UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
CIÊNCIAS EXATAS E TECNOLÓGICAS,
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
APLICADA - PIPCA
NÍVEL MESTRADO

DANIEL DALALANA BERTOGLIO

Um Sistema de Reputação para Ambientes Peer-to-Peer aplicado em Redes Locais

SÃO LEOPOLDO
2011

DANIEL DALALANA BERTOGLIO

Um Sistema de Reputação para Ambientes Peer-to-Peer aplicado em Redes Locais

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

Orientador: Prof. Dr. Rafael Bohrer Ávila

SÃO LEOPOLDO
2011

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

xxx Bertoglio, Daniel Dalalana

Um Sistema de Reputação para Ambientes Peer-to-Peer aplicado em Redes Locais / Daniel Dalalana Bertoglio. — 2011.

82 f.: il. ; 30cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos - UNISINOS, Ciências Exatas e Tecnológicas, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada - PIPCA, 2011.

"Orientador: Prof. Dr. Rafael Bohrer Ávila."

1. Peer-to-Peer. 2. Sistemas Distribuídos. 3. Sistemas de Reputação. 4. Reputação. 5. Confiança. I. Título.

CDU xxx

Dedico este trabalho
aos meus pais Milton e Carmen e
ao meu irmão Fábio.
Sempre comigo.

AGRADECIMENTOS

Agradeço inicialmente à minha família, que representa as pessoas que para sempre me apoiarão não importa qual decisão eu tomar. Meu pai Milton, por me ensinar a ver o mundo do jeito bom como ele próprio vê, acreditando na verdade e em uma vida melhor. Minha mãe Carmen, pela maneira alegre, extrovertida e simpática de ser, coisas que com certeza eu aprendi com ela. Meu irmão Fábio, que eu carrego 24 horas por dia no meu coração, pela persistência, força e talento. Três pessoas ímpares em minha vida.

Aos meus amigos e familiares que tanto confiam no meu potencial e apostam em mim sem receio nenhum.

Ao professor Dr. Rafael Bohrer Ávila, por ter desempenhado o papel de orientador como poucos, estando sempre presente para o esclarecimento de eventuais dúvidas e problemas, resolvidos considerando a grande facilidade com que repassa seu conhecimento. Dedico este trabalho a ele também, por acreditar em mim e também por ser uma pessoa exemplar, seja como professor ou como amigo.

"Se eu vi mais longe,
foi por estar de pé sobre ombros de gigantes."
Isaac Newton

RESUMO

Atualmente, a atribuição de novas tecnologias aliadas à computação distribuída tem crescido exponencialmente devido a necessidade cada vez mais imposta de otimização dos recursos computacionais. As redes P2P (Peer-to-Peer) são um exemplo de arquitetura de sistemas distribuídos onde os nós atuam tanto na função de clientes como na de servidores. A relação desses nós participantes apresenta questões relacionadas à confiança devido as comunicações que esses realizam, e uma vez que o comportamento dos nós é dinâmico, indicam-se problemas voltados a segurança. Dessa forma, um sistema de reputação e confiança é um meio de se controlar o funcionamento da rede e demais problemas através de métodos de recompensa e penalização. Analisando o contexto de aplicação, nota-se que utilizar um mecanismo de penalização para limitar largura de banda, como acontece tradicionalmente em sistemas P2P, não é uma alternativa viável para ambientes LAN, devido justamente aos recursos que esse tipo de rede detém como característica própria. Dentre diversas outras possibilidades, a limitação de acesso às informações sobre os arquivos compartilhados evidencia adequadamente uma maneira concisa e aplicável para tratar a penalização em redes locais. Este trabalho tem como objetivo então o desenvolvimento de um sistema de reputação para redes locais, denominado TrustLP2P. O TrustLP2P propõe um modelo que visa adequar os aspectos do ambiente de rede local ao mesmo tempo em que atribui conceitos baseados em outros sistemas de reputação existentes, fazendo uso de valores de reputação e confiança para que os nós possam classificar uns aos outros. Aliado a isso, mensurando através dos valores de reputação os participantes, o sistema também tem por finalidade recompensar ou penalizar os nós de acordo com seu comportamento. Um exemplo de aplicação para o TrustLP2P é o LP2P (*Local Peer-to-Peer*), uma plataforma de comunicação para ambientes distribuídos voltada para o compartilhamento de arquivos, desenvolvida pelo projeto da linha de pesquisa de Redes de Computadores e Sistemas Distribuídos do PIPCA, Unisinos. Assim, o sistema proposto proporciona ao LP2P um controle adequado sobre as ações dos nós, através de aspectos consistentes que são descritos no seu modelo.

Palavras-chave: Peer-to-Peer, Sistemas Distribuídos, Sistemas de Reputação, Reputação, Confiança.

TITLE: "A Reputation System for Peer-to-Peer Environments applied in Local Networks"

ABSTRACT

Currently, assign new technologies designed to distributed computing has grown exponentially due to the need for optimization of computing resources. The P2P (Peer-to-Peer) is an example of architecture of distributed systems where the nodes work both in the role of client as the server. The relationship of these nodes presents issues relating to the trust due the communications between them, and since the behavior of nodes is dynamic, security problems are indicated. Thus, a reputation and trust system is one way to control the operation of the network and other problems through methods of reward and punishment. Analyzing the application context, it is notable that use a penalty mechanism to limit bandwidth, as traditionally happens in P2P systems, it is not a viable alternative for LAN environments, due to the resources that this type of network has its own characteristic. Among several other possibilities, limiting access to information about the shared files shows a concise and applicable manner to establish the penalty in local networks. This study aims to develop a reputation system for local networks, called TrustLP2P. The TrustLP2P proposes a model that adapts aspects of the local network environment at the sime time that related concepts based on other existing reputation systems, making use of reputation and trust values so that the nodes can classify each other. Allied to this, measuring the values of reputation through the participants, the system also aims to reward or penalize the nodes according to their behavior. An application for TrustLP2P is LP2P (*Local Peer-to-Peer*), a communication platform for distributed environments focused on the sharing of files, developed by the research project of Computer Networks and Distributed Systems of PIPCA, Unisinos. Thus, the proposed system provides for LP2P an adequate control over the actions of the nodes, through consistent aspects that are described in their model.

Keywords: Peer-to-Peer, Distributed Systems, Reputation Systems, Reputation, Trust.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O uso de novas tecnologias aliadas à computação distribuída está com notória evidência devido a necessidade cada vez mais imposta de otimização dos recursos computacionais. Um sistema distribuído pode ser definido como um conjunto de computadores independentes cujo usuário final visualiza como um sistema único e consistente (TANENBAUM; STEEN, 2001).

As redes P2P (Par-a-Par) são um exemplo de arquitetura de sistemas distribuídos, e dispõem de uma característica principal onde os nós dessa rede são identificados como serventes, ou seja, atuam tanto na função de clientes como na de servidores. Esse tipo de arquitetura tem por finalidade o compartilhamento de arquivos, informações e recursos, sem a necessidade de um servidor central para executar essa distribuição (COULOURIS et al., 2005).

O aproveitamento adequado das redes P2P permite o desenvolvimento de aplicações específicas para essa arquitetura, explorando suas características e naturalmente, suas definições de uso e objetivo. As aplicações precursoras dos sistemas par-a-par se tornaram mundialmente conhecidas pelo compartilhamento de arquivos de músicas, que até hoje corresponde a boa parte da atividade na Internet. Em uma mesma atuação, sistemas como Gnutella (ADAR; HUBERMAN, 2000), BitTorrent (COHEN, 2003) e Kazaa (LEIBOWITZ et al., 2003) são exemplos referenciados para ambientes P2P.

Assim como em grande parte da área da computação, a segurança é um aspecto crucial a ser tratado dentro das redes par-a-par. Como são muitas informações que trafegam entre os nós devido ao compartilhamento de dados, o campo da segurança da informação atua juntamente as demais questões de segurança centrais.

Por sua vez, a segurança da informação é uma área computacional que vêm adquirindo novas formas de tratamento com o passar dos anos. A necessidade de redes e computadores seguros existe há muitas décadas, e há de se manter a responsabilidade de permitir um controle completo sobre os dados e informações relevantes que ficam armazenados em equipamentos diversos (FIGG; ZHOU, 2007).

Com um ambiente descentralizado, fica evidente uma série de problemas relacionados às redes P2P. Existem diversos tipos de ataques específicos para esse tipo de arquitetura, como por exemplo, o ataque *Sybil*, que consiste na falsificação de múltiplas identidades, e também o ataque *Man-in-the-Middle*, onde um nó malicioso entra em meio a comunicação de outros dois nós e passa a responder determinadas mensagens como um desses (ZHU et al., 2006).

Com a notável percepção de que os ataques principalmente a redes P2P, que visam compartilhamento de conteúdo, trazem diversos problemas ao funcionamento dos sistemas devido a sua efetividade, é notoriamente importante tratar esse tipo de problema de forma a garantir um ambiente adequado para determinado sistema P2P (MARTI; GARCIA-MOLINA, 2006). Dentre os vários mecanismos de segurança existentes, um sistema de reputação e confiança é um meio de se controlar o funcionamento da rede e demais problemas através de métodos de recompensa e penalização. Através da opinião dos participantes da rede, se formam cadeias que resultam em um valor de reputação, o qual mensura muitas métricas visando confirmar a qualidade tanto do funcionamento como do serviço geral de um modelo P2P.

Fazer uso de um sistema P2P em uma rede local apresenta vantagens significativas, analisando o contexto onde o mesmo é aplicado. Em primeira instância, é importante analisar a associação da arquitetura P2P com o conceito de rede local. A aplicação de uma plataforma distribuída dentro de uma LAN determina benefícios evidentes uma vez que a latência é baixa e as taxas de transmissão são elevadas, situação que claramente viabiliza um cenário como o de compartilhamento de arquivos.

Analisando os diversos sistemas de reputação já existentes, logo nota-se que estes são aplicados a redes Gnutella, em sua grande maioria. Isso enfatiza a escassez de propostas em torno de ambientes diversificados, como por exemplo, uma rede local. Esse fato deve-se principalmente a intenção pela qual os modelos são desenvolvidos, pois as principais diferenças se dão mais pelos objetivos e métricas do que pela aplicação desses modelos em si. Alterar o ambiente de utilização de um sistema de reputação é uma alternativa evidente, porém, como citado, de pouca realização. Dessa forma, a avaliação dos cuidados referentes aos ataques a sistemas P2P e também da falta incisiva de trabalhos relacionados a modelos de reputação P2P em redes locais, no seu devido contexto, fornece consistência e subsídio à pesquisa e projeto de um sistema de reputação aplicado especificamente para o ambiente de uma rede local, nomeado TrustLP2P.

Desenvolver o TrustLP2P ou, em geral, qualquer outro sistema de reputação para uma rede local, cujo ambiente é controlado, tem a pertinência evidenciada tanto por problemas relacionados a segurança como por problemas relacionados ao desempenho. No aspecto segurança, um exemplo seria o compartilhamento de arquivos de atualização de um sistema operacional, onde um nó poderia então fornecer um desses arquivos maliciosamente ou até mesmo um arquivo corrompido. Assim, é importante que esse nó seja penalizado em virtude dessa possível ação. Já levando em conta o critério desempenho, pode-se analisar que um nó malicioso que faz uso incessante da rede sem contribuir com a mesma e ainda esporadicamente efetua ataques, gera um tráfego desnecessário para o sistema que pode ser controlado através do uso da reputação.

O TrustLP2P apresenta características específicas para tratar o comportamento dos nós em um sistema P2P. A existência dos valores de reputação e de confiança fornecem uma idéia mais adequada para as relações entre esses nós quando da execução das operações do protocolo. Há também aspectos definidos que se baseiam em sistemas de reputação conceituados, de forma a conceder mais credibilidade para a proposta.

As questões de recompensa e penalização tornaram-se, além da principal contribuição da pesquisa, o objetivo central do modelo de reputação tendo em vista o controle que isso exerce sobre o comportamento dos nós. Tradicionalmente, sabe-se que os sistemas P2P penalizam nós maliciosos ou nós que não compartilham limitando a largura de banda disponível para esses, o que não faz sentido para o TrustLP2P, considerando o ambiente de uma rede local. Tornou-se mais interessante limitar o conteúdo, ou seja, limitar que um determinado nó irá receber as informações apenas sobre certa quantidade de arquivos de acordo com o seu valor de reputação. Com isso, um exemplo para aplicação do TrustLP2P é o *middleware* LP2P (*Local Peer-to-Peer*), desenvolvido pela linha de pesquisa de Redes de Computadores e Sistemas Distribuídos do Programa de Pós-Graduação em Computação Aplicada da Unisinos, por ser um sistema P2P aplicado a rede local e principalmente por tratar adequadamente esse aspecto da distribuição da informações para os nós participantes da rede.

O projeto LP2P parte do princípio do aproveitamento adequado de um sistema P2P em uma rede local, com o objetivo de utilizar recursos característicos desse ambiente para a execução do protocolo. Ele é disposto de uma estrutura de comunicação própria, com primitivas, operações e repositórios que atuam de forma distribuída na rede. A idéia basilar, como foco do projeto, é de que os nós participantes dessa rede compartilhem seus próprios arquivos com os demais, formando um único local de compartilhamento lógico. Além disso, ao invés de utilizar uma aplicação separada, como alguns sistemas de compartilhamento existentes, o LP2P tem seu acesso juntamente à interface de um sistema operacional, como se fosse uma pasta compartilhada na rede local.

O restante deste trabalho organizado nos seguintes capítulos: O capítulo 2 trata o referencial teórico, apresentando os conceitos básicos, explicações sobre a segurança da informação e seus atributos, e principalmente detalhando os sistemas de reputação com definições gerais, além das subseções tratarem os modelos existentes de reputação. O final do capítulo ainda contém um resumo das principais contribuições teóricas para a proposta. O capítulo 3 descreve o sistema proposto, TrustLP2P, com toda a sua modelagem e estrutura. Já o capítulo 4 relata a parte da validação, com a análise da aplicação do modelo sobre dois cenários, consituída de diversas situações verificadas. Por fim o capítulo 5 detém as considerações finais sobre a pesquisa e o trabalho realizado.

2 REFERENCIAL TEÓRICO

2.1 CONCEITOS BÁSICOS

Este capítulo apresenta os conceitos básicos utilizados no decorrer do trabalho. Como a construção de um sistema de reputação está diretamente relacionada com um aspecto de segurança, e ainda mais tratando-se de uma aplicação em compartilhamento de arquivos, a segurança da informação é o principal tópico a ser abordado. O entendimento dos atributos e demais itens que compõem essa área se faz necessário para propor o modelo e também para a compreensão de algumas definições que aparecem no referencial teórico.

2.1.1 Segurança da Informação

A segurança das informações, atualmente, é uma questão essencial dentro da computação e até mesmo para outras áreas, uma vez que o advento da tecnologia proporciona um grande número de possibilidades de acesso e uso dessas informações. O crescimento exponencial da acessibilidade traz também determinadas análises, como por exemplo, sabe-se que muitas vezes a importância de certos dados é tão notória que deve-se ter um cuidado especial ao lidar com os mesmos.

Sêmola (2003) aborda a segurança das informações como "uma área do conhecimento dedicada à proteção de ativos da informação contra acessos não autorizados, alterações indevidas ou sua indisponibilidade". Além disso, o autor ainda define que todo componente que faz parte de algum processo que executa determinada ação com a informação é considerado um ativo. Assim, os meios de comunicação e armazenamento, e até mesmo a própria informação são valores desse conceito.

Um dos objetivos da área de segurança da informação é minimizar a freqüência e o impacto de incidentes ocorridos. Dessa forma, a segurança da informação trata a proteção dos dados, dos recursos, dos sistemas e dos serviços, visando prevenir contra esses possíveis danos gerados pelos incidentes (CARUSO; STEFFEN, 1999).

Caruso e Steffen (1999) delimitam ainda, que ao lidar com a segurança das informações, é necessário o entendimento de algumas definições gerais, tais como:

- Acesso lógico: refere-se ao acesso do conteúdo computacional;
- Propriedade ou gestão: de quem pertence o ativo e tem o direito sobre ele;
- Custódia: quem é responsável pelo armazenamento e guarda das informações de outras pessoas/empresas;

- Controle de acesso: definições de senha, permissões, privilégios, identificações e chaves de acesso, assim como ferramentas utilizadas para determinado controle;
- Acesso físico: o modo como se faz uso de algum recurso, informação ou processo;
- Plano de contingência: trata-se de um conjunto de instruções voltado para a manutenção do ambiente de ativos da empresa, que idealiza a segurança contra diversos riscos e ameaças as informações;
- Preservação e recuperação de informações: dentro do ambiente de ativos, a preservação está relacionada à importância de não haver riscos para os dados, garantindo sua sobrevivência. A parte de recuperação diz respeito às informações que foram alteradas ou perdidas, possibilitando uma nova utilização das mesmas.

Percebe-se que a segurança da informação combate os riscos em torno dos dados e vulnerabilidades do sistema, que são problemas notáveis que surgiram juntamente com a tecnologia. Isso é visto, por exemplo, através do armazenamento e o fluxo dos dados que passaram a ser realizados computacionalmente e por consequência otimizaram os processos. Quanto da manipulação das informações, sendo essas criadas, utilizadas, armazenadas ou transportadas, existe uma maior chance de que seja sofrido algum dano, seja repassando os dados para um software, autenticando com utilização de senha, armazenando-o em um banco de dados, enviando por correio eletrônico, respectivamente (SÊMOLA, 2003).

