Dessa forma, a partir de r e s podem ser obtidos os escores de crença e descrença de que a versão é de fato correta. O método proposto neste trabalho baseia-se na Lógica Subjetiva [Josang et al., 2006], explicada a seguir.

A Lógica Subjetiva baseia-se na avaliação de uma proposição $\omega = (b, d, u, a)$, onde b, d, u e a indicam crença (*belief*), descrença (*disbelief*), incerteza (*uncertainty*) e fator moderador (*base rate*), respectivamente; $b, d, u, a \in [0, 1]$ e $b + d + u = 1$. O fator a define o peso da incerteza no cálculo do escore unificado de reputação. Este

escore é obtido por $E[\omega] = b + au$. Conforme demonstrado em [Josang et al., 2006], em um sistema binário de reputação, o escore é expresso pela Equação 4.1.

$$E[\omega] = \frac{r + 2a}{r + s + 2} \tag{4.1}$$

Neste caso, a reputação é usada para refletir a expectativa de um conteúdo não ser poluído; mais precisamente, ω significa "a versão possui conteúdo íntegro". A Figura 4.1 ilustra o valor de $E[\omega]$ em função do número de votos (no eixo x), para três casos: reputação de uma versão correta ($r = x$,$s = 0$); de uma versão indefinida ($r = \frac{x}{2}$, $s = \frac{x}{2}$) e de uma versão poluída ($r = 0$, $s = x$). Considera-se ainda os fatores moderadores $a = 0,9$, $a = 0,5$ e $a = 0,1$, gerando nove curvas no total. Conforme pode ser visto na figura, o fator moderador a influi principalmente na reputação inicial, onde a incerteza é muito alta. Conforme aumenta o número de votos, o primeiro grupo de curvas converge para um valor próximo de 1, o segundo, a 0,5 e o terceiro a 0, em velocidade influenciada por a.

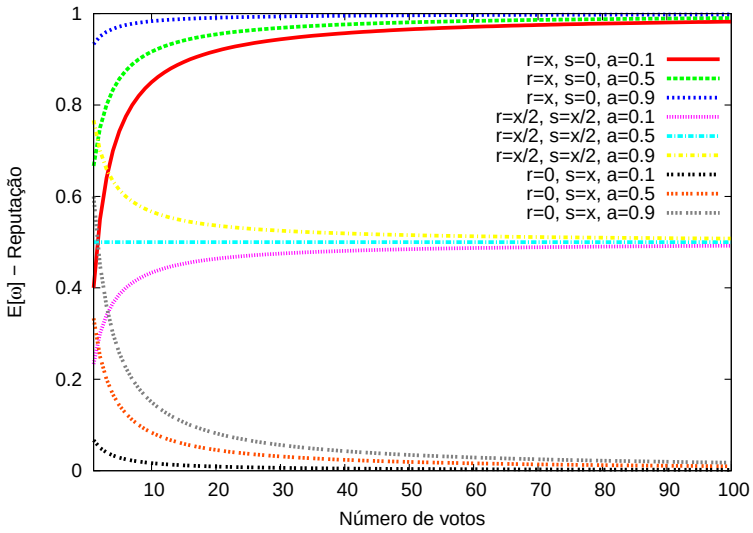


FIGURA 4.1 – Reputação Teórica $E[\omega]$

Para ser resiliente a ataques de traição (conforme mencionado na Seção 3.2.6, *traição* refere-se ao ataque onde uma entidade acumula uma reputação artificial, que depois é explorada de forma maliciosa) a solução adotada segue a literatura atual em sistemas de reputação: atribuir maior peso aos votos mais recentes no cálculo da reputação final. Neste sentido, considera-se duas possibilidades de solução, mencionadas a seguir.

- *Aplicar um fator de longevidade λ ao valor acumulado de votos já recebidos.* Como sugerido em [Josang et al., 2006], a cada unidade de tempo, os votos

positivos e negativos já recebidos sofrem um decréscimo ($r \leftarrow r \times \lambda$ e $s \leftarrow s \times \lambda$), onde $0 \leq \lambda \leq 1$. Naturalmente, quanto maior o valor de λ, maior será a influência do histórico na reputação atual $E[\omega]$ e menor será a desvalorização de um determinado voto ao longo do tempo.

- *Considerar uma janela de κ unidades de tempo no cômputo da reputação.* No tempo t serão apenas considerados os votos recebidos entre $t - \kappa + 1$ e t. Naturalmente, quanto maior o valor κ, maior a influência do histórico na reputação final $E[\omega]$. Como diferença da abordagem anterior, os votos não são desvalorizados a cada rodada, mas sim desconsiderados totalmente após κ unidades de tempo.

Assim, o método de contenção proposto se baseia em $E[\omega]$, tal como expresso até o momento, para obter a reputação de uma versão e limitar a sua taxa de downloads simultâneos. Duas variáveis são fundamentais ao método de contenção: X e Y. X representa o número máximo de downloads permitido em determinado momento, enquanto Y representa o número atual de downloads em realização. O valor de X é computado em função dos votos positivos e negativos emitidos, ou seja, X é baseado em $E[\omega]$. O valor de Y é incrementado quando um download inicia, e decrementado quando o download é finalizado (completado ou abortado). Por fim, os valores X e Y são empregados pelo método de contenção ao garantir que $Y \leq X$. Define-se a seguir o comportamento desejado a X em três casos básicos :

- $r \gg s$: X deve atingir um limiar superior, denominado X_{free}, partir do qual os downloads são inteiramente liberados (sem contenção);
- $r \simeq s$: X deve atingir um valor médio, denominado X_{med}, quando há uma divisão equânime entre votos positivos e negativos;
- $r \ll s$: X deve atingir um limiar mínimo, denominado X_{min}, a ser preservado, garantindo o progresso através de um número mínimo de downloads autorizados.

Ressalta-se a semântica dos limites superior e inferior definidos no método. Mesmo com uma reputação muito baixa, próxima de zero, X_{min} downloads serão garantidos. Entenda-se X_{free} como o limiar superior, a partir do qual a versão pode ser considerada correta. Ou seja, todos os downloads solicitados serão permitidos. Na necessidade de redução da taxa permitida de downloads, o valor considerado na queda deve ser X_{free} e não o número arbitrário de downloads correntes no momento.

Em suma, a configuração dos parâmetros X_{min} e X_{free} rege o funcionamento do sistema, fato pelo qual estes valores devem ser cuidadosamente definidos.

Até o momento, foram discutidos o método de obtenção da reputação e os requisitos necessários a X, mas não a forma de se obter X. Como $E[\omega]$ é um valor real com limites superior e inferior claros, define-se X em função de $E[\omega]$ através de γ, com $X = \gamma(E[\omega])$. A escolha de γ é requisito fundamental ao funcionamento do método de contenção, o que motiva a sua análise cuidadosa.

Na análise idealizada, assume-se que o tempo de download de uma versão, denotado como T_d, é fixo para todos os pares (e por simplicidade, igual a 1). Além disso, todos os demais tempos são negligíveis. O sistema trabalha em "rodadas": a cada rodada, um par solicita autorização e caso seja autorizado, efetua o download, verifica a integridade da versão obtida, e emite um voto, positivo ou negativo. Caso a solicitação seja negada, o par volta a efetuar uma nova solicitação apenas na rodada seguinte. Além disso, considera-se que há sempre pares solicitando o download (sistema "ganancioso") e que a disseminação de uma versão sem controle é realizada sobre grau δ, conforme já comentado. Na próxima seção diferentes alternativas à γ serão discutidas.

4.2 Definição e Análise de γ

Essa seção avalia alternativas para γ. Dois comportamentos de função serão avaliados neste trabalho: linear (γ_a) e exponencial (γ_b). São discutidas as vantagens e desvantagens de ambos os casos. A seguir são apresentados os critérios utilizados na avaliação das funções estudadas.

- **Efetividade na contenção de versões poluídas**: apresenta um ambiente com uma versão poluída, comparando a sua disseminação com e sem o método de contenção proposto. É esperado que a função imponha fortes restrições de download a este cenário. No experimento, foi considerado um ambiente com 1 semeador inicial e grau de disseminação $\delta = 2$, ou seja, cada semeador faz até dois uploads por vez. É analisada a contenção ocasionada por votos subseqüentes, negativos e divididos, em um período equivalente a 20 unidades de tempo (o que pode corresponder a inúmeros downloads, devido ao paralelismo).

- **Penalidade ocasionada ao download de versões íntegras**: ilustra o impacto ocasionado à disseminação de versões corretas. Uma função para ser adequada não deve afetar fortemente a disseminação neste cenário, sob

pena de comprometer o funcionamento da rede. No experimento é analisada a emissão de votos positivos, com um semeador inicial, $\delta = 2$ em 20 unidades de tempo.

- **Adequação a ambientes dinâmicos**: Deve-se avaliar a aplicação das funções em cenários com mudanças na tendência dos votos, garantido o seu melhor entendimento. A avaliação é realizada em 4 períodos distintos, T_1, T_2, T_3 e T_4, cada um com 50 unidades de tempo. Um período T_n pode exibir uma entre três tendências: votos positivos (**P**), negativos (**N**) ou divididos (**D**). Além disso, todas as configurações iniciam com um semeador e considera-se um sistema "ganancioso" com infinitas requisições a serem atendidas. A análise das diferentes funções foi realizada considerando todos os casos possíveis, variando também os diferentes parâmetros disponíveis.

Em relação ao tratamento de traição, ambas as alternativas foram avaliadas (uso do coeficiente de longevidade λ e da janela κ). Verificou-se que a utilização de uma janela de reputação κ apresentou resultados superiores ao fator de longevidade λ.

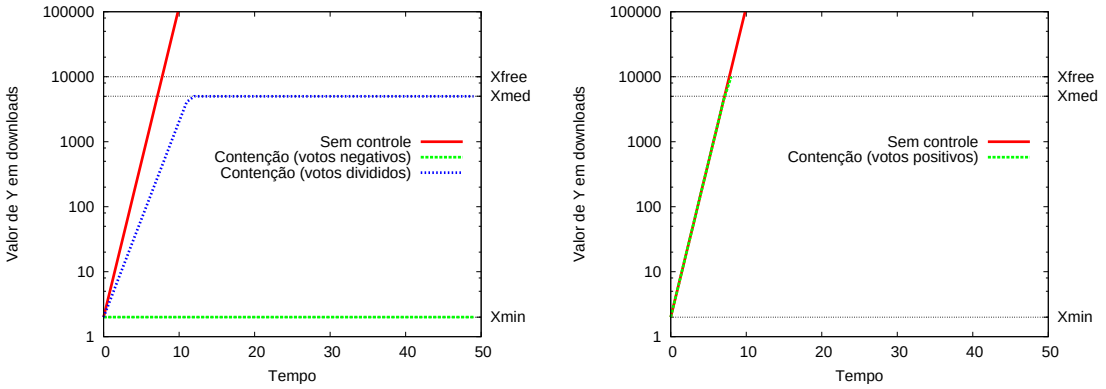
4.2.1 Comportamento Linear: γ_a

A maneira mais direta de se especificar uma função refere-se à sua modelagem em comportamento linear. Conforme anteriormente comentado, o valor de $E[\omega]$ está definido entre $[0, 1]$. O comportamento esperado de γ_a varia entre $[X_{min}, X_{free}]$. Devido à linearidade, tem-se X proporcional à $E[\omega]$. A partir dos extremos, obtém-se dois pares ordenados $(0, X_{min})$ e $(1, X_{free})$, onde, trivialmente pode ser aplicada a Equação da Reta [Leithold, 1986] para obter a função:

$$X = E[\omega](X_{free} - X_{min}) + X_{min} \tag{4.2}$$

A seguir é expressa a avaliação de γ_a, segundo os critérios estabelecidos. Para o tratamento de traição foi utilizada a janela de reputação κ=40, que significa que as 40 unidades de tempo que antecedem o momento atual serão utilizadas no cálculo de $E[\omega]$ e, por conseguinte, de X.

Efetividade do método na contenção de versões poluídas. A contenção de versões poluídas é mostrada na Figura 4.2(a). Verifica-se que a curva sem contenção apresenta uma disseminação agressiva, ilustrando o espalhamento de uma versão poluída. A contenção de poluição é avaliada sobre duas formas: emissão de votos divididos e de votos negativos. Com a inserção de votos negativos pelo



(a) (b)

FIGURA 4.2 – Disseminação de versão utilizando função linear (γ_a): (a) poluída e (b) correta, com X_{min}=2, X_{free}=10000, κ=40 e a=0,1.

download dos sugadores, a taxa de download é rapidamente contida, não permitindo a sua elevação. Já com votos divididos, a curva converge a $X_{med} \cong 5001$, ilustrando a abordagem otimista da função. Conforme o contexto, X_{med} poderá representar uma taxa bastante elevada para garantir a contenção adequada de poluição.

Penalidade ocasionada à disseminação de versões íntegras. O método de contenção atribui certa penalização à disseminação de versões corretas. Um dos requisitos principais deste método é justamente minimizar essa penalidade. A Figura 4.2(b) ilustra essa análise. Em suma, verifica-se que neste experimento a penalidade inserida é nula: até o tempo 8 a disseminação sem contenção e com contenção possuem o mesmo comportamento; após o tempo 8, a curva com contenção atinge X_{free}, onde qualquer quantidade de downloads passa a ser liberada.

Adequação a ambientes dinâmicos. A Figura 4.3 apresenta a adequação de γ_a ao cenário N-D-P-N, ilustrando X, Y e $E[\omega]$ derivados. O eixo x representa o tempo, em downloads consecutivos, enquanto o eixo y está em escala logarítmica para X e Y, porém linear para $E[\omega]$. São utilizados como parâmetros $X_{min} = 2$, $X_{free} = 10000$, $\delta = 2$, κ=40 e a=0,1. No primeiro período, com a inserção massiva de votos negativos, verifica-se a queda de X proporcional à queda de $E[\omega]$. Y, entretanto, mantém-se baixo, pois quando um sugador obtém uma versão que julga ser incorreta, além do voto, ele exclui a cópia, não tornando-se semeador. No segundo período, onde há inserção de votos divididos, verifica-se a elevação de Y até seu valor médio (quando $E[\omega] = 0,5$; $X = \frac{X_{free}+X_{min}}{2}$). A partir de T_2, Y passa a ser limitado por X. Em T_3, os votos positivos fazem com que $E[\omega]$, X e Y aproximem-se dos patamares máximos. Para fins de ilustração, considera-se que o número de downloads está limitado em X_{free}, mas na prática este limiar indica a liberação de

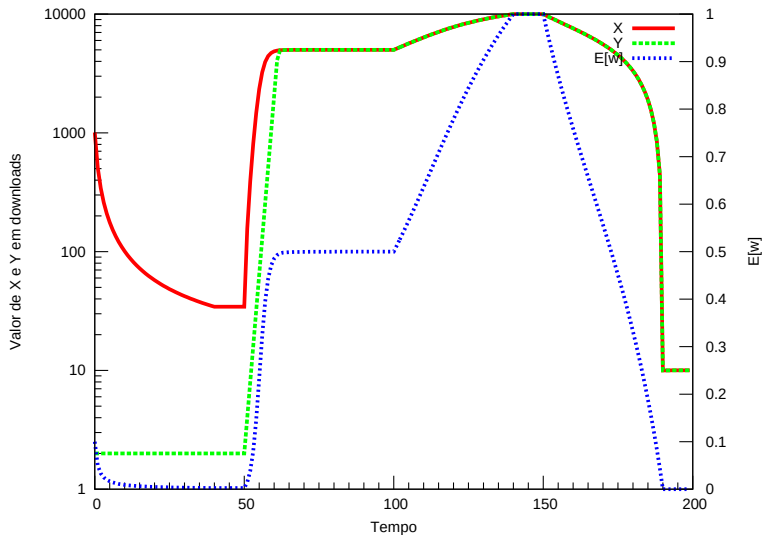


FIGURA 4.3 – Contenção em cenário N-D-P-N utilizando função linear (γ_a), com X_{min}=2 e X_{free}=10000, κ=40 e a=0,1.

qualquer número de downloads. Por fim, Y e X caem proporcionalmente a $E[\omega]$.

Outros comportamentos de X são analisados na Figura 4.4, nos quais são utilizados os mesmos parâmetros já definidos. Essa análise concentra-se unicamente no limite de contenção atribuído por X. Diferentes casos são analisados: P-D-D-D, P-N-N-N e N-P-N-P. Em P-D-D-D há três períodos consecutivos de votos divididos, após a inserção de votos positivos em T_1. Em P-D-D-D verifica-se que o limiar X_{free} foi atingido por X, que em seguida converge a $X_{med} \cong 5.001$. No caso P-N-N-N verifica-se que em T_1, X converge rapidamente a X_{free}. Com a inserção de votos negativos, em T_2, X atinge X_{min}; em T_3, X tem uma pequena elevação, que se mantém em T_4. Esse comportamento de X em T_3 e T_4, em princípio anômalo, refere-se à grande quantidade de votos negativos que passa a ser desconsiderada, conforme κ. Com a isso a reputação tem uma pequena elevação, que afeta igualmente X. Em N-P-N-P, que se refere à alternância entre períodos de votos positivos e negativos, ocorre a elevação da curva a X_{free} nos períodos de votos positivos. Nos períodos em que ocorre a inserção de votos negativos a função nem sempre atinge X_{min}. Em T_3, X atinge X_{min} pela grande quantidade de votos negativos recebida, fruto do número de alto de votos negativos recebido no início do período.

Avaliação geral de γ_a. Por referir-se a uma função linear, o comportamento reflete igualmente o de $E[\omega]$. Sua aplicação a ambientes dinâmicos ilustra a sua dificuldade em reduzir o limite de downloads a X_{min} quando há um número baixo de votos negativos no sistema, mesmo sendo este comportamento constante. A análise da influência do método de contenção na disseminação de versões poluídas

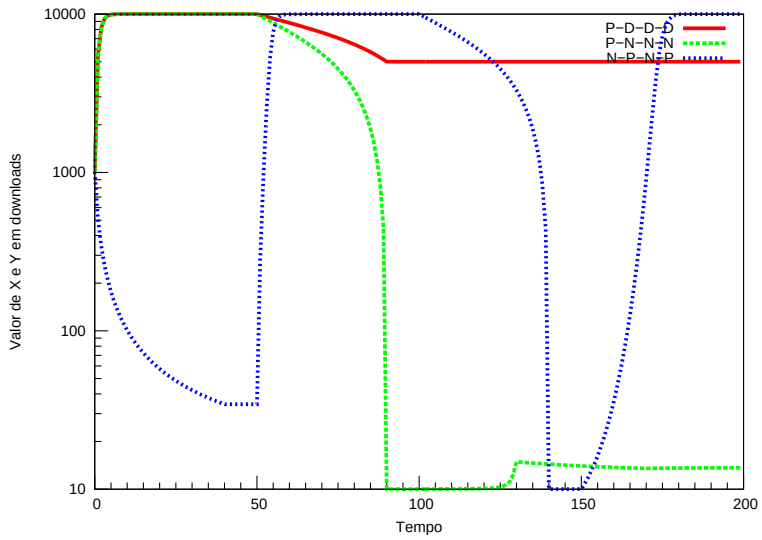


FIGURA 4.4 – Contenção em cenários dinâmicos utilizando função linear (γ_a), com X_{min}=2 e X_{free}=10000, κ=40 e a=0,1.

demonstrou a eficiência da função, limitando a poluição com votos negativos e divididos a X_{min} e X_{med}, mesmo que este último apresente uma abordagem menos restritiva. A redução de X_{med} pode ser obtida a partir da redefinição de X_{min} e X_{free}. Claramente, tem-se o *trade-off* entre a definição de um limite superior e inferior relevantes e da convergência média necessária.