Com a análise de ameaças e vulnerabilidades, é possível compreender e assim auxiliar a prática de uma política de segurança da informação. Sêmola (2003) define que ameaças são "agentes ou condições que causam incidentes que comprometem as informações e seus ativos por meio da exploração de vulnerabilidades, provocando perdas de confidencialidade, integridade e disponibilidade". A relação da análise deixa claro que as vulnerabilidades não causam problemas à segurança, elas representam o meio para o causador desses problemas, que são justamente as ameaças. Dias (2000) detalha que as ameaças em um ambiente informatizado são:

- Vazamento de informações: informações desprotegidas ou reveladas a pessoas não-autorizadas. A questão importante envolvida nesse tipo de ameaça é o valor que os dados descobertos têm para a corporação ou para o negócio representado;
- Violação de integridade: comprometimento da consistência dos dados. Podem ter sido mal preservados ou seguros, sofrendo alguma modificação;
- Indisponibilidade de serviços de informática: impedimento de acesso a recursos computacionais, feito por usuários autorizados;

- Acesso e uso não-autorizado: como citado anteriormente, é um dos principais pontos na segurança da informação. Algum recurso computacional é utilizado por alguém não autorizado ou de forma não autorizada.

Considerando que existem outras diversas ameaças, se percebe a relação direta que uma garantia de melhora na segurança das informações tem com a necessidade de conhecer os tipos de ataques que possam vir a ocorrer. Naturalmente, o bom entendimento das análises de incidente realizadas e também das formas de defesa às ameaças contribui para a mesma garantia.

Para se formar uma idéia base que pode auxiliar no processo de segurança, existem atributos que atendem a necessidade da manutenção dos dados: integridade, confidencialidade e disponibilidade.

2.1.1.1 Integridade

A integridade refere-se principalmente a proteção das informações para que as mesmas não sejam modificadas, seja de forma autorizada ou não autorizada. É um dos pontos mais relevantes devido a criação de uma dependência direta em relação aos outros princípios. Dias (2000) explica que "em termos de comunicação de dados, integridade restringe-se à detecção de alterações nos dados transmitidos". Já Caruso e Steffen (1999), ressaltando os cuidados com a quebra de integridade em arquivos de uma empresa, dizem que "é fundamental que as informações tenham um alto grau de segurança quanto às modificações não autorizadas e/ou acidentais; deve haver somente um proprietário para cada conjunto de informações".

Quando existe uma conexão e, por consequência, uma comunicação através de mensagens, a integridade visa garantir que essas mensagens sejam recebidas conforme enviadas, sem duplicação, inserção, modificação, reordenação ou repetição. Da mesma forma, a perda ou destruição das informações também faz parte do serviço de integridade, o que direciona esse serviço de integridade em uma comunicação a tratar tanto à modificação de mensagens quanto à negação de serviço. Na situação adversa a essa, analisando a integridade sem comunicação/conexão, o objetivo da mesma é a proteção apenas contra modificação da mensagem, sem uma análise em um contexto determinado (STALLINGS, 2008).

2.1.1.2 Confidencialidade

Trata objetivamente a questão das permissões, pelas restrições de acesso. Assim, a informação é protegida de acessos não autorizados, limitando esse acesso apenas para quem lhe é destinado, visando garantir a identificação e a autorização dos envolvidos.

Citado anteriormente, o controle de acesso está envolvido nessa parte juntamente com métodos que permitem a criptografia dos dados.

2.1.1.3 Disponibilidade

Consiste na disponibilidade das informações a qualquer momento, quando necessário. É importante ressaltar que essa condição de acessibilidade é voltada para usuários autorizados, já que para eles, o fato de algum sistema não estar disponível pode ser tão ruim quanto a violação ou destruição do mesmo. Tratando-se de segurança de informações, Dias (2000) cita que "a principal preocupação é prevenir que ataques deliberados ou maliciosos evitem ou dificultem o acesso de usuários autorizados a seus sistemas".

2.2 SISTEMAS DE REPUTAÇÃO

Sistemas de reputação em redes P2P visam balancear e melhorar o funcionamento da rede de maneira geral. Isso se dá pelo fato de que os usuários, em um sistema de compartilhamento, não atuam de forma trivial e padrão, o que significa que não se pode partir da premissa que todos os participantes têm boa intenção em suas ações. Com essa conceituação, percebe-se que os sistemas de reputação agem como um mecanismo de incentivo, fornecendo uma determinada recompensa pelo bom comportamento dos participantes, sendo esse verificado pela sua colaboração ou não à rede. Claramente nota-se que os critérios de avaliação de comportamento, assim como a definição de recompensa e demais métricas, são questões adequadas a cada modelo ou esquema de reputação específico (ANDROULAKI et al., 2008).

Chen e Chen (2007) classificam os sistemas de reputação em três categorias: sistemas de administração centralizada, de abordagem global e de mecanismo localizado.

Administração Centralizada: Os sistemas de reputação de administração centralizada são modelos usados no lado comercial, como por exemplo, Amazon[1] e eBay[2]. No eBay, cada reputação de nó é quantificada. Após cada transação, o comprador pode classificar o vendedor e vice-versa, e então alguns nós centrais monitoram o sistema e tomam as gravações de todos os resultados de classificação. Um problema evidente é que esses nós de monitoramento central podem ser alvo fácil a ataques, e por conseqüência, caso esses falhem, todo sistema sofrerá danos.

Abordagem Global: Sistemas de reputação de abordagem global podem ser exemplificados pelo funcionamento básico do EigenTrust, que é um modelo de reputação

[1] http://www.amazon.com
[2] http://www.ebay.com

explicado mais detalhadamente em uma das subseções seguintes. Nele, uma classificação global de reputação é computada e publicada para cada nó na rede. Nota-se que cada nó atribui ao seu nó vizinho um valor de reputação devido à transações realizadas anteriormente. É importante ressaltar que a abordagem global não é muito escalável, já que em redes P2P de larga escala uma simples transação trará uma rodada de iterações em todo o sistema.

Mecanismo Localizado: Os sistemas de reputação de mecanismo localizado descrevem um tipo de mecanismo onde a informação de avaliação sobre interações passadas entre nós é dispersa através da rede. Essas informações de avaliação, na grande maioria das vezes, são fornecidas por um nó servidor logo depois que dois outros nós interagiram. Naturalmente, essas avaliações no qual descrevem o resultado de interações anteriores auxiliam outros solicitantes a classificar os provedores ou os recursos. Assim, os participantes consultam com outros, adquirindo mais informações antes de fazerem as escolhas de recursos, e a partir dessa consulta, constroem sua própria visão sobre os elementos no sistema em geral.

Chen e Chen (2007) também determinam outros dois aspectos que podem ser divididos quanto a sua classificação em sistemas de reputação: os tipos de nós participantes da rede e o objetivo de avaliação do qual é baseado o sistema de reputação. O primeiro aspecto, que trata os tipos dos nós, pode ser dividido em três categorias:

- Nós Generosos (G): são nós que compartilham recursos autênticos e respondem com a verdadeira opinião quando consultados;
- Nós Egoístas (E): são nós que nem compartilham recursos nem fornecem opinião qualquer quando consultados;
- Nós Maliciosos (M): são nós que poluem o sistema com recursos corrompidos ou inválidos. Eles enganam os demais para que esses façam o *download* de maus recursos e denigrem arquivos autênticos. Além disso, esses nós maliciosos podem atuar em conjunto, conspirando para ganhar a confiança dos outros compartilhando alguns recursos autênticos ou fornecendo opiniões certas.

O segundo aspecto, sobre a base do sistema de reputação, Chen e Chen (2007) dividem em sistemas de reputação baseados em recursos, que lidam com a validação dos recursos, e sistemas de reputação baseados em nós, que abordam a confiança dos nós. Com essa diferença, é defendido que os sistemas de reputação baseados em recursos são mais adequados pelas seguintes razões:

- A avaliação de um nó envolve a interação de alguns fatores externos, como o status de conectividade de rede entre dois nós, enquanto a avaliação da qualidade de um recurso só depende das propriedades do recurso em si;
- O comportamento de um nó pode mudar ao longo do tempo, enquanto a qualidade de um determinado recurso não;
- O critério de avaliação por nós apresenta diferentes aspectos em relação à capacidade desse nó. Em contraste, a validade de um recurso não depende da preferência do nó, de modo que podemos esperar que participantes honestos tenham a mesma visão sobre determinado recurso.

Com essas definições é possível perceber a estruturação de um sistema de reputação, através de métricas, conceitos e demais componentes essenciais para seu funcionamento. É possível ressaltar ainda outros três requerimentos básicos voltados para sistemas de reputação em geral: o primeiro é que para o estabelecimento de uma reputação, um nó precisa de uma identidade de longa duração. O segundo requisito define que é necessária a possibilidade de capturar e distribuir as experiências dos parceiros feitas com a entidade. O terceiro define que é necessário convencer os usuários que a reputação que uma entidade tem é confiável e valiosa, para que parceiros futuros tenham atenção para a reputação. Percebe-se que dentre os requerimentos, o terceiro é o mais complicado de se adquirir, já que depende diretamente da confiança do usuário no sistema. Assim, quando os usuários acreditam e confiam no sistema de reputação, logo são motivados a contribuir com os demais participantes (RESNICK et al., 2000).

Além desses, Zhou e Hwang (2007) apresentam outras características importantes para sistemas de reputação:

- Alta precisão: O valor de reputação calculado deve ser mais perto da real confiança o quanto possível;
- Rápida convergência: O valor de reputação calculado deve se adaptar rapidamente a mudanças de comportamento do nó;
- Baixo *overhead*: O sistema deve não requerer muito poder computacional, armazenamento ou infra-estrutura;
- Adaptação dinâmica aos participantes: Participantes entram e saem do sistema o tempo todo. O sistema deve lidar com essa dinâmica ao invés de confiar em nós pré-determinados;
- Robusto para participantes maliciosos: O sistema deve ser robusto para nós maliciosos independentes e grupos maliciosos de nós usando uma variedade de ataques diferentes;

- Escalabilidade: O sistema deve lidar com sistemas largos o mais eficiente possível.

Essas diversas especificações e características visam constituir um completo sistema de reputação, mesmo que seja notoriamente complicado adequar um sistema a essa abordagem. Xiong e Liu (2004) listam alguns problemas comuns ocorridos na maioria dos sistemas atuais de reputação:

- Não ter a capacidade de diferenciar *feedbacks* desonestos de *feedbacks* honestos. Isso torna o sistema de reputação vulnerável a manipulações de nós maliciosos que forneçam *feedback* desonesto;
- Não oferecer suporte a diferentes contextos para a avaliação da confiabilidade de seus nós. Por exemplo, um nó pode conseguir uma boa reputação por ser honesto em várias pequenas transações e, em seguida, tentar se beneficiar de forma enganosa em grandes transações.
- Não oferecer incentivos para um nó avaliar outros e assim padecer de informações insuficientes;
- Não conseguir lidar com a personalidade dinâmica estratégica dos nós. Por exemplo, os nós maliciosos podem construir uma reputação e, em seguida, começar a prejudicar a rede.

Assim, detalhando a conceituação base sobre a estrutura que compõe os sistemas de reputação, é possível perceber os aspectos os quais diferenciam os modelos de reputação existentes, e por conseqüência efetuar devidas comparações utilizando os mesmos. As subseções a seguir apresentam sistemas de reputação propostos, com sucintas explicações sobre o funcionamento, objetivos, métricas e definições de cada respectivo.

2.2.1 Sistema de Gupta, Judge e Ammar

O sistema de reputação proposto por Gupta et al. (2003) apresenta especificamente um esquema de reputação para redes não estruturadas e descentralizadas. De maneira geral, o objetivo principal desse esquema é fornecer aos participantes um incentivo para participar ativamente no sistema, foco essencial delimitado em mecanismos de reputação. Além disso, há também o objetivo específico de fazer com que os *free-riders*[3] não consigam manter seu comportamento baseado no bom funcionamento do sistema (HAM; AGHA, 2005).

[3]Free-Riders são nós dentro de um sistema de compartilhamento P2P que não colaboram com o fornecimento de arquivos, apenas fazem uso dos recursos do sistema.

A composição para que o esquema de reputação seja distribuído envolve determinados aspectos. Contudo, esse sistema de reputação discutido não fornece a possibilidade de que a solução seja de uma maneira completamente distribuída, pois envolve um Agente de Computação de Reputação (RCA) dentro do sistema (CIGNO et al., 2009).

O sistema propõe basicamente dois mecanismos alternados de computação, a Computação de Reputação Débito-Crédito (DCRC) e a Computação de Reputação de apenas Crédito (CORC). Esses mecanismos são voltados para o esquema de reputação que mapeia cada atividade do nó na rede P2P, relacionando diretamente a um valor de reputação atualizado de forma dinâmica. Para o valor a ser especificado na reputação, pode-se dividir em duas partes a composição do mesmo, conforme visto na Figura 2.1. A primeira representa o comportamento do participante, que consiste na contribuição da procura de conteúdo, que descreve a boa disposição do nó de repassar e processar solicitações, e a contribuição de *download* de conteúdo, na qual descreve a boa disposição do nó para servir e fornecer dados. A segunda parte trata das capacidades, consistindo nas definições de poder de processamento, largura de banda, capacidade de armazenamento e a memória do nó em questão (GUPTA et al., 2003).

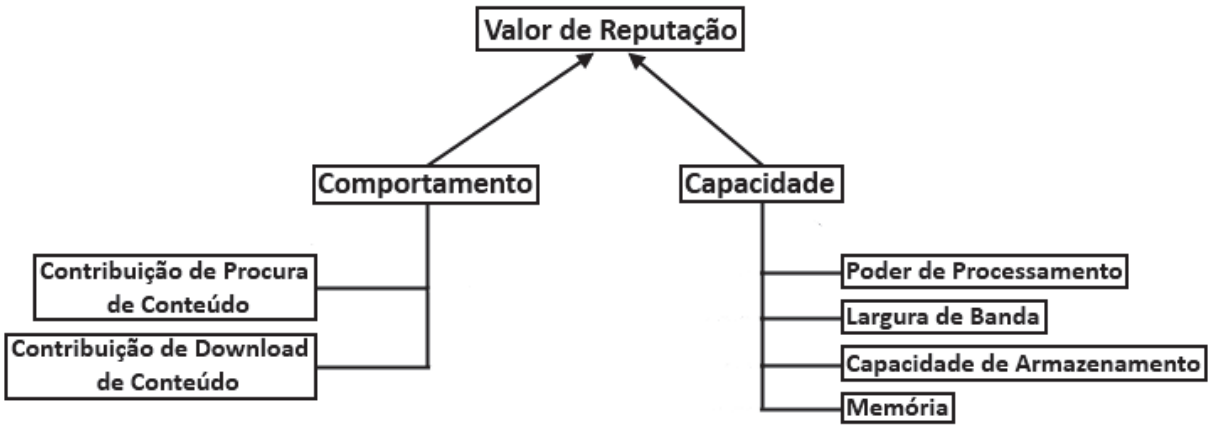

Figura 2.1: Composição do valor de reputação. Adaptada de (GUPTA et al., 2003)

2.2.1.1 Computação de Reputação Débito-Crédito (DCRC)

De acordo com Gupta et al. (2003), o mecanismo DCRC tem o intuito básico de creditar valores de reputação aos participantes por realizarem *upload* e debitar valores pela execução de *downloads*. Seu funcionamento baseia-se em três parâmetros:

1. Parâmetro f: representa o tamanho do arquivo, é usado para determinar quantos MB devem ser transferidos para efetuar a reputação;

2. Parâmetro b: representa a largura de banda, é usado para classificar os nós de acordo com sua largura de banda;

3. Parâmetro t: representa o fator tempo, é usado para determinar quanto tempo um nó tem que compartilhar ou ficar online para começar a ser recompensado.

Baseado nas especificações do mecanismo, percebe-se que os participantes fazem muito mais o processamento de mensagens de consulta-resposta do que servem arquivos para o sistema. A partir da análise e utilizando os parâmetros delimitados anteriormente, o valor total de reputação de um determinado nó é obtido através de quatro definições:

- Crédito Consulta-Resposta (QRC): Define que o mecanismo DCRC usa a média do tamanho das mensagens de consulta-resposta para dar crédito aos nós para estarem online e processarem essas mensagens de consulta-resposta, ou seja, recompensar o participante por esse processamento.
- Crédito Upload (UC): Como cada nó ganha crédito por servir o conteúdo, se define que este nó, com largura de banda bw, servindo um arquivo de tamanho s (MBytes) ganha este crédito calculado por:

$$\frac{s}{f} \times \frac{bw}{b} \tag{2.1}$$

- Débito Download (DD): Da mesma forma que o UC, cada nó que faz *download* de determinado arquivo fica com um débito. Para um *download* de um arquivo de tamanho s (MBytes), o nó i com largura de banda bw, tem seu débito calculado por:

$$\frac{s}{f} \times \frac{bw_i}{b} \tag{2.2}$$

- Crédito Compartilhado (SC): O SC é usado para levar em conta que alguns nós podem compatilhar conteúdos difíceis de encontrar, conteúdos que tem seu *download* feito com menos freqüência. A quantidade de pontos pode ser calculada por:

$$\sum_{j=1}^{n} \frac{s_j}{f} \tag{2.3}$$

  Onde s_j é o tamanho do arquivo j e n é a quantidade de arquivos compartilhados pelos nós.

Assim, o valor total de reputação de um nó k que processa a mensagens de consulta-resposta, facilita b *uploads*, realiza c *downloads* em um d fator tempo é dado por:

$$Reputacao_k = a * QRC + \sum_{l} b_l * UC_l - \sum_{m} c_m * DD_m + d * SC \tag{2.4}$$

Onde UC_l e DD_m são o crédito de *upload* e o débito de *download* de arquivos l e m respectivamente.

2.2.1.2 Computação de Reputação apenas Crédito (CORC)

O mecanismo CORC tem as mesmas atribuições que o DCRC, porém não utiliza os fatores que envolvem *download*, por exemplo, o termo Débito de *Download* (DD). Nesse caso, percebe-se que o valor de reputação calculado somente incrementa, uma vez que, após creditar um nó por realizar um *upload*, não haverá débito para diminuir esses créditos.