4.2.2 Comportamento Exponencial: γ_b

Foi avaliada também a utilização de funções exponenciais como alternativa à função γ. Optou-se, neste sentido, por uma função com a possibilidade de ponderar taxa de crescimento e convergência do valor X resultante. Neste contexto, é apresentada a função γ_b:

$$X = \left(\frac{\alpha}{\max(\beta, (1 - E[\omega]))} \right)^{\alpha + E[\omega]} \tag{4.3}$$

Resumidamente, a combinação de α e β gera configurações de X_{min}, X_{free} e graus diferentes de crescimento. Para a avaliação, foram considerados os valores α=1,3 e β=0,024, que originam $X_{free} \cong 10000$, $X_{med} = 6$ e $X_{min} = 2$, considerando arredondamento superior. Para a comparação com a abordagem anterior também foi utilizado κ=40 unidades de tempo. A seguir é realizada a avaliação de γ_b.

Efetividade do método na contenção de versões poluídas. A Figura 4.5(a) mostra, inicialmente, que a curva sem contenção sobe agressivamente, levando a uma rápida disseminação da cópia poluída. Segundo, mostra que na curva

com contenção e apenas votos negativos a taxa permitida de downloads decresce rapidamente, fazendo Y (por causa de X) convergir a $X_{min} \cong 2$. Com votos divididos, há igualmente uma convergência a um valor baixo, em torno de 6, refletindo um tratamento "conservador" em relação à suspeita de poluição. Observa-se, portanto, a efetividade do método: a taxa de disseminação de conteúdo poluído é bastante reduzida se comparada com o download sem contenção.

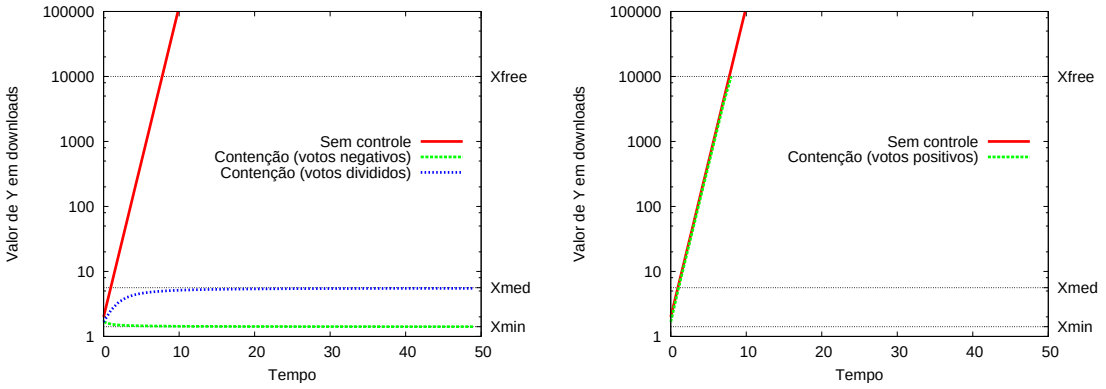


FIGURA 4.5 – Disseminação de versão utilizando função exponencial (γ_b): (a) poluída e (b) correta, com α=1,3 e β=0,024, κ=40 e a=0,1.

Penalidade ocasionada ao download de versões íntegras. A Figura 4.5(b) ilustra essa análise, sob os mesmos parâmetros apresentados anteriormente. A Figura 4.5(b) mostra que não há contenção indesejável no esquema proposto, na medida em que o valor de Y acompanha a velocidade máxima de disseminação, ou seja, sem contenção. Isso é explicado pelo fato que X, não representado na figura, permanece sempre acima de Y, fazendo com a velocidade seja limitada apenas por δ. Note-se que X atinge o patamar X_{free}, quando quaisquer restrições a downloads são suspensas. Pelo exposto, observa-se que não há impacto à disseminação de conteúdo correto no modelo proposto a partir do uso de γ_b.

Método de contenção aplicado em ambientes dinâmicos. Objetivando entender o comportamento da função, foi realizada uma extensa análise com diferentes alternativas aos parâmetros (α, β, δ, κ) nos cenários possíveis. A Figura 4.6 apresenta o mesmo caso da Figura 4.3 para fins de comparação: N-D-P-N, ilustrando X, Y e $E[\omega]$. Em T_1, a inserção de votos negativos mantém X e Y em X_{min}. Na inserção de votos divididos, em T_2, ocorre a igualmente a convergência de X e Y a X_{med}. Em T_3, a inserção de votos positivos eleva rapidamente o comportamento de X e Y a X_{free}. Após isso, T_4 apresenta a queda de X, fazendo a sua curva decair a X_{min}, devido à inserção de votos negativos. Apesar de refletir

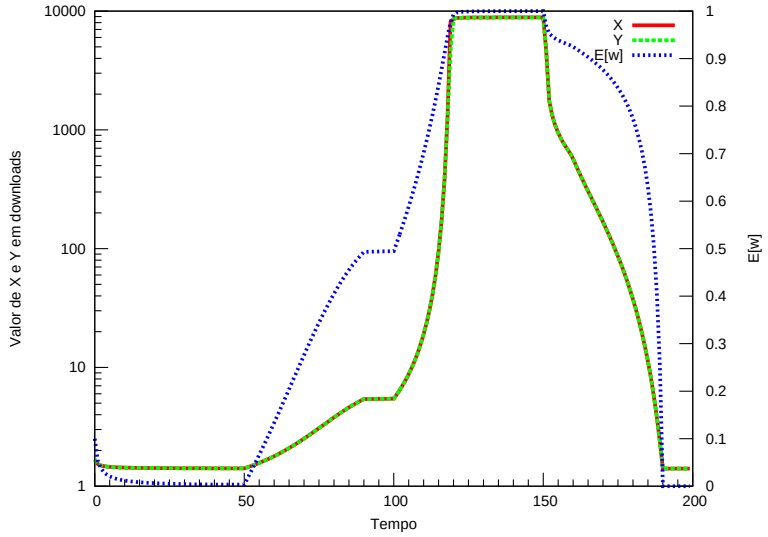


FIGURA 4.6 – Contenção em cenário N-D-P-N utilizando função exponencial (γ_b), com α=1,3 e β=0,024, κ=40 e a=0,1.

a mesma seqüência de comportamentos da Figura 4.3, deve-se observar que o $E[\omega]$ possui comportamento diferente já que a reputação se dá pelo número de votos emitidos, que é diferente neste experimento devido à diferença de realimentação a r e a s inserida por Y.

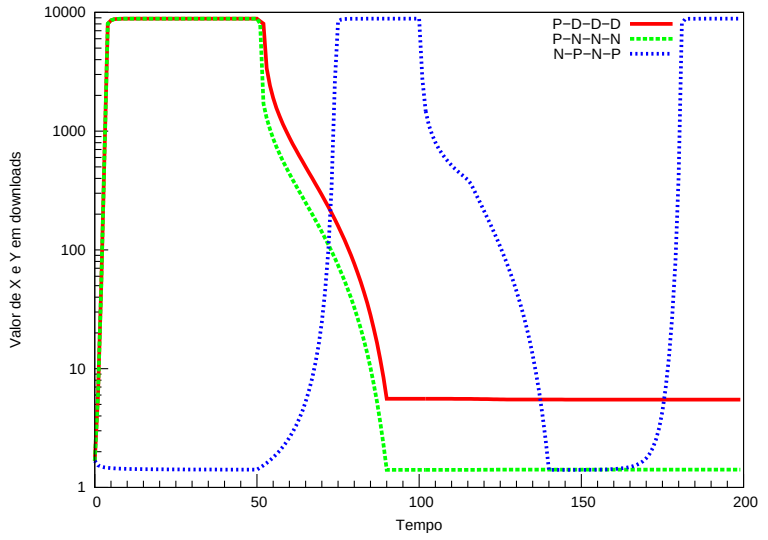


FIGURA 4.7 – Contenção em cenários dinâmicos utilizando função exponencial (γ_b), com α=1,3 e β=0,024, κ=40 e a=0,1.

A Figura 4.7 apresenta outros casos possíveis. Em N-P-N-P, é ilustrada a capacidade de o método elevar a taxa permitida de downloads (T_2) com o recebimento de votos positivos, mesmo após um período de mesmo tamanho com

votos negativos (T_1). Já quando o sistema passa a receber grande quantidade de votos negativos (T_3), há uma queda na taxa de downloads permitidos. Em P-D-D-D, verifica-se a convergência gradual a $X_{med} = 6$, gerada pela emissão de votos divididos em três períodos (T_2, T_3 e T_4), mesmo após grande quantidade de votos positivos. Da mesma forma, ilustra-se a convergência a X_{min} em P-N-N-N.

Avaliação geral de γ_b. A função atinge os limites X_{free} e X_{min}, adequando-se à especificação do modelo. Sua convergência média refere-se a um valor relativamente baixo, o que atribui uma tendência conservadora à essa função neste quesito. A contenção de poluição, conforme demonstrado nos experimentos, é realizada de maneira eficiente, ao passo que o impacto à disseminação de cópias corretas é baixo. A definição de X_{min} e X_{free}, entretanto, não é realizada de forma direta na função, levando à necessidade de parametrizá-los a partir de α e β.

4.2.3 Considerações Gerais

Na comparação entre γ_a e γ_b verifica-se que ambas possuem vantagens e desvantagens entre si. Por exemplo, γ_a utiliza como parâmetros X_{min} e X_{free}, representando diretamente os seus limites superior e inferior, o que não ocorre em γ_b. γ_a, por conseguinte, possui tendência otimista na definição de um caso (de ataque) em que há tanto votos positivos como negativos, como ilustrado por X_{med}. γ_b atribui um valor mais baixo a X_{med}, ilustrando sua tendência conservadora – o que pode ser mais adequado a determinados ambientes. Ressalta-se, entretanto, que nenhuma das funções estudadas pode ser considerada perfeita. A análise de novas funções e de outros modelos matemáticos (como por exemplo Sistemas Dinâmicos) será aprofundada no prosseguimento dessa pesquisa. A metodologia expressa na seção é, entretanto, considerada a base para os trabalhos futuros nessa linha.

Capítulo 5

Contenção de Poluição em Ambiente Distribuído

Este capítulo trata do problema da contenção distribuída de poluição, apresentando alternativas que estendem o método apresentado no Capítulo 4. Ao contrário de contenção em um ambiente ideal, torna-se necessário lidar com os desafios de sistemas distribuídos, intrínsecos a redes P2P, como baixo acoplamento entre os pares e atrasos imprevisíveis de comunicação e execução. Assim, pela impossibilidade de trabalhar com os valores precisos de X e Y em um ambiente descentralizado, seus valores estimados devem ser determinados através de um protocolo distribuído. Naturalmente, quanto mais precisa a estimativa, melhor.

Existem diferentes abordagens para controlar de forma distribuída os valores de X e de Y, bem como garantir $Y \leq X$. Guiando a escolha, há premissas ambientais (por exemplo, a estruturação da rede) e operacionais (expectativa dos usuários em relação à proteção contra versões corrompidas). Dessa forma, é necessário elencar os critérios básicos que influenciam as soluções para a contenção distribuída de poluição. Abaixo são citadas as principais decisões de projeto a serem consideradas:

- **segmentação** (Global, Segmentada): relativo à divisão da rede P2P em segmentos menores, que podem ser autônomos ou cooperar (segmentos podem ser ainda subdivididos, formando uma hierarquia, mas esta abordagem não é explorada no trabalho);
- **centralização** (Centralizada, Descentralizada): relativo ao grau de centralização, há um claro *trade-off* entre simplicidade/controle e tolerância a falhas/escalabilidade;
- **estruturação** (Estruturada, Não-estruturada): se o método de contenção

depende de uma rede P2P estruturada para funcionar eficientemente;

- **autonomia** (Autônoma, Dependente): concerne a autonomia de pares em cumprir ou não as decisões do método de contenção – em soluções autônomas, o método de contenção é apenas uma recomendação.

Considerando a combinação dessas decisões de projeto, são possíveis diferentes métodos de contenção distribuída. Neste trabalho, foram exploradas quatro alternativas principais: GCED, GDNA, SCED e SCND. Nesse contexto, uma solução distribuída pode ser basicamente descrita através do conjunto de operações fundamentais a serem executadas: (a) obtenção da estimativa de X; (b) obtenção da estimativa de Y; (c) autorização ou negação de download de acordo com $Y \leq X$; (d) incremento de Y perante a autorização de um novo download; (e) decremento de Y e recálculo de X ao término de um download.

Adicionalmente, T_a denota o tempo para a decisão sobre a autorização do download; T_d, o tempo de realização do download e T_v, o tempo da verificação da integridade da versão e envio de voto. `RTT` representa *round-trip time*. Note-se que em transferências típicas de sistemas P2P, usualmente $T_d \gg$ `RTT`, sendo T_d também dominante (minutos ou horas) sobre os demais tempos (milisegundos). Exemplificando, para um arquivo de 10MB, se `RTT` é 200 ms e a largura de banda dedicada à transferência da versão é 128Kbs, então T_d será pelo menos 625s ou 10min25s. O T_a, nesse caso, representa aproximadamente 0,032% do tempo total.

Por fim, juntamente com cada abordagem são discutidas questões essenciais de projeto, como tolerância a falhas, escalabilidade, vulnerabilidades potenciais, precisão e sobrecarga. Considera-se a sobrecarga como a latência induzida por uma abordagem, denotada como l, e definida conforme a Equação 5.1. Assim, l fornecerá o percentual que a sobrecarga de tempo inserida ($T_a + T_v$) representa no tempo total de download ($T_a + T_d + T_v$). Obviamente, os tempos T_a e T_v são dependentes da abordagem distribuída.

$$l = \frac{T_a + T_v}{T_a + T_d + T_v} \tag{5.1}$$

5.1 GCED: Global, Centralizada, Estruturada, Dependente

GCED refere-se à abordagem mais simples, baseando-se em um "gerente de contenção". Um par solicita autorização para download de uma versão ao gerente via mensagem `REQUEST`; o gerente verifica o valor corrente de X, de Y, e com base em tais valores, decide sobre a autorização. Se $Y \geq X$, a solicitação é negada, e o

gerente responde ao par, que deve tentar de novo posteriormente. Caso contrário, o gerente faz $Y \leftarrow Y + 1$ e responde ao par com uma mensagem GRANT, autorizando a realização do download. A seguir, o par realiza o download enviando a mensagem RETR. Ao finalizar, verifica a integridade do arquivo e envia uma mensagem de voto VOTE, positivo ou negativo ao gerente, de acordo com o resultado. O gerente recebe o voto, atualizando r ou s, alterando X, e então $Y \leftarrow Y - 1$. Este esquema está ilustrado na Figura 5.1.

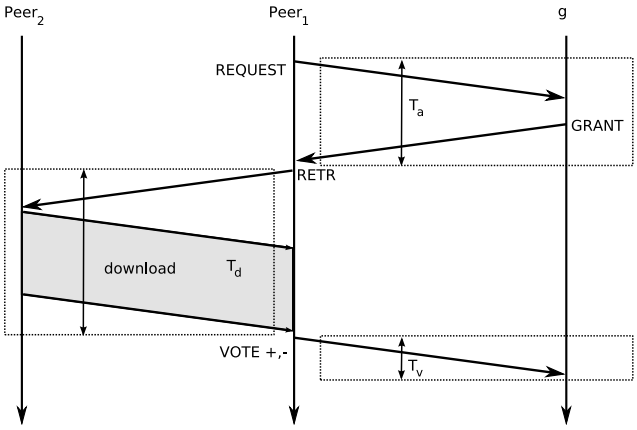


FIGURA 5.1 – Diagrama de tempo correspondente ao esquema GCED

A operação do gerente pode ser definida sobre duas modalidades principais: gerente de contenção global ou gerente de contenção de versão. A primeira modalidade refere-se a uma abordagem pragmática e limitada, onde há apenas um gerente que atende à toda a rede P2P. A segunda denota a especificação de um método onde a contenção é realizada por pares específicos, responsáveis pela contenção de cada versão independentemente. Embora o método GCED adote uma rede estruturada como base, ele não depende de tal facilidade. Uma DHT permite que o gerente de contenção de uma dada versão seja encontrado de forma eficiente; do contrário, uma inundação é necessária, tornando possível encontrar o gerente responsável pela versão.

Naturalmente, um gerente de contenção torna-se um ponto central de falhas e de ataques, e um candidato à ocorrência de gargalos na comunicação. Para manter estimativas corretas o gerente deve ser capaz de detectar downloads não finalizados ou falhas – o que agrega certa complexidade ao método. Essa questão poderia ser implementada a partir de mensagens periódicas de status (*keep alive*) ou mesmo designar que o mantenedor da versão informe o gerente de contenção sobre o encerramento de um download.

Como a gerência é centralizada, as estimativas são tão precisas quanto atuais

e corretos os valores de r e s. Devido ao controle central, considera-se que não há diferenças entre a estimativa e o seu valor real. Considera-se $T_a = RTT$, pois refere-se ao tempo da mensagem de autorização chegar até o gerente e ser respondida, onde o tempo de processamento da requisição é negligível, pois a obtenção de X é instantânea. $T_v = \frac{RTT}{2}$, pois refere-se ao tempo de o requisitante encaminhar o voto ao gerente.

5.2 GDNA: Global, Descentralizada, Não-estruturada, Autônoma

Ao contrário do anterior, este método é totalmente descentralizado: todos os pares possuem o mesmo papel. Cada par é autônomo ao decidir se pode ou não realizar um download. Sua decisão é baseada na execução *correta* do algoritmo proposto, que inclui a determinação de estimativas para X e Y. Em uma rede P2P usual, não é viável empregar protocolos distribuídos, como por exemplo os de consenso [Veríssimo and Rodrigues, 2001]. No esquema proposto, antes de requisitar o download, um par deve consultar os demais pares na rede para obter estimativas atuais de X e Y. Cada par P_i armazena uma tripla de valores binários (y_i, r_i, s_i) por versão, que representa, respectivamente, se o par está realizando o download no momento, se há um voto positivo, e se há um voto negativo em relação à determinada versão. A consulta aos valores em cada par é realizada através de um algoritmo sistólico sobre a rede de sobreposição, descrito a seguir.