Com essa delimitação, determinou-se um problema para que o incentivo da contribuição dos nós continuasse acontecendo. Para isso, o CORC atribui *timestamps* aos valores de reputação, de forma que esse valor de reputação tem um tempo para expirar, caso determinado nó pare de contribuir (GUPTA et al., 2003).

2.2.1.3 Especificações de Segurança

Existem algumas especificações que definem aspectos de segurança do sistema. Esse modelo de reputação determina que os valores de reputação devem ser armazenados localmente nos participantes da rede para que, quando seja necessário recuperar essa informação, o processo seja mais rápido. O uso do agente, o RCA, auxilia contra problemas como a possibilidade dos nós de retornar dados incorretos sobre o seu valor de reputação. Com isso, a atuação desse RCA centralizado é voltada para o cálculo das reputações. Contra a adulteração dos dados, o sistema utiliza criptografia de chave pública.

Assim, quando um determinado participante A deseja ingressar na rede, ele precisa de um par de chaves: a chave pública PK_A e a chave privada SK_A. Da mesma forma, o RCA precisa ter seu par de chaves PK_{RCA}, SK_{RCA}. É possível garantir a não adulteração através da segurança dos valores de reputação pois o nó precisa obter o seu valor de reputação atualizado do RCA através do envio da seguinte mensagem: $\{identidade_{RCA}, timestamp, valor, identidade_{no}\}SK_{RCA}$. Assim ele não pode realizar a alteração do valor, devido ao uso das chaves. Já para garantir a segurança do crédito de consulta-resposta (QRC), determinado nó A envia uma prova de processamento (PP) na forma $\{identidade_{solicitante}, palavras-chave_{consulta}, tamanho_{consulta}, timestamp, identidade_{propria}\}SK_A$ para o RCA e então recebe um novo valor de reputação (GUPTA et al., 2003).

Por sua vez, para garantir a segurança do crédito de *upload* (UC) no CORC, o receptor r do arquivo manda uma mensagem $\{identidade_{receptor}$, $nomedoarquivo$, $tamanhodoarquivo$, $timestamp$, $informacao$ $\}SK_r$ depois do recebimento do arquivo. O remetente s pode verificar a informação usando PK_r e enviar uma mensagem

$\{\{identidade_{receptor}$, $nomedoarquivo$, $tamanhodoarquivo$, $timestamp$, $informacao\}SK_r$, $identidade_{remetente}$, $larguradebanda_{remetente}\}$ para o RCA e obter o seu valor de reputação atualizado.

No esquema de segurança do DCRC, existe o problema que os receptores tem um grande incentivo para evitar o envio da primeira mensagem, já que isso afirma que uma diminuição de sua reputação para a realização do *download* é justificada. Se qualquer remetente ou receptor de transferência não participar do sistema de reputação, não haverá alterações na reputação. Caso os dois participarem, o solicitante do arquivo envia uma mensagem $\{identidade_{receptor}$, $nomedoarquivo$, $tamanhodoarquivo$, $timestamp$, $informacao\}SK_r$ para o nó que fornece o arquivo desejado. Depois dessa transação o compartilhador pode solicitar uma atualização de reputação enviando, da mesma forma citada anteriormente, uma mensagem $\{\{identidade_{receptor}$, $nomedoarquivo$, $tamanhodoarquivo$, $timestamp$, $informacao\}SK_r$, $identidade_{remetente}$, $larguradebanda_{remetente}\}$ para o RCA e então obter o valor de reputação atualizado.

De maneira geral, o sistema de reputação proposto por Gupta et al. (2003) apresenta uma intenção interessante para fornecer tanto uma melhor distribuição no funcionamento do sistema assim como para um melhor balanceamento de carga da rede. Como certa desvantagem nota-se que o agente RCA, por atuar de maneira centralizada, acaba destoando da natureza dos ambientes distribuídos, fazendo com que o sistema não tenha a característica de descentralização de serviços completa.

2.2.2 P2PRep

O esquema de reputação P2PRep é o modelo base de sua extensão XRep, explicado na subseção seguinte. Esse é um sistema desenvolvido para funcionar em redes Gnutella. O objetivo do P2PRep é diminuir o número de *downloads* de arquivos corrompidos, o que é notável pelo seu modo de funcionamento. Uma especificação primordial é o fato de cada nó armazena as suas opiniões sobre os nós com quem interagiu, já que esse armazenamento o diferencia dos demais esquemas de reputação. De qualquer forma, uma vez que a extensão XRep aborda praticamente todas as definições desse modelo, os conceitos a serem explicados são somente de grau básico, ou da diferença da extensão XRep (HOFFMAN et al., 2009).

Cornelli et al. (2002) atribuem que, como identificador do nó, o P2PRep utiliza o hash da chave pública desse nó, e este é chamado de *servent_ id*. De maneira geral, exemplificando, o sistema visa permitir que p, antes de decidir de onde fazer o *download* do recurso, saiba sobre a reputação dos demais participantes através do voto dos seus nós. São apresentadas duas soluções para a tomada de decisão: na primeira, denotada de Pesquisa Básica, os serventes que respondem a pesquisa não fornecem seus *servent_ id*. A

segunda solução, a Pesquisa Avançada, os votantes também declaram seu *servent_ id*, que pode ser levado em conta por p pesando os votos recebidos, ou seja, p pode julgar alguns votantes como elementos de mais confiança do que outros.

É importante perceber que o funcionamento das soluções de pesquisa são descritos através das etapas explicadas pelo protocolo XRep, onde as definições são melhor elaboradas e objetivamente mais claras. De qualquer forma, a Figura 2.2 e a Figura 2.3 abaixo ilustram o funcionamento da Pesquisa Básica e da Pesquisa Avançada de maneira geral (SRIVATSA et al., 2005).

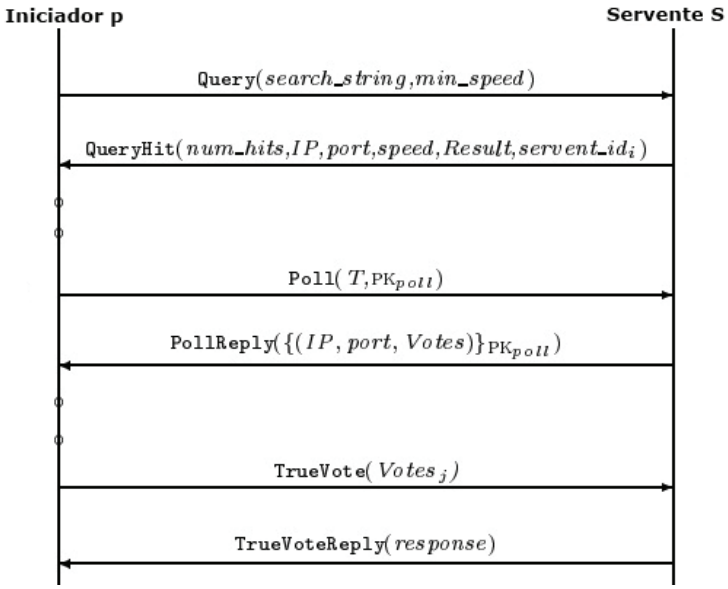

Figura 2.2: Funcionamento da Pesquisa Básica. Adaptada de (CORNELLI et al., 2002)

A Figura 2.2 descreve o funcionamento da Pesquisa Básica, onde o iniciador p envia uma mensagem de consulta (*Query*) com os devidos parâmetros e cada servente que responder a essa consulta envia uma mensagem de êxito (*QueryHit*). Então p seleciona dentre as ofertas um determinado conjunto de serventes, gera um par de chaves e envia uma votação (*Poll*) sobre a reputação desse conjunto para os nós. Esses por sua vez recebem essa lista de serventes contida no conjunto e através da resposta da pesquisa (*PollReply*) informam sua opinião sobre os serventes pelo seus votos. Após isso, p somente se comunica com cada servente para validar os votos que o foram passados. É importante notar que o uso da chave pública nas mensagens de pesquisa serve para garantir a confiabilidade. A Figura 2.3 ilustra a outra solução apresentada.

Nessa Figura 2.3, na solução de Pesquisa Avançada, o funcionamento é bem similar ao da Pesquisa Básica, diferenciado apenas pelo atributo *servent_ id*, que é o identificador do servente. Além disso, na resposta a pesquisa (*PollReply*) é passado o *servent_ id* juntamente com o par de chaves. A mensagem *AreYou* serve para associar o *servent_ id* e o par que foi passado para o voto. Assim, o votante responde com uma mensagem

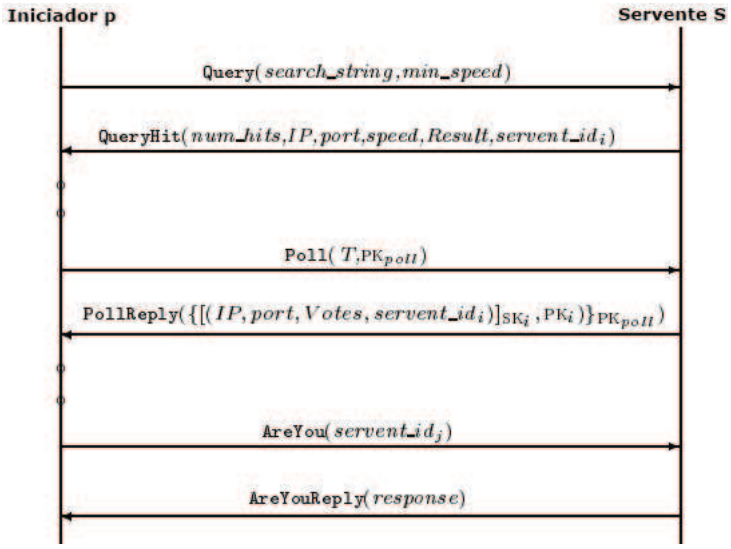

Figura 2.3: Funcionamento da Pesquisa Avançada. Adaptada de (CORNELLI et al., 2002)

AreYouReply confirmando o seu *servent_ id*.

2.2.3 XRep

O XRep é um protocolo estendido do esquema de reputação P2PRep, explicado na subseção anterior. Esse modelo é executado em ambientes Gnutella e fornece facilidades para assinatura, compartilhamento, e combinação de reputações entre serventes e recursos.

O seu funcionamento base no sistema Gnutella especifica que um determinado servente p, procurando por um recurso, envia uma mensagem de consulta (*Query*) por *broadcast* e recebe de volta um conjunto de respostas para *download*. A seleção do recurso do qual será feito o *download* é baseada na qualidade da oferta ou em um critério de preferência baseado nas experiências passadas do solicitante. A qualidade da oferta pode ser, por exemplo, o número de êxitos e a velocidade de conexão declarada por determinado nó. O foco do XRep permite p melhorar a escolha do seu processo de seleção solicitando pela rede a opinião dos participantes (votos) no recursos e suas ofertas (DAMIANI et al., 2002).

De acordo com Damiani et al. (2002), essencialmente, para o tratamento dos recursos e serventes, são assumidas duas definições:

- É requerido que o ID do servente seja um resumo da chave pública, obtido usando uma função de hash e para qual o servente conhece a correspondente chave privada;
- Cada recurso é associado com um identificador obtido através de uma função de hash para o conteúdo do recurso.

A idéia base da abordagem XRep é que cada servente mantém informação de sua

própria experiência nos recursos e outros serventes, e pode compartilhar tal experiência com outros mediante solicitação. Para armazenar essas experiências, se assume que cada nó mantém dois repositórios: o primeiro é o repositório de recursos, uma base contida em uma tabela com os atributos ID do recurso (*resource_id*) e valor (*value*), onde cada *resource_id* que o nó teve experiência é associado a um valor binário descrevendo se o recurso é bom (+) ou ruim (-), expressando a opinião desse nó. O segundo é o repositório de servente, que é uma tabela com atributos (*servent_id*, *num_plus*, *num_minus*), onde é associado a cada nó que o participante interagiu o número de *downloads* com e sem sucesso (DAMIANI et al., 2002).

No repositório de recurso é interpretado um '+' como uma satisfação do nó quanto a um recurso, e '-' como a falta de satisfação. Essa falta de satisfação pode refletir o fato que o recurso não preencheu completamente as expectativas do nó ou também que o mesmo foi corrompido ou malicioso. Como especificação mínima, o valor associado com um recurso deve capturar a confiança do servente na integridade do recurso.

Os valores mantidos nos repositórios são usados para expressar os votos nos recursos e serventes no *framework* do XRep. Os votos, por serem binários, são codificados 1 e 0, onde 1 significa expressa a opinião positiva em um recurso/servente e 0 expressa a negativa. Por exemplo, um servente pode votar 1 apenas nos serventes os quais o atributo *num_minus* é zero ou nos servente os quais o atributo *num_plus* é muito maior que *num_minus*.

Damiani et al. (2002) define que o protocolo XRep é dividido em cinco fases, que são descritas abaixo:

1. Procura de recursos: Um iniciador p envia por *broadcast* para seus participantes da rede uma mensagem de consulta (*Query*) contendo as palavras–chave da sua busca. Quando um servente recebe essa mensagem de consulta para qual ele tem uma palavra coincidente, o mesmo responde com uma mensagem de êxito de consulta (*QueryHit*). A Figura 2.4 ilustra o procedimento de procura de recursos. A composição da mensagem de êxito de consulta inclui o número de arquivos *num_hits* que coincidem com as palavras–chave, um conjunto *ResultSet* contendo os nomes dos arquivos e a informação relacionada, a velocidade em Kbps do participante que responde, assim como o *servent_id* e o par {IP, porta} para ser usado para o *download* dos arquivos.

2. Seleção de recurso e obtenção de votos: Após receber as mensagens de êxito de consulta (*QueryHit*), o iniciador p seleciona, entre as diferentes possibilidades de recursos oferecidos, o recurso r que melhor satisfaz sua solicitação. Tal seleção é realizada de acordo com preferências do usuário e/ou pelo número de ofertas. A solução proposta pelo XRep permite p investigar os nós sobre o *download* e sobre

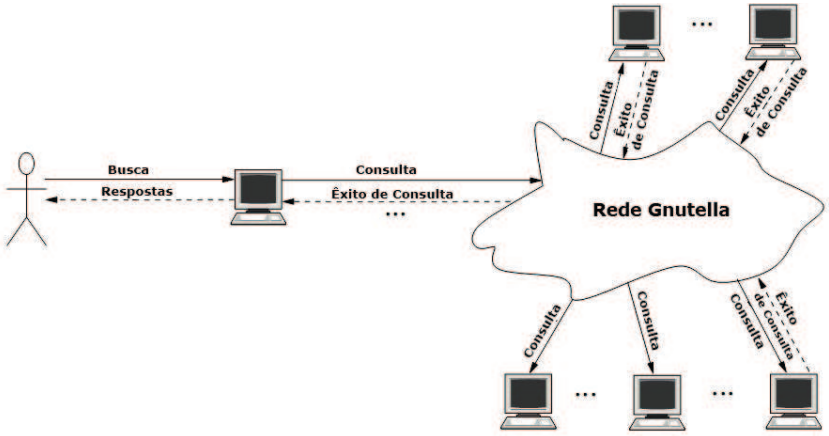

Figura 2.4: Fase 1: Procura de Recursos. Adaptada de (DAMIANI et al., 2002)

a execução. Tal obtenção pode ser feita perguntando aos participantes sua opinião em qualquer recurso ou servente que oferece, conforme pode ser visto na Figura 2.5.

A obtenção dos votos é realizada também via *broadcast* e as mensagens de apuração (*Poll*) são implementadas no topo de mensagens de consulta comum. Para proteger a integridade e confidencialidade dessas respostas *poll*, o solicitante também inclui uma chave pública PK_{poll}, com qual as respostas precisarão ser encriptadas. Tal chave pública pode ser gerada para cada solicitante *poll* ou ser uma chave que p pode usar múltiplas vezes. É importante perceber que o uso de uma encriptação com chave pública não requer uma CA central ou uma alta autoridade, então não é necessário estabelecer ou certificar a correspondência entre um par de chaves e uma identidade. O requerimento é que p seja apenas o único a saber a correspondente chave privada.

Após o recebimento da mensagem *poll*, cada nó verifica seu repositório de experiência e pode responder comunicando seus votos dos recursos assim como dos serventes. Os votos são comunicados de volta como uma mensagem *PollReply* explorando a mensagem de êxito de consulta (*QueryHit*). A mensagem, encriptada com a chave PK_{poll}, inclui os votos do nó, junto com seu IP e porta.

3. Avaliação dos votos: Como resultado da fase anterior, p coleta um conjunto de votos dos recursos e suas ofertas. Baseada sua decisão nos votos recebidos, p precisa confiar na fidelidade desses votos e esse processo de confiabilidade é composto por três passos. No primeiro, p usa a decriptação para detectar votos adulterados e descartá-los. O segundo passo é para reconhecer os votantes falsos ou controlados. No terceiro, p randomicamente seleciona um conjunto V de votantes dentro de cada *cluster*, e diretamente contata cada $v_j \in V$ (usando o endereço IP e porta recebida por ele) com uma mensagem *TrueVote*, solicitando confirmação dos votos que p tem

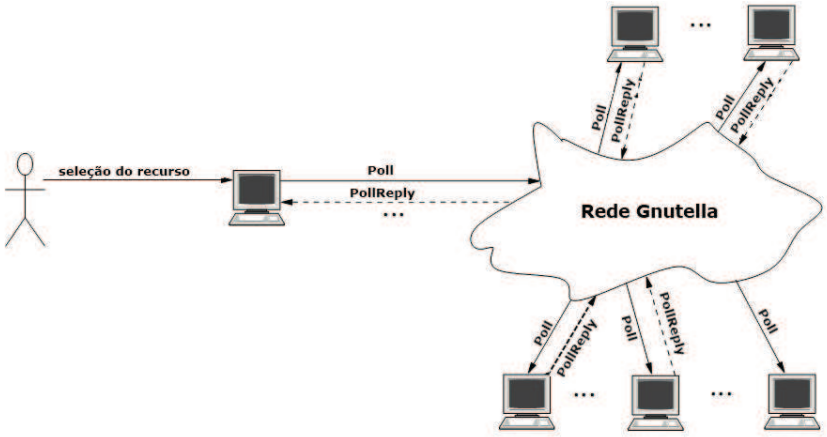

Figura 2.5: Fase 2: Seleção de Recurso e Obtenção de Votos. Adaptada de (DAMIANI et al., 2002)

recebido do mesmo. Cada votante contatado é requerido a enviar uma mensagem *TrustVoteReply* para confirmação desse voto.