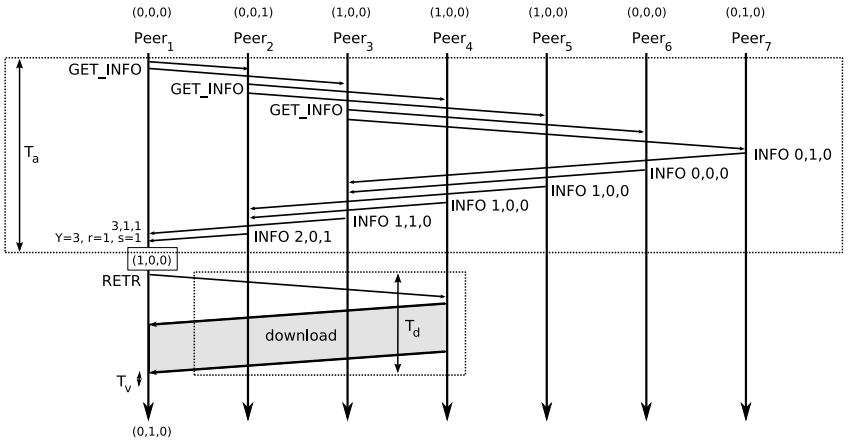


FIGURA 5.2 – Diagrama de tempo correspondente ao esquema GDNA

Além da tripla, pares mantém uma variável booleana que indica se estão marcados ou não. Inicialmente, todos os (demais) pares estão desmarcados. A rede

é inundada com mensagens de coleta de informações sobre uma versão, processo que inicia com o envio da mensagem GET_INFO pelo par (que deseja o download) aos seus vizinhos; cada par que ainda não recebeu a marca, é marcado e encaminha a mensagem aos seus vizinhos a não ser para a ligação pela qual a mensagem chegou. Pares que recebem uma mensagem GET_INFO, mais cedo ou mais tarde, retornam ao emissor uma mensagem INFO (y, r, s) contendo sua tripla (y, r, s). Nestes dois últimos casos, assume-se que o par tenha completado o download e verificado a cópia. Pares marcados que recebem uma mensagem GET_INFO retornam imediatamente uma mensagem INFO com $(0, 0, 0)$; os não marcados encaminham a mensagem a cada um de seus filhos, consolidando as respostas recebidas através de uma *soma* dos valores individuais da tripla: para um par com n vizinhos, $(y, r, s) = (\sum_{i=0}^{n} y_i, \sum_{i=0}^{n} r_i, \sum_{i=0}^{n} s_i,)$, onde y_i, r_i, s_i representam os valores recebidos do vizinho i. A mensagem com os valores consolidados é enviada ao par que encaminhou a primeira mensagem GET_INFO. De maneira recursiva, mensagens INFO convergem para o par responsável pela solicitação, permitindo ao mesmo obter ao final uma tripla (y, r, s) que representa estimativas do somatório de downloads em execução, do número de votos positivos e do número de votos negativos. Com base nas informações recebidas, o solicitante inicial computa localmente o valor de X – com base na reputação da versão – através da função γ. A Figura 5.2 ilustra essa abordagem para uma rede com 7 pares.

Considerando que as operações de download ocorrem em um arcabouço temporal de ordens de magnitude superior ao RTT, assume-se que a abordagem proposta produza estimativas relativamente atualizadas. Para ilustrar esse ponto, considere-se o caso de uma consulta a uma árvore binária. A altura da árvore corresponde ao número de saltos que devem ser dados até encontrar o par mais distante, e o dobro para o caminho de ida e volta. O tempo para que tal consulta seja completada, definindo o valor de T_a, é da ordem de $\log_2(N+1) \times RTT$, onde N representa o número de pares na rede. Exemplificando, uma consulta em uma rede com 500.000 pares teria T_a=18,2 s, para um RTT de 200 ms e um tempo de download superior a 10 min. Apesar desse atraso ser comparativamente aceitável (apenas 2,9%), sabe-se que inundações apresentam baixa escalabilidade, sendo proibitivos o tráfego e o processamento gerados.

A solução usualmente empregada para aumentar a escalabilidade em tais sistemas, tal como [Gnutella, 2007], é limitar o escopo das mensagens (seu tempo de vida ou *horizonte*) e o número máximo de vizinhos para os quais um par encaminha uma mensagem. No entanto, por não consultar todos os pares, limitar o escopo da inundação pode *subestimar* os valores de Y, r e s, potencialmente diminuindo a

precisão das estimativas.

A abordagem não possui elementos centralizadores, bem como as características negativas a eles associadas. Para obter a autorização para efetuar o download, um par solicita referências de outros pares sobre a versão (votos). Assim, as estimativas obtidas podem não ser exatas, afetadas diretamente pela alta transiência de pares, horizonte limitado e mensagens em trânsito. Outro ponto negativo é que os participantes podem optar por não seguir o protocolo especificado, ignorando a decisão local do sistema de contenção, porém, com risco de obtenção de conteúdo poluído.

A escalabilidade da abordagem é comprometida pela necessidade de inundar a rede (no algoritmo sistólico), devido à quantidade de mensagens. Para fins de avaliação, o protocolo é considerado sobre uma árvore de pesquisa completa de grau gr. Nesse contexto, o tempo de autorização é definido por $T_a = (\log_{gr} N) \times RTT$, onde N refere-se ao número de nodos que recebem a solicitação e emitem o voto, conforme o horizonte h. Ao término do download os nodos não encaminham o voto sobre a versão, apenas o armazenam localmente, portanto, $T_v = 0$.

A ação de pares maliciosos pode afetar o método, na medida em que é possível a esses pares adulterar as informações consolidadas recebidas na fase de contração do algoritmo. Como alternativa, pode-se adaptar o método de forma que todos os pares consultados encaminhem seus votos diretamente ao requisitante, elevando, em contrapartida, a sobrecarga de mensagens na rede. Além disso, para evitar postergação indefinida no protocolo, deve ser estipulado determinado tempo de expiração às requisições encaminhadas aos filhos por determinado par. O sistema deve ser capaz de prosseguir, mesmo que determinados pares não retornem as informações solicitadas.

5.3 SCED: Segmentada, Centralizada, Estruturada, Dependente

Essa abordagem, para controle de contenção distribuída, baseia-se na divisão de uma rede P2P estruturada (como Chord [Stoica et al., 2003]) em segmentos. A rede, com 2^τ identificadores de pares, é particionada em 2^ϕ segmentos, cada um com $2^{\tau-\phi}$ identificadores. O valor de X é adaptado de acordo com o tamanho do segmento, para representar uma fração do global. Refletindo isso, o cômputo de Y_i é obtido a partir do somatório dos downloads ocorrendo no segmento. Portanto, $X = \sum_{i=1}^{2^\phi} X_i$ e $Y = \sum_{i=1}^{2^\phi} Y_i$, onde 2^ϕ é o número de segmentos.

Tal como a primeira abordagem proposta, a decisão de autorização de downloads é tomada por um gerente de contenção[1]. Entretanto, diferentemente, existe um gerente por segmento. Cada segmento é criado dinamicamente em função das versões existentes. Ou seja, a segmentação é virtual e relativa à cada versão do sistema. Com isso, cada gerente é responsável pelo controle de contenção de apenas uma versão. Em cada segmento S_i, um gerente g_i é autônomo para controlar o valor X_i correspondente à versão sendo obtida no segmento e o número corrente de downloads no mesmo, Y_i. O gerente g_i é escolhido de forma determinística, como, por exemplo, o último par do segmento. Baseado nos valores de X_i e Y_i locais, o gerente autoriza ou não os downloads.

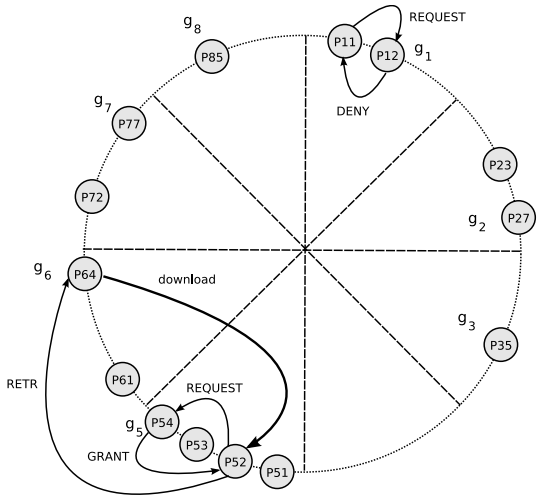


FIGURA 5.3 – Exemplo ilustrando arquitetura do esquema SCED

Assume-se a título de exemplo que $\tau = 128$ e $\phi = 3$. A Figura 5.3 ilustra o funcionamento do esquema: par $P11$ requisita autorização de download ao seu gerente em $P12$, que o nega (as mensagens são equivalentes às abordagens anteriores). Já o par $P52$ solicita autorização ao seu gerente, g_5 (em $P54$), e recebe uma resposta positiva. O par então solicita o download a $P64$, que detém uma cópia da versão em questão. Nota-se que o exemplo ilustra um caso onde a versão a ser obtida se encontra fora do segmento em que se encontra o par requisitante.

Os valores de τ e ϕ são fundamentais para o bom funcionamento do esquema. A medida que ϕ se aproxima de τ, o número de segmentos aumenta e seu tamanho diminui. Em um extremo, se $\phi = \tau$, existem 2^τ segmentos com um (ou nenhum) par cada um. Neste caso, como $X_i \leq X_{min}$ e $X_{min} \geq 1$, isto é o equivalente à inexistência de controle, pois todo par está autorizado a fazer um download. No outro, se $\phi = 0$,

[1]gerentes podem atuar tanto de forma isolada como cooperativa; neste trabalho, como um primeiro passo, é considerada apenas a alternativa mais simples, sem cooperação entre gerentes.

existe apenas um segmento abrigando todos os pares, caso específico em que SCED é idêntico ao GCED.

Naturalmente, a existência de múltiplos gerentes diminui a dependência do sistema por uma entidade única, central. O fato deste esquema empregar um gerente por segmento reduz o grau de centralização, entretanto, não elimina as vulnerabilidades de pontos centrais. Essas características prejudicam diretamente tolerância a falhas, escalabilidade e segurança do sistema. Como um gerente é responsável pelas decisões de contenção de apenas um segmento, em caso de falha, isso afeta apenas o segmento em questão. Além disso, redes P2P estruturadas, tradicionalmente, possuem métodos para realocação de identificadores, necessários à acomodação da rede devido à entrada e saída de pares; tal pode ser usado para aumentar a tolerância a falhas do SCED.

O fato de o gerente ser malicioso tende a prejudicar o funcionamento do método; por outro lado, trata-se de apenas um segmento, o que agrega robustez ao método frente à abordagem de gerente global. O número de segmentos (2^{ϕ}) afeta diretamente a precisão das estimativas do método: quanto maior for o número de segmentos, menos precisas serão as estimativas, obtidas por uma visão local com um número menor de votantes. O intercâmbio de informações entre os gerentes de cada segmento pode aumentar a precisão na estimativa global de X. A sobrecarga ocasionada em cada segmento depende de T_a, que no pior caso corresponde a contactar (usando a *finger table*) um gerente mais distante no segmento de tamanho 2^{ϕ}, e portanto $T_a = \phi \times \frac{RTT}{2} + \frac{RTT}{2}$. O tempo de verificação e voto, que se refere à mensagem encaminhada ao gerente, é obtido por é $T_v = \frac{RTT}{2}$.

5.4 SCND: Segmentada, Centralizada, Não-estruturada, Dependente

Em uma rede P2P não-estruturada, garantir a segmentação de forma adequada não é uma tarefa trivial. A criação de segmentos é dificultada pelo desconhecimento do tamanho da rede pelos pares e a alta transiência da rede. A alternativa proposta é utilizar a organização criada por super-pares para a criação de segmentos em redes não-estruturadas. Assim, pares ligados a um mesmo super-par pertencem ao mesmo segmento.

O papel de gerente de contenção é mantido pelo super-par do segmento. A rede P2P, assim, será dividida em U segmentos, onde U é o número de super-pares. A Figura 5.4 ilustra o funcionamento desse método. Um par P_i, que deseja efetuar o

download, encaminha uma requisição ao super-par g_i (REQUEST). Em g_i são mantidas as estimativas locais X_i e Y_i. Ao avaliar $X_i \leq Y_i$, é realizada a decisão de autorizar (GRANT) ou não (DENY) o download. Ao aprovar a requisição, g_i deverá incrementar Y_i. Por fim, P_i encaminha um voto positivo ou negativo ao gerente, que atualizará a sua estimativa X_i.

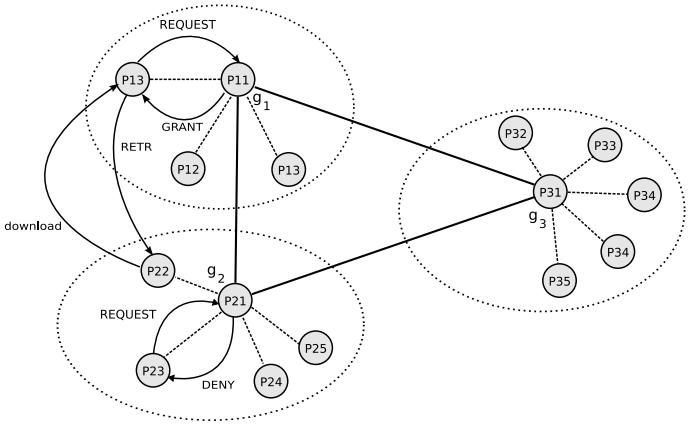


FIGURA 5.4 – Exemplo ilustrando arquitetura do esquema SCND

Uma possibilidade alternativa refere-se a promover a cooperação entre os segmentos através dos seus super-pares. Como os mesmos estão interligados, os valores pertinentes a cada segmento (números de votos e downloads correntes) poderiam ser trocados entre os super-pares, possibilitando estimativas globais. Com base nas estimativas consolidadas de r e s, os super-pares teriam condições de obter a estimativa X, derivando o seu limite global a partir de $X_i = \frac{X}{U}$, permitindo assim um limiar mais preciso. O número corrente de downloads no segmento, no entanto, seguiria a estratégia local, já que manter Y de forma global e assegurar que $Y \leq X$ demandaria maior acoplamento (comunicação e sincronização) entre os pares.

A existência de um gerente por segmento reduz o grau de centralização, conforme já comentado, não havendo ponto central ao mecanismo de contenção. Tolerância a falhas vale-se de mecanismos de contingência já existentes em redes não-estruturadas semi-centralizadas, mitigando falhas do gerente (super-par). Além disso, ataques ou falhas de um gerente não prejudicam a rede como um todo (dada a redundância de conexões), agregando também robustez à abordagem. Entretanto, como os segmentos podem estar desbalanceados, o comprometimento de um gerente pode afetar um número arbitrariamente grande de participantes. Além disso, o super-par que faz o papel de gerente a todas as requisições do segmento pode tornar-se um possível gargalo.

As estimativas X_i e Y_i são precisas para cada segmento S_i. Apesar de X_i ser

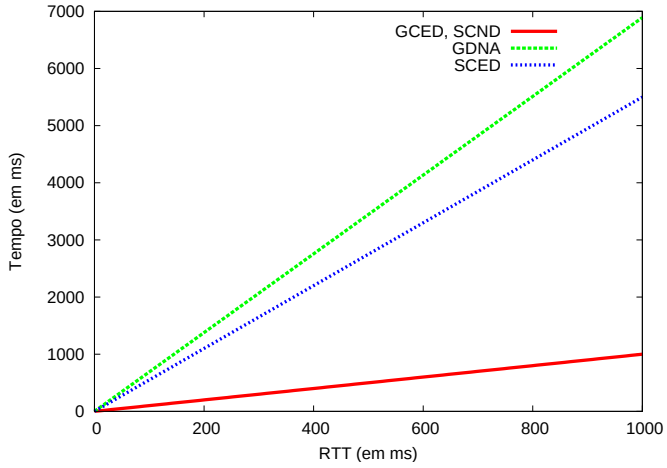


FIGURA 5.5 – Comparação de tempo de autorização (T_a) entre as abordagens de contenção distribuída de poluição

preciso, a reputação é obtida a partir das experiências de uma fração dos pares (no próprio segmento), levando a uma maior incerteza sobre a reputação de tal conteúdo. Quanto à Y_i, o gerente controla precisamente downloads iniciados e finalizados no próprio segmento. Embora as estimativas de X_i e Y_i sejam adequadas em cada segmento S_i, a medida que se segmenta o X e Y globais em X e Y menores, permite-se que mais ou menos pares executem download do que o desejado globalmente. A sobrecarga para cada segmento é definida de maneira análoga à GCED: $T_a = RTT$ e $T_v = \frac{RTT}{2}$, possibilitando a obtenção de l conforme a Equação 5.1.

5.5 Avaliação das Abordagens Distribuídas

Essa seção apresenta uma avaliação das abordagens distribuídas de contenção de poluição. Inicialmente, é realizada uma comparação entre T_v e T_a, conforme valores já apresentados na seção anterior. A sobrecarga geral (Equação 5.1) de cada método é comparada às demais. Nessa avaliação foi assumido um tempo de download igual a 10 minutos ($T_d = 600.000ms$) e uma rede com 2^{16} pares ($N = 65.536$). Além disso, para GDNA foi utilizado $gr = 5$ e para SCED, $\phi = 10$.

Conforme ilustrado na Figura 5.5, o tempo de autorização é negligível se comparado ao tempo total de download, em todas as abordagens. Como esperado, as estratégias GCED e SCND obtiveram o menor tempo de autorização, por referirem-se a esquemas centralizados; entretanto, realisticamente, tais soluções seriam inadequadas para sistemas com o tamanho considerado. A Figura 5.5 con-

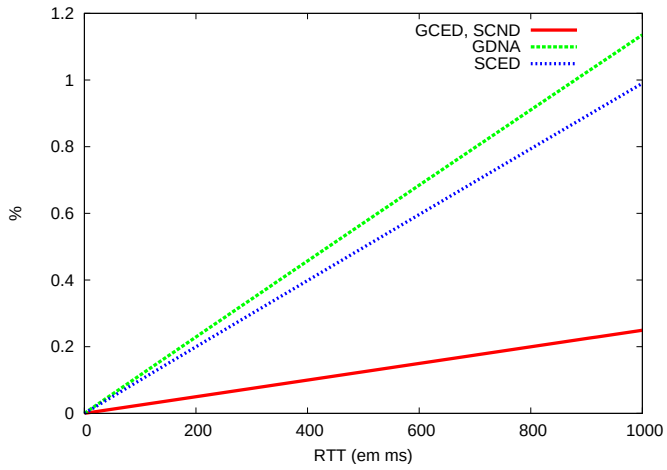


FIGURA 5.6 – Comparação de sobrecarga l das abordagens de contenção distribuída de poluição

firma a baixa sobrecarga esperada dos métodos propostos, com novamente as estratégias GCED e SCND possuindo os menores valores.