Após todo esse processo de verificação, visualizado na Figura 2.6, p pode confiar sua avaliação de reputação de recursos, e então finalmente decidir fazer o *download*. Se p julgar que a evidência da qualidade do recurso não é suficiente, ele pode repetir o processo de voto em outro recurso.

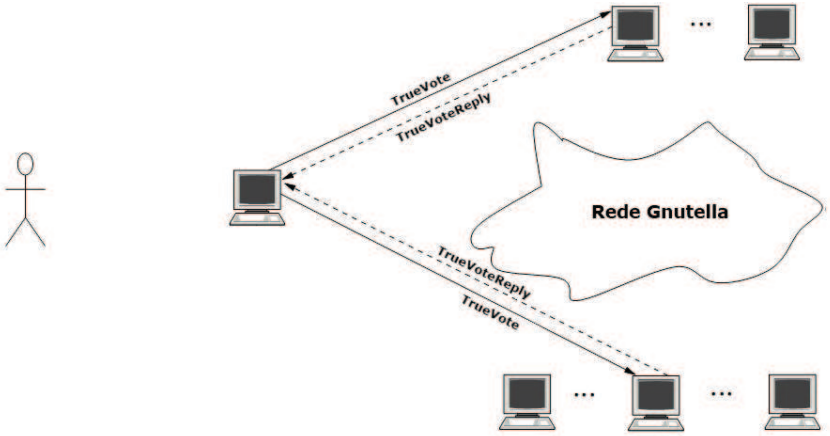

Figura 2.6: Fase 3: Avaliação de Votos. Adaptada de (DAMIANI et al., 2002)

4. Verificação do melhor servente: Uma vez tendo que levar a decisão de fazer *download* de um recurso, p deve selecionar a oferta a qual quer executar esse *download*, conforme mostrado na Figura 2.7.

5. *Download* do recurso: Nessa etapa p decide de qual servente irá fazer o *download* do recurso e então contata o servente escolhido para solicitar o mesmo. Depois do

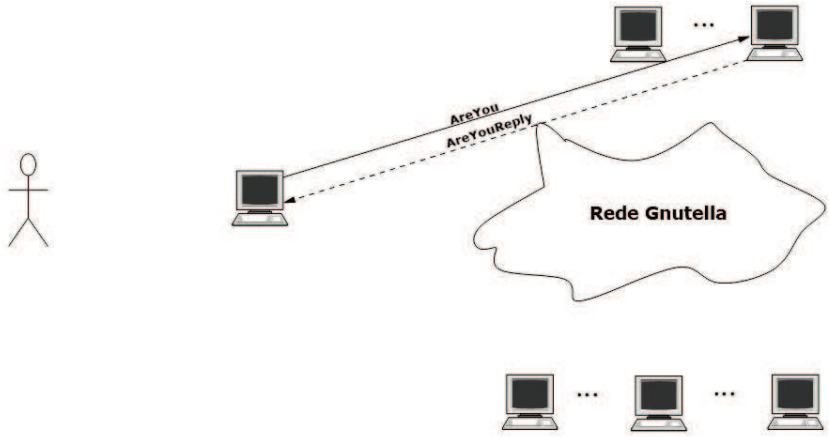

Figura 2.7: Fase 4: Verificação do Melhor Servente. Adaptada de (DAMIANI et al., 2002)

download, p verificará o recurso para garantir a integridade, além de atualizar seus repositórios com a opinião sobre toda transação em si.

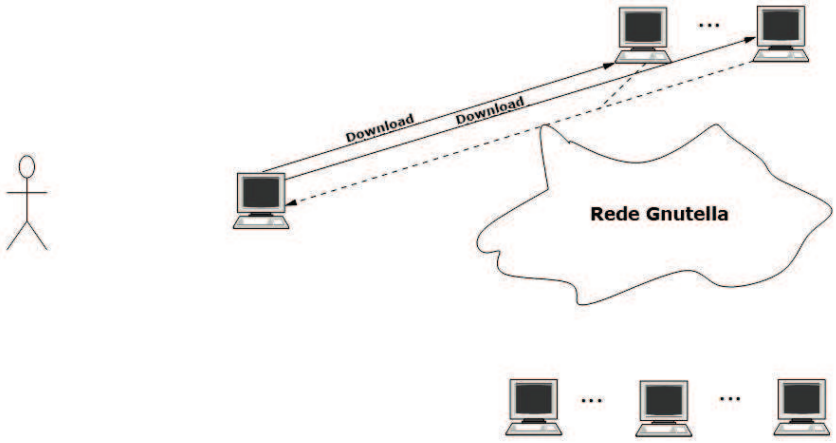

Figura 2.8: Fase 5: Download do Recurso. Adaptada de (DAMIANI et al., 2002)

2.2.4 PeerTrust

O sistema de reputação PeerTrust é um modelo de confiança P2P para quantificar e avaliar a confiabilidade dos nós em comunidades *e-commerce* P2P, implementado em uma rede P-Grid. Nesse sistema, a confiabilidade de um nó é definida por uma avaliação que esse recebe provendo serviço para outros nós no passado, ou seja, do *feedback* recebido através das transações realizadas (PIATEK et al., 2008). Tal reputação reflete o grau de confiança que outros nós na comunidade têm sobre o determinado participante baseado nas experiências passadas. Para essa avaliação, Xiong e Liu (2004) determinam cinco importantes fatores:

1. *Feedback*: O *feedback* que um nó obtém de outros nós, recebido como um valor. Reflete quão bem esse participante tem preenchido sua parte do acordo de serviço;

2. Escopo do *Feedback*: O escopo do *feedback* pode ser representado, por exemplo, pelo número total de transações que um nó teve com outros nós, que se torna um importante fator de escopo para comparar o *feedback* em termos de grau de satisfação entre diferentes participantes;

3. Fator de credibilidade da fonte do *feedback*: no PeerTrust é introduzido a credibilidade do *feedback* como um parâmetro básico para construção de confiança, no qual é igualmente importante como o número de transações e o *feedback*. O *feedback* dos nós com alta credibilidade deve ser mais válido do que os nós que tem baixa credibilidade;

4. Fator de contexto da transação: agregado ao *feedback*, serve para para diferenciar as transações mais importantes das menos importantes;

5. Fator de contexto da comunidade: fator que serve para tratar características relacionadas a comunidade e vulnerabilidades.

Para efetuar o cálculo da reputação de um determinado nó, é necessário calcular uma parte que lida com as interações, onde a credibilidade da reputação e o fator de contexto da transação ponderam as opiniões, e outra que trata o contexto da comunidade, que tem um valor de peso ajustável de acordo com a relevância de cada uma. Assim, para a obtenção do fator de interação, são considerados os principais elementos: a opinião do nó, o número de transações e o fator de credibilidade. Esse último é o único elemento que não é um valor armazenado pelo sistema, o que dificulta o processo de computação (XIONG; LIU, 2004).

Xiong e Liu (2004) consideram duas possibilidades para tratar essa questão: a primeira é a utilização da reputação do nó dando a avaliação, porém nem sempre existe relação entre o comportamento em transações e o comportamento nas avaliações, ou seja, determinado nó pode ser honesto nas transações que faz, contudo dar opiniões desonestas para outros nós. A segunda possibilidade é realizar um cálculo com o objetivo de determinar a semelhança entre o nó avaliado e o nó avaliador, verificando as notas fornecidas a nós que os dois tenham interagido. De qualquer maneira, o cálculo para essa segunda possibilidade é deveras complexo.

A métrica geral de confiança é feita então, dada uma recente janela de tempo, com as seguintes especificações:

- $I(u, v)$ denotando o número total de transações realizadas pelo nó u com o nó v;

- $I(u)$ denotando o número total de transações realizadas pelo nó u com todos outros nós;
- $p(u, i)$ denotando outro nó participante na transação nó u's ith;
- $S(u, i)$ denotando a quantidade normalizada de satisfação que o nó u recebe de $p(u, i)$ na sua ith transação;
- $Cr(v)$ denotando a credibilidade do *feedback* submetido por v;
- $TF(u, i)$ denotando o adaptado fator de contexto de transação para o nó u na $i$nésima transação;
- $CF(u)$ denotando o adaptado fator de contexto de comunidade para o nó u.

A fórmula geral do valor de confiança do nó u denotado por $T(u)$, com α e β sendo fatores normalizados de peso para a avaliação coletiva e o fator de contexto de comunidade, é definida por:

$$T(u) = \alpha * \sum_{i=1}^{I(u)} S(u, i) * Cr(p(u, i)) * TF(u, i) + \beta * CF(u) \tag{2.5}$$

Complementando as definições anteriores ao cálculo da métrica, o cálculo do fator de transação é definido pelo tamanho da transação, enquanto o cálculo do fator de comunidade é definido pela razão entre o número de opiniões emitidas e o número de interações realizadas. Entende-se que o estabelecimento do fator de transação tem o intuito de prevenir que os nós acumulem reputação através de transações pequenas, e então, já com uma boa reputação, atuem de forma maliciosa em uma transação grande (SELCUK et al., 2004).

Uma das considerações sobre segurança é o uso de chave pública no PeerTrust. Cada nodo necessite ter um par de chaves para assinar suas mensagens de *feedback* garantindo a integridade e autenticidade. A outra é para lidar com o problema do traidor, onde o modelo além de calcular em todas as interações, efetua também um cálculo da reputação com as interações realizadas em uma determinada janela de tempo. As simulações resultaram positivamente para essa solução de que, se a diferença entre essas reputações seja menor que o limite de erro, a reputação final é calculada nesse intervalo, o que causou uma diminuição relevante na reputação de um nó que esteja sendo malicioso.

2.2.5 Sistema de Marti e Garcia-Molina

O sistema de reputação proposto por Marti e Garcia-Molina (2004) tem o seu modelo baseado no estudo de dois sistemas de reputação, um sistema no qual os nós compartilham

sua opinião e outro no qual apenas estatísticas locais são utilizadas. O principal objetivo desse modelo de reputação, no qual é implementado sobre o Gnutella, é combater nós de rede maliciosos. Marti e Garcia-Molina (2004) classificam três comportamentos para nós maliciosos, denotados pelas letras N, L e C:

- N: Nós servem arquivos inválidos, porém dão opiniões sinceras;
- L: Nós maliciosos mentem para seu próprio ganho, dando uma má opinião para qualquer outro nó;
- C: Nós maliciosos conspiram. Eles dão boas opiniões para cada outro e más opiniões para nós bem comportados.

A partir da identificação dos nós, as interações entre os mesmos se realizam de maneira criteriosa, uma vez que antes de se iniciar uma determinada transação, cada nó analisa as opiniões sobre aquele com quem irá interagir. As opiniões são armazenadas em cada nó, sendo essas opiniões apenas sobre os nós com quem o mesmo interagiu. Para padronizar as medidas de comparação entre os nós, a métrica utilizada é a taxa de arquivos autênticos que cada nó forneceu sobre o número total de arquivos.

Com esse sistema onde cada nó mantém estatísticas sobre quantos arquivos tem verificado a partir de cada outro participante e quantos desses eram autênticos, a avaliação da reputação é calculada, conforme citado anteriormente, como a fração de arquivos verificados que eram autênticos pelo total de arquivos. Essa avaliação resulta então em um intervalo que varia de 0 até 1, com 0 significando nenhuma verificação de autenticidade passada e 1 significando todas verificações de autenticidade passadas.

A partir dessa padronização da métrica, Marti e Garcia-Molina (2004) definem que a seleção dos nós pode ser feita de duas formas:

- *Select Best*: esse procedimento de seleção define o nó com a maior avaliação como o escolhido. Caso a resposta selecionada seja inválida, o procedimento escolhe o próximo nó melhor avaliado. Existe a possibilidade de sobrecarregar um determinado nó, o que pode ocasionar o fato de que outros nós não terão chance de melhorar sua reputação;
- *Weighted Select Best*: visando a não sobrecarga, conforme citado anteriormente, o procedimento *Weight Select Best* define que cada nó que possui o arquivo desejado possui uma probabilidade de ser escolhido, sendo essa relacionada de maneira proporcional ao seu valor de reputação. Por exemplo, se os nós i e j fornecerem respostas ao nó q, e a reputação de i para q denotada por $R(q, i) = 0.1$ e a reputação

de j para q por $R(q, j) = 0.9$, significa então que j tem nove vezes mais chances de ser escolhido comparado a i.

Existe também a possibilidade de não se utilizar apenas as próprias opiniões, e também analisar as opiniões de outros nós, fazendo isso através do compartilhamento dos valores de reputação. Marti e Garcia-Molina (2004) definem esse tipo de avaliação compartilhada como uma extensão do modelo de reputação local. A extensão delimita que, quando um nó, q, recebeu os resultados da consulta pelo arquivo, esse contata um conjunto de outros nós, Q, para obter as avaliações sobre os que apresentaram resultado. Cada nó votado, ou votante $v \in Q$, responde com sua avaliação (de 0 até 1) para qualquer um que o responde que tem interagido e, portanto, coletou estatísticas. A determinação dos nós que farão parte desse conjunto Q pode ser realizada ou escolhendo entre os vizinhos ou entre aqueles nós que já interagiram.

Foram feitos experimentos descrevendo que a utilização de nós que já interagiram diminui o número de downloads de arquivos que não são autênticos, devido ao fato de que o cálculo, explicado posteriormente, considera a opinião dos nós v que estão fornecendo opiniões sobre o nó r. Uma forma para aumentar o desempenho é a utilização de um cache dos nós conhecidos com maiores reputações.

A fórmula então para o cálculo da avaliação final possui um fator com a nota/opinião local e outro fator que contém a nota/opinião dos outros nós, sendo esses fatores ponderados, ou seja, cada um tem determinado peso para indicar sua relevância. Esse fator que delimita a parte das notas compartilhadas é calculado como uma média ponderada, de forma que a opinião do próprio nó em questão sobre os nós que o estão avaliando serve como peso para essa ponderação. Com isso, a avaliação do nó r, feita pelo nó q, levando em consideração as opiniões dos nós $v \in Q$ é:

$$\rho_r = (1 - w_Q)R(q, r) + w_Q \frac{\sum_{v \in Q} R(q, v)R(v, r)}{\sum_{v \in Q} R(q, v)} \tag{2.6}$$

Para cada nó que responde, denotado por r, o nó solicitante q soma cada votante (v) classificado de r ponderado pela avaliação de q sobre r. Esse resultado é o quorum de avaliação. Se o nó q não tem conhecimento prévio de r, ele usa a avaliação do quorum como a avaliação de r no procedimento de seleção. Se q já tem estatísticas a partir da interação com o nó r, a avaliação para o nó r é a combinação das estatísticas locais e do quorum de avaliação, dada por alguns pesos chamados de pesos do quorum, wQ. Nota-se que quando $wQ = 0$ o sistema de votação trabalha exatamente como o sistema local.

2.2.6 EigenTrust

EigenTrust é um conhecido modelo para gerência de reputação para sistemas de compartilhamento de arquivos, com o objetivo de identificar nós maliciosos através do uso de redes de confiança. Basicamente para alcançar o objetivo, o algoritmo EigenTrust monta uma matriz com as opiniões de todos os nós sobre os outros, e a partir disso calcula a reputação dos nós como o autovetor desta matriz (LEE et al., 2005).

Neste sistema, o armazenamento da reputação local é feito no nó que emitiu a opinião. A reputação global, que é o resultado do cálculo do autovetor, é armazenada em uma DHT, por eficiência de acesso e segurança. Essa reputação global é baseada no histórico de *uploads* de arquivos e é calculada com base nos índices de reputação atribuídos localmente a i por cada nó da rede, pesados de acordo com a própria reputação daqueles nós. Dessa forma, percebe-se que cada nó tem uma visão local de confiança que é descrita pela quantidade de transações satisfatórias e insatisfatórias que foram executadas com os demais nós. A percepção local é denotada pelo valor s_ij, que representa a opinião do nó i em relação a j (KAMVAR et al., 2003).

$$s_{ij} = sat(i,j) - unsat(i,j) \tag{2.7}$$

Os valores de reputação nesse sistema não são contidos em algum intervalo definido, ou seja, cada nó pode utilizar o intervalo que desejar para a atribuição de notas para cada serviço. Contudo, para otimizar a realização de comparações entre as reputações, é feita a normalização desse valor de reputação dividindo-o pela soma de todas as notas dadas pelo nó. Por exemplo, em um nó i, a reputação do nó j é normalizada efetuando a divisão do valor de reputação de j em i pelo somatório de todos os valores de reputação que i armazena. Esta normalização também pode evitar que um nó malicioso atribua um valor consideravelmente baixo ou elevado à um outro nó da rede (KAMVAR et al., 2003).

$$c_{ij} = \frac{max(s_{ij}, 0)}{\sum_j max(sij, 0)} \tag{2.8}$$

Existem algumas desvantagens em normalizar o valor, e uma delas é que os valores são relativos, o que significa que os mesmos não podem ser interpretados de maneira absoluta. Isso acontece porque se no armazenamento em i existem dois nós j e k que possuem o mesmo determinado valor de reputação r, no entendimento de i, j e k são igualmente reputáveis, porém ele não sabe se a reputação dos dois é boa ou má.

O EigenTrust permite que determinado nó consiga, de forma recursiva, contato com todos nós da rede para conseguir a opinião sobre o nó o qual deseja efetuar a comunicação. Esse nó pergunta aos seus nós conhecidos sobre os valores de reputação que esses definiram

ao nó o qual ele deseja se comunicar, e então usa uma média ponderada com as reputações deles para calcular a confiança dele nesse nó a ser comunicado. De forma a expandir a quantidade de opiniões sobre esse nó o qual se quer descobrir a reputação, esse nó solicitante pode pedir a opinião dos nós conhecidos dos seus conhecidos, e assim por diante, onde entra a recursividade. A transitividade é uma questão proposta pelo EigenTrust, uma vez que essa confiança transitiva permite que os valores cij sejam agregados. O nó i para conseguir o valor de confiança de k, deve solicitar todos os seus j nós conhecidos, pesando essas opiniões de acordo com o valor que i tem de j (KAMVAR et al., 2003).

$$t_{ik} = \sum_{j} c_{ij} c_{jk} \tag{2.9}$$

Algumas especificações são notáveis, como o fato de que a reputação de um nó deve ser calculada por mais de um nó da rede, e igualmente deve ser armazenada em outro nó. É possível concluir então que não é permitido que o próprio nó seja responsável por calcular sua reputação. Através de uma DHT são determinados os nós que computam o valor de confiança de um nó específico, e o EigenTrust consegue evitar que um nó saiba a identidade dos nós que ele irá calcular essa confiança. Assim, o combate contra nós maliciosos já fica evidente uma vez que um possível nó malicioso não consegue escolher determinado nó para aumentar a reputação.

Como a atribuição dos valores de reputação aos seus gerenciadores é feita através de rede DHT (por exemplo, CAN ou Chord), a localização de determinadas informações de um nó é calculada por um hash de um identificador que seja único, como o IP e a porta TCP. É importante ressaltar que os gerenciadores de reputação são escolhidos através do uso de funções hash distintas, e isso apresenta a vantagem do anonimato de um nó o qual o valor de reputação é calculado. É possível perceber através dos resultados dos experimentos com o EigenTrust realizados que o sistema apresentou uma melhora substancial mesmo que 70% dos nós fossem maliciosos. Para a análise foram divididos possíveis tipos de ataque e esses categorizados. Os piores resultados foram contra o tipo C, que fornece arquivos ruins x% das vezes, com x = 50%, e o tipo D, que é quando os nós comportam-se como nós bons, mas dão notas boas a nós do tipo B (nós que sempre fornecem arquivos ruins, e dão notas boas a determinado grupo de nós). De qualquer forma, pode-se ressaltar que a proporção de arquivos autênticos por arquivos falsos fornecidos por nós maliciosos foi grande (KAMVAR et al., 2003).

2.2.7 Feldman

O modelo de reputação proposto por Feldman et al. (2004) é desenvolvido essencialmente para ter as seguintes propriedades que caracterizam um grande conjunto de

sistemas P2P:

- Dilema Social: Uma cooperação de todos os nós resulta em uma utilidade total ótima, mas participantes que exploram a cooperação dos outros nós enquanto não cooperam propriamente se beneficiam mais que usuários que cooperam. Isso significa que de maneira geral, a cooperação irá oferecer benefício a todos os participantes da rede, mesmo aos que utilizam os outros que terão benefício maior próprio para determinado nó. Dessa forma, o funcionamento implica que a não cooperação diminui consistentemente o desempenho da rede;

- Transações Assimétricas: Um nó pode querer serviço de outro enquanto não está atualmente habilitado a prover o serviço que esse segundo nó deseja. Esses serviços fornecidos e recebidos por um participante não precisam ser necessariamente para o mesmo nó. Isso significa que existe um tipo de assincronia que o sistema tem que tratar, já que um nó pode receber serviço de um e prover para outro;

- Negação de serviço não-rastreável: Um nó não deve ser habilitado para determinar a identidade de participantes que não querem oferecer o serviço. Não existe a possibilidade de ter conhecimento se um nó tem um arquivo específico ou apenas não quer responder a busca feita pelo nó;

- População dinâmica: Os nós podem mudar seu comportamento e entrar e sair do sistema quando quiserem, de forma contínua.

Assim como os demais sistemas de reputação, o modelo de Feldman propõe o combate a determinados tipos de ataques ou nós maliciosos. No seu modelo, o objetivo desse combate é voltado para os *free-riders*, uma vez que esses podem causar problema a rede devido a carga colocada em um pequeno conjunto de nós. Isso pois os *free-riders*, que não contribuem com o sistema e apenas efetuam *downloads*, se aproveitam desse pequeno conjunto de nós que colabora com a rede, concentrando os *downloads* nos mesmos (FELDMAN et al., 2004).

Além das propriedades citadas anteriormente, o sistema proposto por Feldman apresenta as seguintes características essenciais:

- Discriminação da seleção de servidor: cada nó na rede deve manter um histórico de ações de outros nós para ele ou de nós que receberam algum serviço dele, e então utilizar esta informação para auxiliar a escolha de qual nó utilizar como servidor. Assim, existe uma tendência de que o nó já escolha um servidor confiável, e o mesmo pode também escolher dar chance aos nós que utilizaram seus serviços, como uma forma de retribuição;

- Compartilhamento de histórico: uma vez que cada nó armazena seu histórico, o compartilhamento do mesmo auxilia no processo de reputação dos demais nós que desejam saber informações sobre outros. Isso acontece pois compartilhar as opiniões sobre os nós pode fazer com que um determinado nó consiga a opinião sobre outro que ele nunca teve interação, o que claramente justifica um certo aumento da eficiência do uso desse histórico;

- Reputação baseada no algoritmo de fluxo máximo: age contra a formação de um conluio, ou seja, que um grupo de nós maliciosos seja formado visando fazer mal ao sistema, atribuindo boas notas a esses nós maliciosos. Para isso, o sistema pode fazer uma verificação da reputação do nó que esta fornecendo uma avaliação, e após, a avaliação de quem atribuiu esta avaliação, e assim por diante. O uso do fluxo máximo efetua este cálculo, e esse tem uma complexidade de $O(V)$, onde V é o número de nós da rede. Assim, como o processamento terá uma certa demora, devido a essa complexidade, faz-se uso de um algoritmo aproximado;

- Política adaptativa para estranhos: a política visa combater o *whitewashing*, através do procedimento de fazer com que a reentrada no sistema tenha certo custo. Para isso, os recém-chegados no sistema tem uma taxa de confiança atribuída a eles, obtida pela proporção dos serviços consumidos sobre os fornecidos, ou seja, o número de contribuições e de utilizações desses recém-chegados;

- Histórico de curto prazo: o histórico cria a possibilidade de um nó com bom comportamento se transforme em um traidor e use sua reputação para explorar outros nós. O histórico de curto prazo então combate esse possíveis traidores, detectando rapidamente um nó que passou a se comportar mal, utilizando apenas uma memória de curto prazo.

Além dessas características, Feldman et al. (2004) ainda analisa a dinamicidade de população em redes P2P, assim como o Dilema do Prisioneiro e a função de decisão de reciprocidade. Esta última cabe uma melhor explicação já que aparece mais evidente no sistema proposto. A métrica utilizada nessa reciprocidade, para a escolha sobre cooperar ou não é definida como generosidade normalizada. Essa métrica mede o benefício de uma entidade prover em relação ao benefício dela consumir. Dada uma entidade i, sendo p_i o serviço provido e c_i o serviço consumido, a sua generosidade é definida por:

$$g(i) = \frac{p_i}{c_i} \tag{2.10}$$

Esta medida devido a determinadas políticas (por exemplo, negação de serviços a estranhos) pode fazer com que nós cooperativos neguem serviço um ao outro. Isso pode

ser sanado através da normalização do nível de generosidade (g) do nó provedor do serviço (i) pelo próprio nível (j):

$$g_j(i) = \frac{g(i)}{g(j)} \tag{2.11}$$

2.2.8 RCertP

O RCertP é um protocolo que tem o objetivo de fornecer armazenamento e acesso de forma eficiente de avaliações de reputação. O armazenamento das opiniões é feito numa estrutura denominada RCert, que consiste em dois componentes: o cabeçalho e a estrutura RCertUnit.

O cabeçalho RCert fornece informações sobre o seu proprietário, como a identidade do proprietário e a chave pública do proprietário, assim, essa informação vincula o RCert ao seu proprietário. Além disso, o cabeçalho também inclui informação sobre RCert tal como ID do atual RCert e ID do anterior caso esse certificado não seja o primeiro criado pelo proprietário. Com a informação do ID, proprietário consegue criar um novo RCert mas ainda fornece um ponteiro ao anterior RCert, para que quando um RCert cresça muito, o proprietário pode então criar um novo RCert e fornecer a referência para o velho RCert no cabeçalho. O velho RCert pode ser armazenado localmente no sistema e apenas ser enviado para o serviço solicitante no qual requisita esse determinado RCert. Já cada estrutura RCertUnit contém a assinatura do avaliador abrangendo todo o certificado, de forma que basta conferir a última assinatura para se certificar da validade do certificado todo (OOI et al., 2003). RCertUnit contém as seguintes entradas:

- TimeStamp - emitido pelo titular do direito antes de uma transação ser iniciada. Esse é assinado digitalmente pelo emitente e é usado como prova da transação;
- Rating - esse é o comentário dado por um nó que teve transação com o proprietário. Ele registra a experiência de transação do avaliador com o proprietário;
- RaterID - essa é a identidade do nó que criou essa classificação (RCertUnit);
- Signature - a assinatura é criada pelo avaliador, usando sua chave privada, em todo RCert incluindo o cabeçalho para a integridade do RCert.

De acordo com Liau et al. (2003), o protocolo básico (RCertP) consiste em uma negociação em 6 passos com o nó, conforme visto na Figura 2.9, na seguinte ordem:

- Passo 1 - Solicitação do Serviço: Assume-se que se um nó precisa de certos serviços de outros nós, esse usa o mecanismo de descoberta de recursos para localizar o provedor do serviço;

- Passo 2 - Resposta dos Nós: Todos os nós que tem os recursos necessitados pelo nó solicitante enviam resposta junto com seu Certificado de Reputação (RCert);
- Passo 3 - Reconhecimento: Depois de avaliar todo o RCert, o nó solicitante toma a decisão de qual nó escolher como provedor do serviço e envia um reconhecimento para o provedor. Esse reconhecimento é assinado digitalmente com a chave privada do solicitante e essa deve ser usada como uma prova da transação solicitada;
- Passo 4 - TimeStamp: Após o reconhecimento, há o envio do TimeStamp do provedor para o solicitante. O TimeStamp é assinado pelo provedor e nesse protocolo contém o valor de tempo da máquina provedora. O solicitante verificará então o tempo e assinatura no Timestamp usando a chave pública do provedor. Não é assumido que existe um tempo sincronizado entre solicitante e provedor. Contudo, isso deve ser um meio para o solicitante verificar se o tempo está correto;
- Passo 5 - Transação: A partir desse passo os nós então começam a transação. Completada a transação, o serviço solicitante começa a classificar o serviço provedor. O avaliador (serviço solicitante) atualiza o RCert enviando o mesmo no passo 2 adicionando o Timestamp para o passo 4, seguido pela classificação baseada na experiência da transação. O avaliador também adiciona seu ID e completa a atualização pelo *hashing* do conteúdo do certificado, além de assinar digitalmente o *hash* utilizando sua chave privada;
- Passo 6 - Atualização do RCert: Depois da avaliação do consumidor e demais especificações do passo 5, o novo certificado atualizado é enviado ao provedor para ser apresentado ao próximo serviço solicitante.

Figura 2.9: Os 6 passos do protocolo do RCertP. Adaptada de (LIAU et al., 2003)

2.2.9 RCertPX

O protocolo RCertPX foi proposto visando tratar um problema apresentado no RCertP. Esse problema é que o protocolo permite que o nó avaliado descarte o certificado novo, caso esse possua uma avaliação negativa, já que não existe nenhuma forma

de se determinar se o certificado entregue no início do protocolo é realmente o último emitido. Na extensão do protocolo RCertP, o RCertPX, é introduzido outra estrutura de dados: o Last-TimeStamp (LIAU et al., 2003). O Last-TimeStamp consiste em três elementos:

- TimeStamp emitido pelo serviço provedor;
- Status do TimeStamp (válido/revogado);
- RevokedPeer - identidade da parte autorizada do revogador.

O Last-TimeStamp provê a validação do RCert atual utilizado por um proprietário RCert. A extensão do protocolo envolve os consumidores anteriores do serviço. Esse consumidores agem como Last-TimeStamp holder. Quanto aos passos, na Figura 2.10 nota-se que os passos 1 e 2 são similares aos do RCertP. O passo 3 trata da validade, já que após receber o RCert, o solicitante precisa verificar essa validade do RCert. Isso é feito pela verificação do último RCert Unit no RCert através do contato com o avaliador (OOI et al., 2003).

Figura 2.10: Funcionamento do RCertPX. Adaptada de (LIAU et al., 2003)

No passo 4, que lida com o Last-TimeStamp, caso o avaliador retornar um Last-TimeStamp que não tenha sido revogado, o RCert é validado. Eventualmente onde o último avaliador não está disponível (por exemplo, offline), o solicitante pode tentar contatar os avaliadores precedentes até este um estar disponível. Os passos 5 e 6 são associados aos passos 3 e 4 do RCertP, porém, o TimeStamp usado no RCertPX tem uma informação extra que é o contador de transação. Além disso, ao invés de armazenar

apenas a informação do tempo, o novo protocolo requer ao avaliado incorporar a informação extra no TimeStamp quando enviado para o avaliador. Quando o avaliador recebe o Timestamp ele deve verificar se o tempo e o contador estão corretos no Timestamp. Similar ao RCertP, o Timestamp deve ser assinado digitalmente pelo avaliado. O contador incorporado reflete a última informação sobre a sequência de transação. Por exemplo, se tiveram 20 transações aproximadamente, a informação do contador no TimeStamp deve mostrar 21 como esse valor (LIAU et al., 2003).

É importante destacar que o protocolo RCertP apresenta um modelo um pouco diferente das abordagens dos sistemas de reputação conhecidos, trazendo vantagens e desvantagens. A questão em si é que as opiniões de determinado nó são armazenadas por ele mesmo, o que evidentemente questiona a segurança do protocolo, uma vez que são de responsabilidade do próprio nó as opiniões dadas sobre si. Isso se deve ao fato que esse nó pode adulterar essas opiniões armazenadas, e essa é a grande desvantagem. O fator positivo desse tipo de armazenamento é que a busca por informações é muito mais rápida, já que as opiniões estão de posse do nó avaliado. De qualquer forma, para sanar os problemas acerca dessa questão, o protocolo RCertP utiliza uma infra-estrutura de chave pública, onde cada nó possui um par de chaves e através da chave a opinião fornecida pelo nó é assinada.

2.2.10 LOCKSS

O sistema de preservação digital LOCKSS (*Lots Of Copies Keep Stuff Safe*) é similar a sistemas de armazenamento P2P tais como Free Haven (DINGLEDINE et al., 2001) e Oceanstore (KUBIATOWICZ et al., 2000). Contudo, seus objetivos são mais limitados, descritos através de algumas especificações. A primeira delas é que, em sistemas de armazenamento P2P, os nós cooperam para armazenar dados, com todos nós assumindo juntamente a responsabilidade de armazenar cópias suficientes para prover robustez. No sistema LOCKSS, cada nó é responsável por obter e armazenar sua própria cópia de cada documento a ser preservado. Os nós cooperam apenas para reduzir o custo de preservar sua cópia, detectando e reparando qualquer dano causado pela pouca confiabilidade do hardware (ROSENTHAL et al., 2003).

Outro aspecto no contexto, é que na maioria dos sistemas de armazenamento P2P existe uma dependência de segredos de longo prazo, tanto como base para a identidade entre nós como também de chaves de encriptação para controlar o acesso ao material que eles armazenam. Além disso, como outra especificação pode-se considerar que alguns sistemas de armazenamento P2P lutam pelo anonimato. O DMCA (*Digital Millennium Copyright Act*) exige que os caches LOCKSS tenham permissão do editor para manter seu conteúdo protegido por direitos autorais.

Existem também alguns princípios de design que, a partir de experimentos com o LOCKSS, diferem em quase todos os aspectos do modo convencional. Esses princípios fazem parte de um conjunto de definições abaixo delimitadas (ROSENTHAL et al., 2003):

- Limitar a taxa de operação: nada no sistema deve acontecer mais rápido do que o necessário, o que garante que o sistema degrada o menos possível. Assim nós maliciosos não tem muito tempo para executar suas ações;
- Supor que um adversário é poderoso: não colocar limites arbitrários nas capacidades do adversário;
- Não manter segredos por longo prazo: não assumir que os nós mantenham segredos, como chaves privadas, por mais de alguns dias;
- Independência de identidade do nó: como não há nenhum controle central e não há segredos de longo prazo, existe certa facilidade de falsificar essa identidade, por isso é importante a não-dependência;
- Evitar a reputação de terceiros: confiar na opinião de terceiros quanto ao bom comportamento dos outros nós torna-se um ponto vulnerável devido a possíveis opiniões adulteradas ou falsas, vulnerabilidade esta que é agravada pela falta de identidades estáveis. Basicamente serve para evitar que opiniões corrompidas estraguem o armazenamento;
- Reduzir o acúmulo de crédito: sem identidades fortes não confiáveis, os registros sobre o comportamento em operações passadas não devem ser acumulados por muito tempo;
- Minimizar o número de estados: todas as informações na memória do sistema não são confiáveis devido a longos períodos de tempo, uma vez que essas podem ser perdidas. Com isso, é melhor que o tempo durante o qual o estado tem que ser mantido seja o menor possível;
- Fazer detecção de intrusão inerente: o sistema deve disparar alarmes quando um adversário aplica algum ataque, caso exista um risco significativo de dano irrecuperável. Este alarme deve servir para avisar os administradores responsáveis antes que o ataque seja realmente efetivo.