5.6 Limitações e Desafios

Esta seção finaliza o capítulo discutindo o método de contenção distribuída em termos gerais, enfatizando suas limitações e principais desafios. Primeiramente, quatro modelos básicos de solução foram propostos. No prosseguimento da pesquisa, considera-se necessário avaliá-los a partir de modelos analíticos ou de simulações. A condução dessa análise detalhada das abordagens propostas (e das outras soluções possíveis, conforme as decisões de projeto apresentadas) ilustrará a sua adequação ao problema proposto. Considera-se também que a metodologia de avaliação apresentada no Capítulo 4 possa ser empregada neste sentido.

A análise de sobrecarga considera T_v sob um comportamento simplificado, onde a inspeção do conteúdo e emissão do voto ocorrem imediatamente após o término do download. Apesar de não ilustrado graficamente, sabe-se que T_v na prática poderá ter uma influência bem mais forte no sistema, pois parte dos usuários tipicamente não verifica o download imediatamente (e portanto nem votaria logo após). Conforme [Lee et al., 2006], o tempo de inspeção de download é regido por uma distribuição bimodal: a maioria dos usuários inspeciona o conteúdo logo após o término do download ou muito tempo após o seu término (mais de 12 horas).

Além disso, para a evolução do sistema, os votos emitidos pelos usuários são imprescindíveis. Claramente, é necessário que o sistema possua mecanismos de

incentivo à emissão de votos. Havendo um sistema de gerência de reputação de pares, a reputação de um par poderia ser afetada pela integridade dos votos emitidos e os atrasos na emissão de votos. Pode-se também manter outras informações sobre reputação das versões no sistema, por exemplo, avaliando a sua qualidade. Com isso, um método de incentivo poderia garantir (ou priorizar) o acesso às versões de maior qualidade aos pares que possuam as maiores reputações.

Outro ponto a ser ressaltado refere-se às vulnerabilidades inseridas, que devem ser tratadas para a implementação do método de contenção em redes P2P. Ataques de traição são tratados diretamente no modelo, ao considerar maior peso aos votos mais recentes. Outra técnica de subversão a ser tratada refere-se ao falso testemunho: a emissão de votos incorretos para elevar ou reduzir a taxa de disseminação permitida à determinada versão. Neste contexto, três garantias principais se tornam necessárias: (a) que pares só tenham o seu voto considerado se de fato comprovarem que o download foi realizado, (b) que o voto não foi forjado (enviado em nome de um outro par, que efetuou o download) e (c) que cada par tenha o seu voto considerado apenas uma vez. Este último ítem poderia ser mitigado com a existência de controle adicional na entidade que está consolidando os votos, garantindo que um mesmo voto não seja computado mais de uma vez. A própria literatura de sistemas de reputação propõe medidas às demais garantias.

No XREP (Seção 3.2.1) é proposto que um subconjunto dos votos emitidos seja confirmado junto aos emissores. Este método, porém, apesar de tratar do problema dos votos forjados, não resolve o caso em que um atacante tenha emitido um voto sem ter realizado o download. O TrustGuard (Seção 3.2.4) propõe que o voto somente seja considerado caso o emissor encaminhe juntamente provas de que de fato a transação foi realizada. Isso poderia ser obtido, por exemplo, por métodos baseados em assinatura digital. Apesar de impedir que um par vote incorretamente em uma entidade com a qual não interagiu (par ou versão), não evita que um nodo encaminhe votos desonestos com quem interagiu. Este último ponto pode ser controlado também pelo mecanismo proposto no EigenTrust (Seção 3.2.2), onde o voto é pesado de acordo com a reputação do par emissor. Tal característica minimiza a ação de nodos maliciosos que, com baixa reputação, não terão as suas opiniões fortemente consideradas.

Apesar de a contenção especificada sob a Lógica Subjetiva, outros modelos de reputação poderiam ser utilizados como base. Com isso, redes P2P poderiam implementar o método de contenção, usufruindo de seus sistemas de reputação de versão já existentes. Neste sentido, o mecanismo de reputação deve gerar um escore de confiança entre $[0, 1]$ (ou que seja possível adequá-lo neste intervalo).

Capítulo 6

Conclusão

Compartilhamento de arquivos P2P é uma das principais aplicações, senão a principal, da Internet na atualidade. Esses sistemas têm sido alvo de ataques massivos de poluição de conteúdo, o que poderá vir a comprometer fortemente a sua efetividade. Uma série de trabalhos tem analisado este problema, dos quais boa parte tem se concentrado em mapear e delimitar o ataque, criando o embasamento necessário a propostas de solução. Entretanto, verifica-se ainda a ausência de artigos que proponham métodos robustos e eficazes para o tratamento de ataques de poluição.

A presente dissertação propõe e analisa métodos para contenção de poluição baseados na limitação do número corrente de downloads. A velocidade de disseminação é controlada de acordo com a confiança na integridade da versão, segundo votos emitidos e uma função γ para transformar reputação em número de downloads permitidos. Inicialmente, o método proposto é descrito, incluindo duas funções γ. O método é avaliado em um ambiente ideal, onde foi possível mostrar a eficiência teórica do método na contenção de poluição e a baixa penalidade à disseminação de versões corretas. Posteriormente, foram identificadas as principais questões de projeto, e então propostos quatro métodos básicos de contenção distribuída. Os métodos são comparados à luz de requisitos importantes como escalabilidade e sobrecarga.

O método proposto tem como base um sistema de reputação de versões. Neste trabalho, foi considerada a Lógica Subjetiva, conforme [Josang et al., 2006], por ser um modelo de reputação bastante maduro. Porém, acredita-se que o fundamento de contenção proposto no trabalho possa também ser utilizado junto a outros sistemas de reputação, desde que os mesmos informem um escore numérico de reputação das versões na rede P2P. Isso também garante a implantação gradual do método de contenção a sistemas P2P que já mantenham a reputação de versões.

A principal contribuição desta dissertação é a proposta inédita de contenção de poluição, a partir da limitação de downloads de acordo com a reputação de versões. A metodologia de avaliação adotada, bem como as decisões de projeto de soluções distribuídas, referem-se também a avanços que podem ser tomados como base a outros trabalhos. Sabe-se, entretanto, que a solução para ataques de poluição é uma tarefa complexa e desafiadora; o trabalho não esgota o assunto, mas aponta uma nova possibilidade no espaço de soluções contra poluição de conteúdo em P2P.

Como trabalhos futuros, novos estudos estão sendo conduzidos no sentido de eliminar as premissas simplificatórias adotadas e determinar a resiliência do método à ação de atacantes. Uma outra direção a ser seguida refere-se a avaliar a combinação do método de contenção proposto com sistema de reputação de pares. Além de aperfeiçoar potencialmente a resistência a ataques, a estratégia permitiria identificar os pares poluidores na rede bem como explorar métodos de incentivo à emissão de votos.

Outros modelos básicos de solução distribuída podem ser desenvolvidos e avaliados, a partir das decisões de projeto apresentadas. Deverá também ser conduzida a avaliação da eficácia, sobrecarga de latência e de comunicação dos métodos de contenção distribuída propostos. Por fim, também considera-se a instanciação e avaliação do método de contenção de poluição aplicado a uma rede P2P existente.

Bibliografia

[Andrade et al., 2003] Andrade, N., Cirne, W., Brasileiro, F., and Roisenberg, P. (2003). Ourgrid: An approach to easily assemble grids with equitable resource sharing. In *Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing*, Seattle, WA, USA.

[Barbera et al., 2005] Barbera, M., Lombardo, A., Schembra, G., and Tribastone, M. (2005). A markov model of a freerider in a bittorrent P2P network. In *IEEE Global Telecommunications Conference (GLOBECOM '05)*, volume 2, pages 985–989, St. Louis, MO, USA.

[Barcellos and Gaspary, 2006] Barcellos, M. P. and Gaspary, L. P. (2006). *Segurança em Redes P2P: Princípios, Tecnologias e Desafios*, volume 1, pages 211–260. SBC.

[BBCP2P, 2007] BBCP2P (2007). BBC moves to file-sharing sites. http://news.bbc.co.uk/2/hi/technology/6194929.stm.

[BitTorrent, 2007] BitTorrent (2007). BitTorrent website. http://www.bittorrent.com/.

[Bjurefors et al., 2004] Bjurefors, F., Larzon, L. ., and Gold, R. (2004). Performance of pastry in a heterogeneous system. In *4th International Conference on Peer-to-Peer Computing (P2P'04)*, pages 278–279, Zurich, Switzerland.

[Chang et al., 2005] Chang, E., Dillon, T. S., and Hussain, F. K. (2005). Trust and reputation relationships in service-oriented environments. In *3rd International Conference on Information Technology and Applications (ICITA 2005)*, volume 1, pages 4–14, Sydney, Australia.