Com os princípios definidos pode-se analisar como ocorre o processo de votação. Os participantes utilizam o protocolo de sondagens de opinião para obter uma lista de referência que contém uma amostra da população de nós. Então, é escolhido um subconjunto de participantes a partir dessa lista que são convidados a participar da votação. Após,

cada nó deve executar uma função limitada pela memória (*MBF - Memory Bound Function*) como uma forma de mostrar comprometimento com a operação (ROSENTHAL et al., 2003).

A continuação do processo se dá depois da delimitação dos nós que farão parte da votação, com as quatro seguintes ações:

1. Os nós participantes enviam uma mensagem para o chamador;
2. O chamador executa uma MBF;
3. Os nós chamados determinam um subgrupo de nós que irão participar da votação, a partir da sua lista de nós conhecidos;
4. Por último, é enviado ao chamador da votação os resultados dos cálculos dos votos e da prova de esforço de cada nó, calculados individualmente.

A partir do envio do resultado ao chamador, esse apenas confere a prova de esforço e contabiliza o voto. Então, com o fim do processo de votação, o nó que requisitou essa votação atualiza a sua lista de conhecidos, remove todos nós utilizados na votação e adiciona os nós que votaram favoravelmente.

Com o protocolo básico de votação relatado, LOCKSS ainda cita as características deste sistema que são avaliadas de acordo com três aspectos: memória, esforço e autonomia. A memória é um estado persistente, representada por listas de referência, opiniões sobre outros nós e estatísticas sobre a operação local do passado. Os nós usam a memória para entender, do seu ponto de vista próprio, como o sistema evolui no tempo. No entanto, o recurso a grandes quantidades de memória é arriscado, quando o armazenamento próprio de um par não é confiável e também quando as identidades dos parcitipantes podem ser falsificadas (ROSENTHAL et al., 2003).

O esforço é o aspecto que indica a vontade de um nó e sua capacidade de contribuir para o bem-estar do sistema. Os nós podem provar o esforço do outro no processo de uma operação, o que pode limitar a avaliação de transação para nós maliciosos. Alinhado a isso, tem-se ainda o aspecto autonomia. A autonomia é a medida de quão independente cada participante é dentro do sistema. Se a autonomia é baixa, as operações de nós são determinadas pelas informações de outros nós, ou ainda, a partir de um controlador central. Se a autonomia é de alta, as operações dos nós são determinadas pela informação local, ou no caso do LOCKSS, pelas estimativas locais do consenso da população de nós (ROSENTHAL et al., 2003).

A partir das características, Rosenthal et al. (2003) analisam técnicas utilizadas no sistema LOCKSS, identificando em algumas dessas, de que forma a autonomia, esforço e memória atuam nas mesmas.

- Limite de danos: deve ser identificada a taxa máxima de danos que podem ser realizados no sistema, pois isso se torna uma ferramenta poderosa contra possíveis atacantes, uma vez que protege todo o sistema inclusive contra adversários que tem uma notória quantidade de recursos. Isso é realizado pela delimitação de uma janela de tempo onde um ataque possa ocorrer, determinando que cada nó decida o momento de realizar sua reavaliação de arquivos. Essa delimitação implica que um nó malicioso ou um atacante não possa influenciar uma votação, uma vez que esse terá que esperar que o próprio nó realize a votação. Com as taxas de limitação eleva-se a autonomia, ao passo de mais esforço;

- Sinalização custosa: Um nó deve decidir se quer ou não admitir outros colegas em uma operação do sistema: se o nó está chamando uma votação, esse deve decidir quais dos votos que ele recebeu serão usados. Se o nó está convidado a votar em alguém da votação, esse deve decidir se vai votar ou não. A sinalização custosa é utilizada para auxiliar nessas decisões, além de limitar os danos causados durante uma votação. Sinalização custosa exige um esforço grande, mas é independente da memória e da autonomia;

- Reputação: Reputação é uma métrica de confiabilidade ou qualidade de um nó, usada por um par de decidir se e quanto interagir com o outro. Pode ser em primeira pessoa, baseado exclusivamente na experiência do nó local, ou de terceiros, com base em depoimentos de outros pares. Reputação de terceiros é o resultado da colaboração entre nós para manter o banco de dados de reputação, coletivamente recompensar bons comportamentos e punir os maus comportamentos que sejam observados por subgrupos da população de nós. Nos participantes LOCKSS é evitado o compartilhamento de informações de reputação entre os nós, com o intuito de prevenir que um nó malicioso envie opiniões falsas, induzindo determinado nó a confiar em nós maliciosos;

- Expiração de memória: para que nós maliciosos não infiltrem-se na lista de referência de determinado nó, é utilizado um tempo curto de vida para os dados armazenados na memória. Assim, as entradas na lista de referência têm limite de tempo e são limpas após as votações. Da mesma forma, depois que um nó solicita uma pesquisa, ele remove de sua lista de referência os nós que participaram da votação;

- Economia regulada: parte do princípio que é irrealista esperar um sistema completamente autônomo. Um sistema pode atingir um equilíbrio durante a operação normal, mas é raro para as determinadas atividades, em específico no LOCKSS, um ataque baseado em uma vulnerabilidade comum entre uma fração participantes, é tratada apenas no âmbito do sistema. Nesses casos de fraude no mundo real, é

necessária intervenção de um agente externo, como agências de aplicação da lei e tribunais.

Nós do sistema LOCKSS usam alarmes como indicadores de condições excepcionais no sistema. Um nó gera um alarme quando:

1. detecta danos coerentes para o sistema;
2. suspeita de adulteração de sua rede local ou de outros recursos;
3. tem sido incapaz de participar no sistema por um tempo incompatível com falhas de rede.

Operadores humanos respondem aos alarmes ao identificar o problema usando as informações forenses do sistema de coleta e restauram o funcionamento normal. Em sistemas convencionais de detecção de intrusão, os ataques em um sistema só podem ser repelidos por uma cooperação entre o software e o homem responsável pelo mesmo.

2.2.11 Síntese e Principais Contribuições

Após a apresentação de diversos sistemas de reputação existentes, alguns pontos são interessantes de se analisar, tendo em vista a definição do modelo a ser proposto. Como primeira questão a ser observada, percebe-se que esses sistemas de reputação desenvolvidos têm como finalidade, em sua maioria, solucionar os problemas em torno dos nós maliciosos, em seu diversificado poder de ação.

Os métodos de armazenamento de informações em geral dos sistemas de reputação também são mostrados como um foco importante no desenvolvimento dos modelos, uma vez que o funcionamento desses sistemas envolve uma série de dados como valores, opiniões e descrições como um todo. As opções de armazenamento são determinadas pela estruturação das redes, onde alguns dos sistemas utilizam o armazenamento no próprio nó, como o XRep, P2PRep e o modelo de Marti e Garcia-Molina, e outros, em uma estrutura específica de acordo com o ambiente aplicado, como no caso do PeerTrust e até mesmo do sistema proposto por Feldman. O sistema proposto por Gupta, Judge e Ammar armazena as informações no nó avaliado, com a assinatura de um servidor central, enquanto no EigenTrust cada nó armazena suas opiniões mas a reputação global é armazenada de maneira diferente.

A construção das reputações é bem variável, onde alguns sistemas, por exemplo, fazem uso de cálculos com métricas definidas para definição do valor de reputação final. Essas reputações podem ser representadas por símbolos (+ e -), por textos descritivos

(Bom, Médio ou Ruim) ou por valores dispostos em uma faixa com limites (0 a 1). Essa representação não conta como item de comparação entre os sistemas, porém, a utilização de valores é mais usual que as demais.

A identificação dos nós dentro dos modelos não é especificada, mesmo sendo uma definição de suma importância para a conceituação de um sistema de reputação. Apenas o RCertP que faz o uso de chaves públicas para tratar essa questão. Ainda quanto a estrutura desses modelos, considera-se outro fato importante que quatro dos nove sistemas de reputação tem o Gnutella como seu sistema base, enquanto o PeerTrust é voltado para o P-Grid, e os demais não delimitam.

Fora essas principais contribuições citadas, pode-se acrescentar que a percepção dos aspectos relacionados a formulação, funcionamento e estruturação dos sistemas de reputação permite um desenvolvimento bem adequado do modelo a ser proposto. Assim, é possível determinar diversos pontos em comum em todos os sistemas de reputação apresentados, e assim se torna viável efetuar uma comparação para o estabelecimento do modelo a ser proposto. Naturalmente, esses pontos em comum são verificados analisando o ambiente e escopo onde são aplicados, uma vez que os modelos apresentam também suas agregações e definições específicas.

O modelo do TrustLP2P, verificando a síntese das estruturas dos modelos de reputação, tem suas definições baseadas principalmente nos sistemas de Marti e Garcia-Molina e do P2PRep. Essencialmente essas duas propostas apresentam um método mais adequado para o armazenamento, fazendo com que cada nó guarde suas opiniões para gerenciar de maneira mais eficaz, e também estipulam critérios mais interessantes que os demais para conceituar a reputação, utilizando o cálculo da média ponderada, por exemplo.

Efetuando comparações com os modelos P2PRep e Marti e Garcia-Molina, com as devidas proporções, nota-se que o TrustLP2P se diferencia basicamente em seu objetivo principal, que deixa de ser o combate a nós maliciosos e passa a ser recompensar os nós de acordo com seu comportamento na rede. Os critérios para a formulação dos valores de reputação também são adaptados, de forma que as opiniões são armazenadas e formam através da média, um único valor para ser utilizado dentro do modelo.

Ao fim deste capítulo, após a descrição dos sistemas de reputação existentes e da síntese que descreve algumas contribuições desses para o TrustLP2P, são relacionados na tabela 2.1, para efeito de comparação, os pontos que definem o objetivo principal, a forma de armazenamento e os critérios em geral de cada um dos sistemas apresentados.

Com todo o referencial teórico devidamente explicado, assim como a síntese da seção, nota-se que o método de armazenamento, a forma das classificações e os critérios que especificam o valor de reputação são itens que se baseiam em pesquisas consistentes. Isso implica diretamente na formação do sistema de reputação proposto, que além de abordar

Tabela 2.1: Comparação entre os sistemas de reputação

Sistema	**Objetivo Principal**	**Armazenamento**	**Critérios**
TrustLP2P	Penalização e recompensa.	Informações no próprio nó, que tem suas opiniões compartilhadas.	Reputação determinada por cálculos próprios do modelo.
Gupta, Judge e Ammar	Combate a nós maliciosos.	Informações ficam no nó avaliado, e são assinadas por um servidor central.	São creditados pontos por *upload* e debitados pontos por *download*.
P2PRep	Prevenção contra nós maliciosos.	Informações no próprio nó, que compartilha suas opiniões.	Reputação estipulada pela quantidade de opiniões positivas e negativas.
XRep	Prevenção contra arquivos corrompidos.	Armazenamento igual ao do P2PRep, adicionando avaliações sobre arquivos também.	Critérios semelhantes ao do P2PRep, e as avaliações sobre arquivos só determinam se os mesmos são autênticos ou não.
PeerTrust	Combate a nós maliciosos.	Informações armazenadas em uma DHT.	Reputação estipulada pelos critérios: opinião, escopo do nó, credibilidade e contextos de transação e de comunidade.
Marti e Garcia-Molina	Detecção de nós maliciosos.	Cada nó armazena suas próprias opiniões.	Para reputação local é efetuada a divisão de arquivos autênticos pelo total de arquivos fornecidos, para a global é feita a média ponderada da avaliação dos nós pela sua opinião sobre os mesmos.
EigenTrust	Prevenção contra arquivos corrompidos.	Cada nó armazena suas próprias opiniões, porém a reputação global é armazenada em uma DHT.	As reputações são o autovetor de uma matriz que contém as opiniões a_{ij} das opiniões de i sobre j
Feldman	Combate a *free-riders*.	Sem especificação de armazenamento, contudo indica DHT.	Reputação dada pela taxa de serviços prestados e consumidos.
RCertP	Armazenamento eficiente de informações.	Opiniões ficam no nó avaliado, assinadas por cada avaliador.	Sem especificação de critérios.
LOCKSS	Manutenção de informações em longo prazo.	Cada nó mantém uma lista de nós conhecidos.	Ao invés de critérios de reputação especifica uma prova de esforço.

esses itens engloba também aspectos adicionais de recompensa e penalização, objetivando ser o mais completo possível para tratar qualquer eventual problema de comportamento dos usuários.

3 O SISTEMA TRUSTLP2P

O presente capítulo faz a descrição do sistema proposto, tratando em suas subseções todos os processos que envolvem o trabalho, e também as explicações sobre o projeto LP2P. Há também uma conceituação específica sobre os subitens do sistema de reputação TrustLP2P, como definições das métricas tais como a forma de armazenamento, os critérios de reputação e confiança em geral e também o método de identificação dos nós.

3.1 LP2P

A utilização da tecnologia P2P como arquitetura de uma rede de computadores tem grande notoriedade devido as suas características, e vem se tornando cada vez mais popular em ambientes acadêmicos e corporativos. Com um ambiente descentralizado, é possível dedicar melhor uso de processamento por parte de cada nó participante, além de disponibilizar uma maior capacidade de armazenamento.

Com base na proposta do sistema LP2P (Local P2P), percebe-se uma nova atribuição para a tecnologia P2P, dessa vez direcionada a uma rede local. A idéia tem como intuito o melhor aproveitamento dos aspectos que correspondem a este tipo de rede, como a baixa latência de comunicação e taxas elevadas de transmissão (*throughput*), aliados as características da tecnologia. O projeto LP2P é um trabalho do grupo de pesquisa de Redes de Computadores e Sistemas Distribuídos do Programa de Pós-Graduação em Computação Aplicada da Unisinos, e é definido como uma plataforma de comunicação para ambientes distribuídos, que além de descentralizada, é escalável e auto-gerenciável (ROCHA et al., 2010).

Como citado no capítulo introdutório, a idéia base de que os nós participantes do sistema LP2P compartilhem seus próprios arquivos com os demais, formando um único local de compartilhamento lógico, fornece algumas afirmações delimitadas. Por exemplo, a existência dos compartilhamentos lógicos, ou seja, locais que contém de forma lógica todos os arquivos compartilhados pelos participantes da rede, constitui facilidades para todo o processo de comunicação do protocolo. É possível existir também um compartilhamento lógico restrito, que utiliza técnicas de criptografia para tornar esse espaço sigiloso. De qualquer forma, não sendo esse determinado compartilhamento lógico sigiloso, é importante citar que a partir do momento que um arquivo qualquer está sendo compartilhado, esse torna-se disponível na rede para qualquer usuário. Isso logo associa uma necessidade evidente de um mecanismo de controle para analisar, verificar e balancear o sistema, já que a disponibilidade dos arquivos fornece inúmeras possibilidades de comportamento dos nós, bons ou ruins.

Assim como outros sistemas P2P conhecidos, o LP2P tem como finalidade o compartilhamento de arquivos de maneira geral, e por consequência, a estruturação do protocolo passa por uma percepção mais ampla de acordo com o objetivo do sistema. Para a definição do protocolo foram estipulados aspectos como o formato das mensagens a serem utilizadas, o protocolo de transporte para a comunicação, o tamanho dos blocos de dados e também o conjunto inicial de primitivas.

Os grupos de primitivas são dois, as de manipulação de arquivos e as de notificações. As primitivas de manipulação de arquivos compreendem as operações de:

- Listagem de Arquivos (*list*): Operação para a listagem de conteúdo de um compartilhamento, o qual envia para os demais participantes uma requisição desse conteúdo;
- Envio de Listagem (*sendl*): Operação em resposta a solicitação *list*, no qual é uma mensagem que contém toda a lista de recursos locais em um compartilhamento do nó que recebeu a primeira mensagem;
- Cópia de Arquivos (*get*): Operação que solicita o envio de um arquivo ou de uma parte deste para o nó solicitante;
- Envio de Arquivos (*sendf*): Assim como o *sendl*, porém em resposta a solicitação *get*, *sendf* envia o arquivo ou o trecho de arquivo para o nó que fez a requisição.

Já as primitivas de notificação do sistema são apenas duas: a de adição de arquivos, que envia uma notificação para os demais nós da rede quando um participante adiciona novos arquivos no compartilhamento, e o de exclusão de arquivos, quando o participante exclui um arquivo, este envia aos demais um mensagem de exclusão com o intuito de que todos atualizem sua base de dados local.

As definições mais técnicas servem apenas para exemplificar de forma breve o funcionamento geral da plataforma de comunicação LP2P. É importante ressaltar que por ser um ambiente descentralizado e, conforme já comentado, formado por nós que se comportam de forma dinâmica, a necessidade de módulos de segurança e de controle sustenta a idéia geral da pesquisa para o desenvolvimento de um sistema de reputação que atue também no LP2P. Naturalmente, o estabelecimento desse sistema apresentou adptações, observando que assim como os sistemas de reputação existentes não são aplicados em um ambiente local, as suas estruturas e controles também não são aplicadas diretamente.

3.2 OBJETIVOS

O objetivo principal do sistema de reputação TrustLP2P é recompensar os nós participantes de acordo com o seu comportamento, de forma a incentivar esses nós a colaborarem

com o funcionamento do LP2P.

A grande maioria dos sistemas de reputação existentes tem por objetivo principal o combate a nós maliciosos, ou para outros demais tipos ataques a redes P2P. Isso delimita uma linha de análise mais em torno de estatísticas de desempenho de rede, ou também de estratégias específicas de prevenção às ações maliciosas efetuadas no sistema.

Contudo, com o objetivo proposto de um mecanismo de recompensa e penalização, subentende-se que a limitação da vazão de dados ou até mesmo de tamanhos de arquivos não é uma opção adequada para o contexto de um ambiente local. Isso é plenamente justificado pelo fato de que em ambientes LAN a largura de banda, por exemplo, dificilmente será obstáculo para transferência de arquivos, devido a sua alta disponibilidade. A limitação no acesso das informações sobre os arquivos se tornou uma alternativa para forma de penalização e recompensa em um sistema P2P aplicado a rede local. Assim, o controle desse mecanismo avalia o valor de reputação do nó e em função disso há a probabilidade desse nó receber as informações sobre determinada quantidade de arquivos listados no compartilhamento. Dessa forma, a avaliação de uma forma de recompensar não ficou voltada para recursos e capacidades, e sim para uma especificação de processo dentro do sistema P2P aplicado.

A seção seguinte apresenta a descrição do modelo, com suas definições gerais de métricas, conceitos e fórmulas para o tratamento de confiança e reputação dentro do sistema.

3.3 DESCRIÇÃO DO MODELO

Para propor o TrustLP2P, a análise de vários sistemas de reputação existentes se fez necessária, sendo essa baseada em algumas métricas. Além disso, os ambientes onde esses sistemas são aplicados e o objetivo dos mesmos também são fatores de comparação para a proposta.

3.3.1 Elementos Avaliados

A primeira definição a ser feita é sobre qual elemento da rede será aplicado as avaliações de reputação. Como já citado anteriormente, essa reputação pode ser aplicada para avaliar os arquivos, com o intuito de indicar o grau de qualidade desses, e/ou para avaliar os nós da rede, com o objetivo de representar a confiabilidade das opiniões fornecidas por esses nós. Para o sistema TrustLP2P as avaliações de reputação são delimitadas exclusivamente para os nós.

3.3.2 Identificação

A identificação dos nós é a segunda principal definição, uma vez que essa atua em todos os módulos do protocolo LP2P. Para suprir as necessidades de segurança e facilidade de comunicação, até pela estrutura das mensagens, foi estabelecido que a identificação dos nós é representada pelo *fingerprint* da chave pública de cada nó, ou seja, os últimos 32 bits dessa chave.

3.3.3 Forma das classificações

Como o TrustLP2P conta com dois tipos de informações, de reputação e de confiança, é necessário especificar de que forma essas informações devem ser classificadas. Conforme já analisado anteriormente, o valor de reputação é definido na forma de número, em um intervalo de 0 a 1, com 1 representando o a reputação máxima. Já as informações sobre a confiança são classificadas em quatro níveis: Nenhuma, Fraca, Forte e Plena.

3.3.4 Armazenamento

A quarta definição é determinar o modo de armazenamento das avaliações, opiniões e valores de reputação do sistema TrustLP2P. Entre as opções de armazenar essas informações, por exemplo, globalmente, localmente ou em um conjunto de nós, conforme acontece em alguns sistemas de reputação citados no referencial teórico, o armazenamento local tornou-se a melhor alternativa, baseado principalmente no funcionamento do protocolo LP2P. Assim, cada nó armazena em uma base de reputação própria a opinião que forneceu sobre outros nós.

A base de reputação, para esse armazenamento local, detém as seguintes informações:

- *PeerID*: armazena a identificação do nó no sistema;
- *Reputação*: armazena a média da reputação de determinado nó;
- *Confiança*: armazena o nível de confiança de determinado nó.

Para essa estrutura na base, cada participante tem suas informações armazenadas em uma única linha. Isso é ressaltado devido ao fato de que um nó recebe a opinião dos demais nós após a resposta a sua operação *list*, e justamente por essa razão que os campos armazenam a média ponderada desses valores. A cada nova entrada de uma opinião, a média é recalculada utilizando a seguinte fórmula:

$$Rep_{no} = Rep_{no} + \Delta \tag{3.1}$$

$$\Delta = \frac{Rep_{remota} - Rep_{no}}{2} \tag{3.2}$$

Onde Rep_{no} é o valor atual de reputação e o fator Rep_{remota} representa a nova entrada de uma opinião fornecida por outro nó.

3.3.5 Mecanismo de Recompensa

Em todo o funcionamento do protocolo LP2P, a especificação da recompensa, visando o objetivo do TrustLP2P, poderia abranger alguns serviços da plataforma aproveitando as características da mesma. Após analisar alguns critérios, o método de recompensa foi definido na operação que envolve a solicitação de informações sobre algum compartilhamento.

Quando o nó deseja solicitar uma listagem de arquivos, ele envia via *multicast* uma mensagem *list*, visando obter as informações sobre compartilhamentos específicos. É importante observar que esse envio da mensagem *list* só acontece caso exista no mínimo um participante na rede que detenha o compartilhamento que o nó ingressante deseja participar. Em virtude do envio dessa mensagem *list*, a resposta através de uma mensagem *sendl* é enviada pelos nós em uma probabilidade $1/n$, onde n é o número de participantes na rede. Assim, a recompensa de acordo com a reputação é então dada pela mudança dessa probabilidade que passa a ser Rep_{no}/n, com Rep_{no} sendo o fator que delimita que, quanto maior a reputação do nó, maior a probabilidade e vice-versa.

3.3.6 Critério Confiança

O critério confiança, dentro do sistema proposto, tem seu diferencial devido ao não envolvimento direto com as análises que envolvem as operações de listagem e obtenção de arquivo. A diferença entre os valores de reputação e os valores de confiança se dá pela explicação dos conceitos em torno dessas definições. Confiança é uma característica baseada em uma experiência direta a qual um nó teve com outro, indicando como esse nó comparou suas expectativas com as ações que aconteceram realmente. O nível de confiança, portanto, é formado de maneira única, uma vez que isso parte da percepção do próprio nó sobre uma determinada situação. Por sua vez, a reputação é uma medida formada por mais de um indivíduo envolvido. O valor de reputação é formado por opiniões, com cada uma delas expressando a análise de um nó sobre outro, formando um consenso geral dentro de um sistema.

A confiança então tem a finalidade de afirmar que determinado participante é quem afirma ser. Já a reputação trata o comportamento do nó em suas transações, se ele está agindo de forma correta no sistema. Por exemplo, um nó A tem em sua tabela que o

nó B tem um alto valor na confiança e um baixo valor na reputação, o que significa que ele confia que B é realmente B, porém não irá efetuar transações com ele devido a sua reputação.

No TrustLP2P a confiança basicamente reforça a credibilidade da reputação, atuando dessa forma como um mecanismo de prevenção contra ataques de conluio, por exemplo. Considerando que diversos nós se combinam de estipular apenas bons valores de reputação entre eles, de forma a utilizar essa boa reputação para obter privilégios na rede, a confiança vai atuar diretamente na verificação que um nó pode fazer antes de responder ou não a solicitações desses nós que estão atacando em conluio.

Dessa forma percebe-se que confiança, além de fornecer consistência a todo o mecanismo de reputação, visa combater problemas de identidade funcionando quase como um mecanismo de autenticação, porém, não tão preciso quanto um específico desse tipo. Ademais, há de se considerar também que a transitividade da confiança auxilia o processo de fazer com que o nó se afirme dentro do sistema.

3.3.7 Funcionamento e Definições Gerais

O funcionamento geral do TrustLP2P se dá essencialmente por dois processos: a determinação do valor de reputação e a atualização das tabelas de reputação. No primeiro processo, o qual define o valor de reputação, um nó após receber a resposta de sua operação *list* e verificar todos os arquivos contidos nessa resposta, estipula a reputação de um determinado nó de acordo com a proporção entre a quantidade de arquivos o qual ele tem interesse e a quantidade total de arquivos desse outro nó. Essa proporção é dividida em intervalos, delimitada pelos percentuais, que são obtidos através do cálculo:

$$TP = \frac{IP * 100}{F_{Rede}} \tag{3.3}$$

Além deste cálculo, o TrustLP2P detém outras fórmulas essenciais para seu funcionamento, que são descritas abaixo:

1. Índice de Proporção (IP): Determina o índice o qual é utilizado efetivamente no cálculo da Taxa de Proporção (TP).

$$IP = \frac{(Ativ_{Ni} * IC)}{F_{no} * T_{get}} \tag{3.4}$$

2. Atividade dos Nós Interessados ($Ativ_{Ni}$): Representa a quantidade de mensagens *get* que são feitas pelos nós interessados (N_i).

$$Ativ_{Ni} = Num_{gets} * N_i \tag{3.5}$$

3. Índice de Contribuição (IC): Representa o índice de contribuição relacionado a quantidade média de arquivos que o nó em análise compartilha com a quantidade média de arquivos compartilhados pelos demais nós.

$$IC = F_{no} - F \tag{3.6}$$

Além desses três cálculos que estão envolvidos em todas as situações analisadas no capítulo de validação do modelo, há também outras duas fórmulas que completam mais detalhadamente as análises, que são:

1. Quantidade de Informações Recebidas (Qtd_{ir}): Especifica a quantidade de informações que um nó recebe sobre os arquivos que estão sendo compartilhados, considerando as atualizações e exclusões que acontecem durante o tempo.

$$Qtd_{ir} = F + Num_{add} - Num_{del} \tag{3.7}$$

2. Penalização por Excesso de List (Pen_{el}): Para obter um controle maior sobre os nós de baixa reputação, a quantidade de operações *lists* realizadas em determinado tempo é verificada, através da diferença entre o tempo em que o nó em análise executa as operações ($T_{list-no}$) e o tempo médio dessa mesma execução porém considerando a rede toda (T_{list}).

$$Pen_{el} = T_{list-no} - T_{list} \tag{3.8}$$

Após realizados os cálculos necessários e então obtida a taxa de proporção relatando o percentual calculado, é consultada a tabela com as faixas das proporções para que seja devidamente associado ao cálculo da reputação o valor correspondente, penalizando ou recompensando o nó. Os valores de cada faixa foram assim determinados em função da variação dinâmica da reputação quando esses são utilizados. A faixa inicial tem o valor de 0,06 para que a penalização seja mais atuante quando a proporção for em um nível tão baixo. Em virtude da multiplicação, sabe-se que as faixas menores que a intermediária (46% a 55%) tem seus valores menores do que 1 para que a reputação diminua, naturalmente. Esses valores pré-estabelecidos em toda a tabela indicam rápidas alterações, mesmo que a diferença entre esses seja pouco expressiva. A faixa de proporção acima da intermediária apresenta a alteração mais clara dentre todas as demais, já que um nó com o valor de TP em 55% que passa a compartilhar mais arquivos e troca TP para 56%

vai passar a ter sua reputação dobrada. Percebe-se que há duas faixas com valores mais diferenciados e as demais estipuladas de forma sequencial, feito justamente para que o comportamento de um nó seja tratado de forma adequada.

Tabela 3.1: Atribuição das faixas de proporção

Faixas	**Valores**
0% a 10%	0,6
11% a 20%	0,8
21% a 45%	0,9
46% a 55%	1
56% a 70%	2
71% a 90%	2,5
91% a 100%	3

Assim, dada que a reputação inicial dos nós é 0,5, caso a proporção seja de 15%, o valor de reputação atual diminuirá devido a multiplicação pelo valor de proporção respectivo da faixa e passará a ser 0,4. Basicamente o cálculo de reputação é descrito por:

$$Rep_{A:B} = Rep_{A:B} * ValorProporcao \tag{3.9}$$

Onde $Rep_{A:B}$ é o valor de reputação que o nó A atribui para o nó B e $ValorProporcao$ é o valor consultado da tabela disposta acima.

O processo de atualização das tabelas de reputação é outra forma onde o valor de reputação é alterado, uma vez que essa tarefa envolve a operação *list* e as informações trafegam junto com as mensagens dessa operação. Quando um nó recebe as respostas do *list* que contém as informações sobre o compartilhamento e sobre os arquivos e seus respectivos proprietários, cada nó que enviou essa resposta anexa juntamente os valores de reputação que contém em sua tabela local, disseminando assim sua opinião sobre outros nós. O nó que recebe esses dados adiciona então essas opiniões em sua tabela, através do cálculo comentado anteriormente.

Percebe-se claramente que o TrustLP2P lida basicamente com as mensagens de *get* e *list*, contudo há de se considerar algumas conceituações importantes quando se analisa as mensagens do tipo de notificação. Por exemplo, para controlar a questão das atualizações, considerando que todos os nós iniciam com a reputação 0,5, sabe-se que quando esses solicitarem listagem de arquivos através da operação *list* receberão várias das informações possíveis. Assim, mesmo que um nó não contribua com a rede e tenha baixa reputação ele já irá possuir parcialmente as informações dos arquivos compartilhados, justamente em função desses *lists* iniciais que o mesmo faz. Para tratar esse aspecto onde um nó com reputação baixa visualiza praticamente as mesmas informações que um nó de reputação

alta, é determinado que as mensagens de adição (*add*) e de exclusão (*del*) são enviadas através de unicast com o controle onde um nó com baixa reputação não receberá essas mensagens. Dessa forma, um nó que não está compartilhando e por consequência tem baixa reputação, mesmo com a listagem inicial dos arquivos existentes obtida, não receberá as notificações de atualizações dos arquivos, e será obrigado a realizar uma operação *list* novamente, onde sabe-se que sua reputação é verificada antes do envio.

Como consequência, pode-se analisar outra definição importante imposta pelo modelo. Por exemplo, considerando um nó de baixa reputação que não está recebendo as mensagens de notificação e assim executa operações *list* para tentar obter novas informações de arquivos, nota-se que com isso ele estará gerando tráfego desnecessário, pois a forma correta para que o mesmo volte a receber as mensagens ou até mesmo receba mais respostas de listagem é passar a compartilhar mais, e assim aumentar sua reputação. É necessário, em virtude desse comportamento malicioso do nó, impor uma penalização por excesso de operações *list* realizadas, que é controlada pela média em comparação a rede, ou seja, verifica-se qual o tempo médio de execução de operações *list* da rede e se compara ao tempo médio que o nó malicioso está realizando as operações. Caso estas estejam sendo executadas muito mais rapidamente que as operações da rede o valor de reputação do nó sofre uma penalização de 0,05 na sua reputação.

4 VALIDAÇÃO

Com a devida estruturação do modelo TrustLP2P, percebendo as necessidades adequadas com as operações de listagem e solicitação de arquivos, é possível avaliar uma série de questionamentos relacionando diferentes cenários. Dentre essas possibilidades de análise, consideram-se alternativas para descrever o funcionamento do TrustLP2P questões como a quantidade de informações a serem recebidas, a forma como um nó de baixa reputação deve proceder para conseguir aumentar a mesma e até mesmo o controle de nós que executam ações maliciosas. Contudo, a situação mais interessante é a análise do que acontece com o valor de reputação tanto dos nós que são bem comportados como dos egoístas, já que trata um problema basilar de sistemas de reputação em si.

Determinados os possíveis levantamentos para a análise, foi necessário verificar de que forma os testes seriam executados. Em um primeiro instante, a implementação de um protótipo se mostrou uma idéia mais relevante que logo foi impedida por dois fatores: a amostragem, devido a um número pouco expressivo de usuários para utilizar o sistema, e a necessidade de que todos os usuários presentes estivessem realmente utilizando o sistema, ou seja, a obrigatoriedade de controlar o uso do mesmo. Com esses fatores impeditivos, a alternativa evidente passou a ser uma avaliação analítica, e assim partiu-se para modelagem do cenário, considerando todos os parâmetros necessários para que fosse constituído o ambiente de uso da rede onde o TrustLP2P atuaria. A tabela 2 detém o nome desses parâmetros com a descrição do que cada um deles representa.

Tabela 4.1: Descrição dos parâmetros de simulação

Nome	**Descrição**
N	Quantidade total de nós da rede
N_c	Quantidade de nós de compartilham
N_{nc}	Quantidade de nós que não compartilham
N_i	Quantidade de nós interessados
Num_{get}	Quantidade média de *gets* feitos por um nó
Num_{list}	Quantidade média de *lists* feitos por um nó
Num_{add}	Quantidade média de *add* feitos por um nó
Num_{del}	Quantidade média de *del* feitos por um nó
F	Quantidade média arquivos compartilhados por um nó
T_{get}	Tempo médio em que um nó faz um *get* na rede
T_{list}	Tempo médio em que um nó faz um *list* na rede

Assim, com o envolvimento dos cálculos do modelo e o estabelecimento dos parâmetros necessários para estruturar a rede, a etapa de conceituação da validação do modelo se mostra adequada para realizar as análises. Para melhor exemplificar a simulação do TrustLP2P em uma rede, os parâmetros foram ajustados de acordo com dois cenários: o

primeiro representando os laboratórios de uma ambiente acadêmico de pequeno porte e o segundo representando a rede wireless de grande porte utilizada por alunos da graduação, diferenciados por alguns aspectos relevantes para os cálculos. Essa divisão de cenários para as análises das situações é realizada devido a questões de grandeza, e serve para ressaltar a escalabilidade do TrustLP2P em virtude dos parâmetros da rede, possibilitando o sistema de reputação atuar tanto em uma rede local com muitos usuários como em uma com poucos usuários.

4.1 CENÁRIO 1: AMBIENTE ACADÊMICO DE PEQUENO PORTE

O cenário 1, que trata os laboratórios de uma ambiente acadêmico de pequeno porte, descreve basicamente um ambiente de compartilhamento acadêmico utilizado por diversos alunos. Atribuindo essa questão, pode-se analisar que esse ambiente lida com arquivos de artigos, periódicos, pesquisas e trabalhos em geral, além do conteúdo das próprias disciplinas e dos demais projetos desenvolvidos pelos pesquisadores.

Direcionando esse cenário para a modelagem subentende-se um ambiente de utilização variada, ou seja, onde o compartilhamento de arquivos por vezes é frequente e por vezes não, o que implica diretamente em um número relevante de nós interessados. Nota-se também que o número de arquivos compartilhados tende a ser adequadamente grande, uma vez que o tipo de material dos arquivos tem um conteúdo bastante disseminado, levando em consideração, por exemplo, artigos que são usados para referencial teórico nas pesquisas.