[Chaum, 1981] Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. In *Communications of the ACM*, volume 24, pages 84–90, New York, NY, USA. ACM Press.

[Chen et al., 2002] Chen, B., Gil, T. M., Morris, R., and Muthitacharoen, A. (2002). Ivy: Read-write peer-to-peer filesystem. In *5th Symposium on Operating Systems Design and Implementation*, Boston, Massachusetts, USA.

[Christin et al., 2005] Christin, N., Weigend, A. S., and Chuang, J. (2005). Content availability, pollution and poisoning in file sharing peer-to-peer networks. In *6th ACM conference on Electronic commerce (EC '05)*, pages 68–77, New York, NY, USA. ACM Press.

[Costa et al., 2006] Costa, C., Soares, V., Benevenuto, F., Vasconcelos, M., Almeida, J., Almeida, V., and Mowbray, M. (2006). Disseminação de conteúdo poluído em redes P2P. In *XXIV Simpósio Brasileiro de Redes de Computadores*, Curitiba, PR, Brasil.

[Damiani et al., 2002] Damiani, E., di Vimercati, D. C., Paraboschi, S., Samarati, P., and Violante, F. (2002). A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 207–216, New York, NY, USA. ACM Press.

[Dingledine et al., 2001] Dingledine, R., Freedman, M. J., and Molnar, D. (2001). The free haven project: distributed anonymous storage service. In *International workshop on Designing privacy enhancing technologies*, pages 67–95, New York, NY, USA. Springer-Verlag New York, Inc.

[Dingledine et al., 2004] Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: The second-generation onion router. In *13th USENIX Security Symposium*, San Diego, CA, USA.

[Douceur, 2002] Douceur, J. R. (2002). The sybil attack. In *1st International Workshop on Peer-to-Peer Systems*, pages 251–260, Cambridge, MA, USA.

[Druschel and Rowstron, 2001] Druschel, P. and Rowstron, A. (2001). Past: A large-scale, persistent peer-to-peer storage utility. In *Eighth Workshop on Hot Topics in Operating Systems (HotOS)*, pages 75–80, Elmau/Oberbayern, Germany.

[eBay, 2007] eBay (2007). eBay website. http://www.ebay.com/.

[ESM, 2007] ESM (2007). End System Multicast website. http://esm.cs.cmu.edu/.

[Genome@home, 2007] Genome@home (2007). Genome@home website. http://genomeathome.stanford.edu/.

[Gkantsidis et al., 2004] Gkantsidis, C., Mihail, M., and Saberi, A. (2004). Random walks in peer-to-peer networks. In *The 23rd Conference of the IEEE Communications Society (INFOCOM 2004)*, volume 1, pages 120–130, Hong Kong.

[Gnutella, 2007] Gnutella (2007). Gnutella website. http://www.gnutella.com/.

[Goldschlag et al., 1999] Goldschlag, D., Reed, M., and Syverson, P. (1999). Onion routing. *Communications of the ACM*, 42(2):39–41.

[GoogleTalk, 2007] GoogleTalk (2007). Google talk website. http://www.google.com/talk/.

[ICQ, 2007] ICQ (2007). ICQ.com website. http://www.icq.com/.

[Jabber, 2007] Jabber (2007). Jabber: Open Instant Messaging. http://www.jabber.org/.

[JetFile, 2007] JetFile (2007). Jetfile. http://www.sics.se/cna/jetfile.html.

[Josang et al., 2006] Josang, A., Hayward, R., and Pope, S. (2006). Trust network analysis with subjective logic. In *ACSC '06: Proceedings of the 29th Australasian Computer Science Conference*, pages 85–94, Darlinghurst, Australia. Australian Computer Society, Inc.

[Kamvar et al., 2003] Kamvar, S. D., Schlosser, M. T., and Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th international conference on World Wide Web (WWW '03)*, pages 640–651, New York, NY, USA. ACM Press.

[KaZaA, 2007] KaZaA (2007). Kazaa.com website. http://www.kazaa.com/.

[Kumar et al., 2006] Kumar, R., Yao, D., Bagchi, A., Ross, K. W., and Rubenstein, D. (2006). Fluid modeling of pollution proliferation in P2P

networks. In *ACM/IFIP SIGMETRICS/Performance 2006*, volume 34, pages 335–346, St. Malo, France.

[Lee et al., 2006] Lee, U., Choiz, M., Choy, J., Sanadidiy, M. Y., and Gerla, M. (2006). Understanding pollution dynamics in P2P file sharing. In *5th International Workshop on Peer-to-Peer Systems (IPTPS'06)*, Santa Babara, CA, USA.

[Leithold, 1986] Leithold, L. (1986). *O Cálculo com Geometria Analítica*. Harbra, São Paulo, Brasil, 2 edition.

[Liang et al., 2005a] Liang, J., Kumar, R., Xi, Y., and Ross, K. W. (2005a). Pollution in P2P file sharing systems. In *The 24th Conference on Computer Communications (INFOCOM 2005)*, volume 2, pages 1174–1185, Miami, FL, USA.

[Liang et al., 2005b] Liang, J., Naoumov, N., and Ross, K. W. (2005b). Efficient blacklisting and pollution-level estimation in P2P file-sharing systems. In *ASIAN INTERNET ENGINEERING CONFERENCE (AINTEC)*, pages 1–21, Bangkok, Thailand.

[Liang et al., 2006] Liang, J., Naoumov, N., and Ross, K. W. (2006). The index poisoning attack in P2P file-sharing systems. In *The 25th Conference on Computer Communications (INFOCOM 2006)*, Barcelona, Spain.

[Lua et al., 2005] Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., and Lim, S. (2005). A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials*, 7(2):72–93.

[Marti and Garcia-Molina, 2006] Marti, S. and Garcia-Molina, H. (2006). Taxonomy of trust: Categorizing P2P reputation systems. *Computer Networks*, 50(4):472–484.

[Maymounkov and Mazieres, 2002] Maymounkov, P. and Mazieres, D. (2002). Kademlia: A peerto -peer information system based on the xor metric. In *International Peer-to-Peer Symposium (IPTPS02)*, Cambridge, MA, USA.

[Milgram, 1967] Milgram (1967). The small world problem. *Psychology today*, 61.

[MSN, 2007] MSN (2007). MSN.com website. http://www.msn.com/.

[Napster, 2007] Napster (2007). Napster. http://www.napster.com/.

[OceanStore, 2007] OceanStore (2007). The OceanStore Project website. http://oceanstore.cs.berkeley.edu/.

[O'Hara et al., 2004] O'Hara, K., Alani, H., Kalfoglou, Y., and Shadbolt, N. (2004). Trust strategies for the semantic web. In *3rd International Semantic Web Conference (ISWC2004)*, volume 127, Hiroshima, Japan.

[QTrax, 2006] QTrax (2006). Emi music becomes the first major music company to make its catalog available to qtrax: the world's first ad-supported, legitimate p2p service. http://www.viralg.com.

[Ratnasamy et al., 2001] Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S. (2001). A scalable content addressable network. In *ACM SIGCOMM 2001*, pages 161–172, San Diego, CA, USA.

[Seti@Home, 2007] Seti@Home (2007). SETI@home website. http://setiathome.ssl.berkeley.edu/.

[Skype, 2007] Skype (2007). Skype. http://www.skype.com/.

[Song et al., 2005] Song, S., Hwang, K., Zhou, R., and Kwok, Y. K. (2005). Trusted P2P transactions with fuzzy reputation aggregation. *Internet Computing, IEEE*, 9(6):24–34.

[Srivatsa et al., 2005] Srivatsa, M., Xiong, L., and Liu, L. (2005). Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 422–431, New York, NY, USA. ACM Press.

[Stoica et al., 2003] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, F. M., Dabek, F., and Balakrishnan, H. (2003). Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking (TON)*, 11(1):17–32.

[Theotokis and Spinellis, 2004] Theotokis, S. A. and Spinellis, D. (2004). A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4):335–371.

[Tran et al., 2005] Tran, H., Hitchens, M., Varadharajan, V., and Watters, P. (2005). A trust based access control framework for P2P file-sharing systems.

In *HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 9*, Washington, DC, USA. IEEE Computer Society.

[Tsoumakos and Roussopoulos, 2003] Tsoumakos, D. and Roussopoulos, N. (2003). A comparison of peer-to-peer search methods. In *6th International Workshop on the Web and Databases (WebDB 2003)*, San Diego, CA, USA.

[Veríssimo and Rodrigues, 2001] Veríssimo, P. and Rodrigues, L. (2001). *Distributed Systems for System Architects*. Springer, Boston, USA, 1 edition.

[Walsh and Sirer, 2005] Walsh, K. and Sirer, E. G. (2005). Fighting peer-to-peer spam and decoys with object reputation. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 138–143, New York, NY, USA. ACM Press.

[Walsh and Sirer, 2006] Walsh, K. and Sirer, E. G. (2006). Experience with an object reputation system for peer-to-peer filesharing. In *Proceedings of Symposium on Networked System Design and Implementation (NSDI)*, San Jose, CA, USA.

[Yu et al., 2004] Yu, B., Singh, M. P., and Sycara, K. (2004). Developing trust in large-scale peer-to-peer systems. In *1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Zaragoza, Spain.

[Zhao et al., 2004] Zhao, B. Y., Huang, L., Stribling, J., Rhea, S. C., Joseph, A. D., and Kubiatowicz, J. D. (2004). Tapestry: a resilient global-scale overlay for service deployment. *IEEE Journal on Selected Areas in Communications*, 22(1):41–53.