Considerando os parâmetros anteriormente descritos, a tabela abaixo mostra os valores que foram associados a cada um deles, de acordo com este primeiro cenário.

Tabela 4.2: Valores dos parâmetros de rede do cenário 1

Parâmetro	**Valor**
N	30
N_c	15
N_{nc}	15
N_i	10
Num_{get}	17
Num_{list}	0,08
Num_{add}	1
Num_{del}	0,5
F	100
T_{get}	0,058
T_{list}	12,5

Os valores descrevem um cenário contendo 30 nós participantes, onde 15 estão com-

partilhando, 15 não estão, e do total, 10 estão interessados nos arquivos compartilhados. Define-se também que os 60 nós realizam em média 17 operações *get* por hora, sendo que cada nó faz uma operação dessas em média a cada 0,058 horas, ou seja, 3 minutos e 48 segundos. Além disso, cada nó compartilha em média 100 arquivos, totalizando 1.500 arquivos compartilhados na rede.

Com o contexto do cenário devidamente definido, são demonstradas nas subseções a seguir as situações especificadas que ressaltam a atuação do TrustLP2P dentro da rede. É importante ressaltar que para efeitos de comparação alguns gráficos apresentam a análise de um nó que compartilha certa quantidade de arquivos e aliado a isso um nó (definido como Nó 2) que representa o nó com comportamento egoísta, ou seja, que não compartilha nada.

4.1.1 Situação 1: Análise do valor de reputação

A primeira situação descreve o que acontece com o valor de reputação de um nó, e nela são avaliados dois nós: o Nó 1 que compartilha a média de arquivos da rede, e o Nó 2 que não compartilha nada. Nessa situação estão envolvidos apenas os cálculos básicos do modelo, como o IC, IP e o $Ativ_{Ni}$.

O gráfico da Figura 4.1 mostra então o que acontece com a reputação tanto do Nó 1 como do Nó 2.

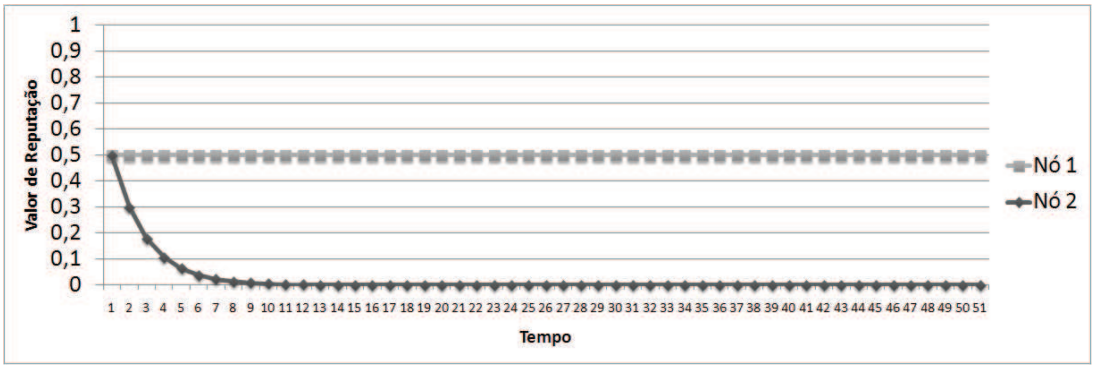

Figura 4.1: Situação 1 - Cenário 1 com parâmetros *default*

No gráfico disposto na Figura 4.1 nota-se que o valor da reputação do Nó 1 se mantém fixo em 0.5, pelo motivo de que o mesmo está compartilhando a média de arquivos da rede e consequentemente faz com que a proporção para o cálculo se mantenha na faixa de 46% à 55%, especificamente em 48,16%. Nota-se que o Nó 2 que não está compartilhando nada, tem sua reputação diminuída rapidamente em virtude da faixa de proporção ser a mínima, o que reforça a atuação da penalização dentro do modelo.

Já o gráfico da Figura 4.2 ilustra a situação 1 porém com a variação na quantidade de

nós interessados (N_i). Nesse caso a quantidade de nós interessados torna a ser o dobro, passando a ser 20.

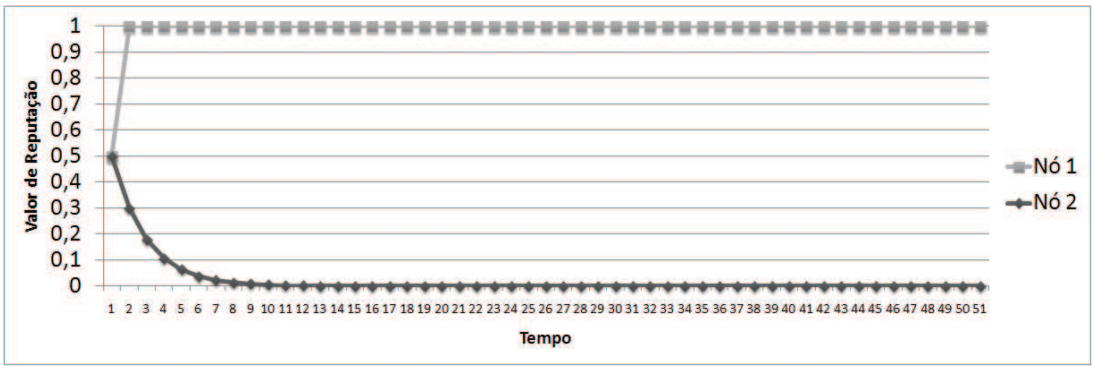

Figura 4.2: Situação 1 - Cenário 1 com aumento de N_i

Percebe-se no gráfico da Figura 4.2 que o aumento do número de nós interessados implicou no aumento da taxa de proporção e consequentemente no crescimento da reputação. Isso se deve pelo fato de que uma rede, contando com mais nós interessados do que normalmente, tem mais transações e então a procura maior auxilia o aumento da reputação.

Por sua vez, o gráfico da Figura 4.3 descreve a situação 1 da mesma forma que explicada no gráfico anterior, porém com a quantidade de nós interessados sendo a metade da quantidade padrão, sendo então 5.

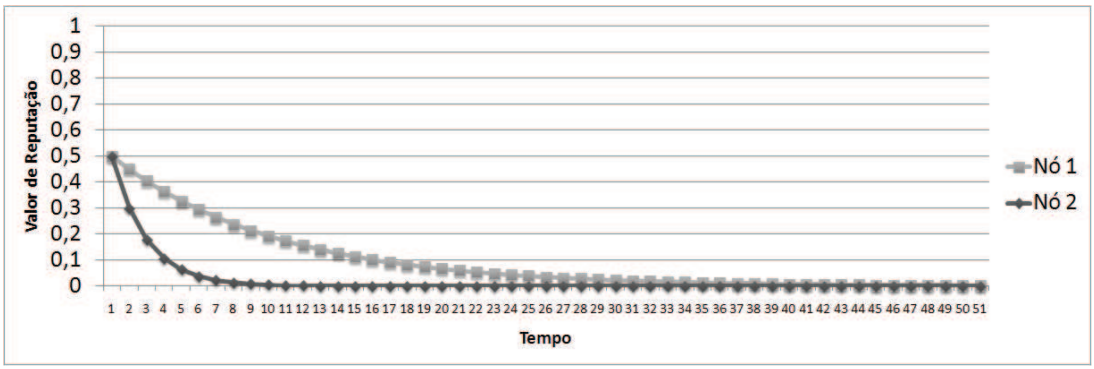

Figura 4.3: Situação 1 - Cenário 1 com diminuição de N_i

O gráfico da Figura 4.3, com a diminuição da quantidade de nós interessados se tem uma queda no valor de reputação, mesmo compartilhando a quantidade de arquivos média da rede. Isso se dá pela mesma justificativa anterior, afinal, com menos interesse mais difícil se torna aumentar o valor de reputação de um nó. A queda da reputação, ilustrada na imagem, implica que o nó irá receber menos respostas para as operações *list* que fizer, tendo acesso a menos informações sobre os arquivos compartilhados. Naturalmente, compreende-se que o tráfego na rede também será menor.

4.1.2 Situação 2: Quantidade de informações recebidas

Na segunda situação é analisada outra questão importante relacionada com o valor de reputação: a quantidade de informações sobre os arquivos que um nó recebe. Nesse caso, além dos cálculos básicos do modelo (IC, IP e o $Ativ_{Ni}$) está envolvido também o cálculo específico Qtd_{ir}. Os gráficos ilustram, na devida proporção, a quantidade de arquivos que um nó recebe de acordo com a sua reputação. Foi especificado que o limite mínimo para a reputação de um nó que quer continuar recebendo as mensagens de notificação é 0,3. Assim, para descrever isso considera-se também que nessa situação 2 um nó compartilha metade da quantidade média de arquivos compartilhados pela rede, ou seja, 50. É necessário ressaltar também que enquanto o nó tem sua reputação igual ou superior a 0,3 e então está recebendo as atualizações, as informações que o mesmo obtém a cada operação *list* que faz envolvem a probabilidade e por isso o nó já começa recebendo apenas metade dos dados sobre os arquivos.

No gráfico da Figura 4.4 é visível então o que acontece com o acesso as informações que o nó tem de acordo com sua reputação.

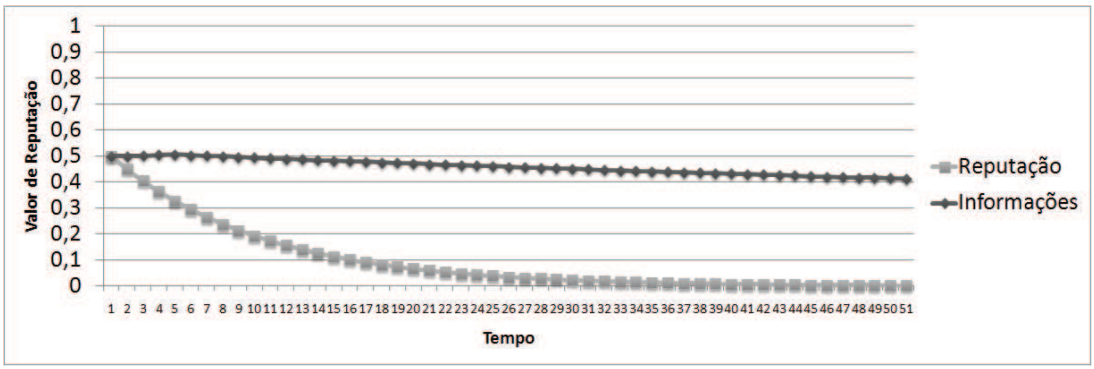

Figura 4.4: Situação 2 - Cenário 1 com parâmetros *default*

O gráfico da Figura 4.4 mostra que, inicialmente, o nó está recebendo informações de 750 dos 1500 arquivos compartilhados na rede. Apenas até o instante de tempo 4 ele continua recebendo as atualizações e se mantém com o mesmo padrão de recebimento. A partir desse momento ele passa a não receber mais as mensagens de notificação e continua possuindo apenas as informações que já tinha conhecimento. Como essas vão a cada hora aumentando, proporcionalmente a relação das informações que ele detém com as informações que estão trafegando se distancia cada vez mais, conforme demonstra a queda no gráfico.

Já no gráfico da Figura 4.5 se tem o aumento do número de nós interessados e o que isso implica na quantidade de informações recebidas.

Percebe-se no gráfico da Figura 4.5 que o aumento do número de nós interessados proporcionou no aumento da taxa de proporção e consequentemente no crescimento da

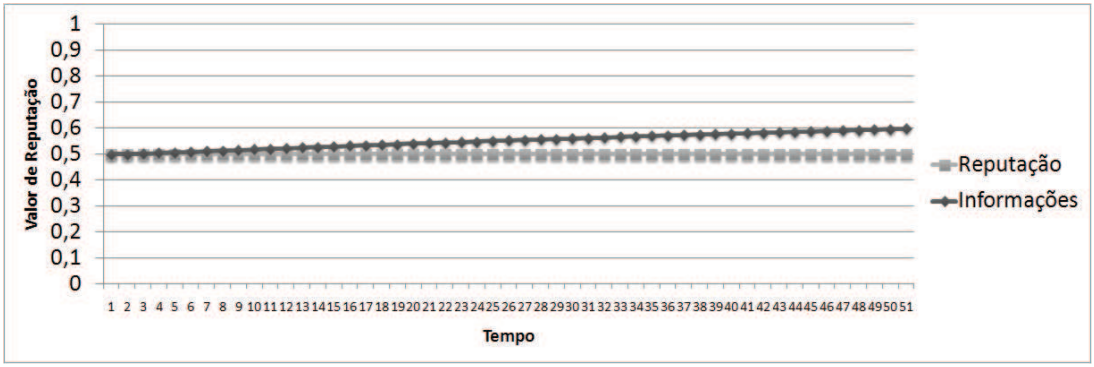

Figura 4.5: Situação 2 - Cenário 1 com aumento de N_i

reputação. Dessa forma, o nó passa a receber as informações sobre os arquivos de forma plena e igualmente crescente, sendo recompensado mesmo compartilhando a quantidade média de arquivos.

O seguinte gráfico exposto na Figura 4.6 apresenta a situação 2 com a diminuição do número de nós interessados.

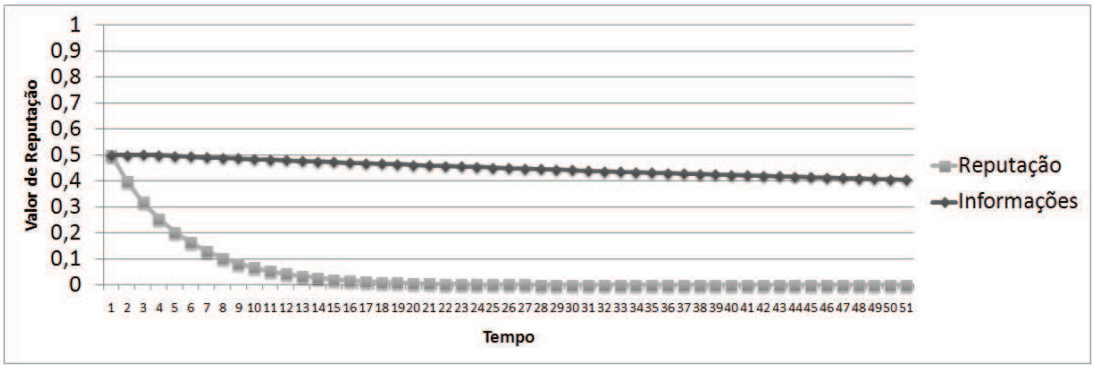

Figura 4.6: Situação 2 - Cenário 1 com diminuição de N_i

Conforme o gráfico da Figura 4.6, a diminuição da quantidade de nós interessados fez, naturalmente, com que o valor de reputação diminuísse mais rapidamente. Isso é perceptível a medida que no cenário padrão da rede o valor de reputação chegava ao seu mínimo no instante de tempo 51, e agora passa então ser no instante de tempo 28, já que o valor da taxa de proporção resultou em 16,06%. Da mesma forma, o nó parou de receber as mensagens de notificação muito mais rapidamente comparado a situação anterior.

4.1.3 Situação 3: Melhora da reputação

Sabendo então a variação e as consequências as quais se relacionam o valor de reputação, é importante estabelecer a forma de, por exemplo, recuperar uma baixa reputação para que o nó não seja prejudicado. Nesse caso, a situação 3 lida basicamente com a mudança do índice de contribuição (IC), sendo o aumento do número de arquivos com-

partilhados a maneira mais adequada e de controle do nó para que sua reputação aumente.

Considerando novamente que o Nó 1 compartilha a média de arquivos da rede e o Nó 2 não compartilha nada, o gráfico da Figura 4.7 descreve a situação padrão dos parâmetros da rede, com alterações em certo instante de tempo. As alterações que ocorrem no decorrer da modelagem representam, nesse primeiro caso, um nó que não é egoísta e está de acordo com a média da rede, que tem sua reputação apenas em metade da máxima e por não receber completamente as informações sobre os arquivos, deseja aumentar o seu valor de reputação para corrigir esse problema.

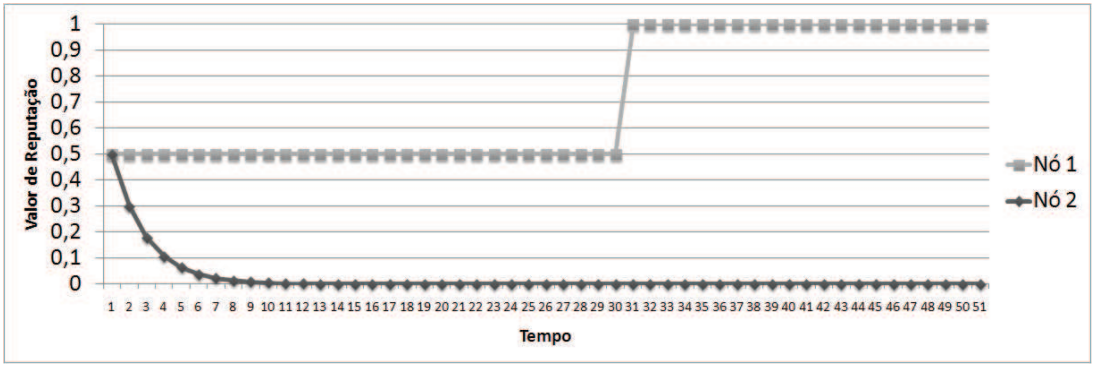

Figura 4.7: Situação 3 - Cenário 1 com parâmetros *default*

No gráfico da Figura 4.7 um nó mantém sua reputação igual a inicial por estar compartilhando a média de arquivos da rede e pelos parâmetros da rede estarem em sua situação padrão. Querendo ter sua reputação melhorada, há a necessidade de que o mesmo passe a compartilhar no mínimo 115 arquivos para que a proporção troque para outra faixa e então passe a aumentar, que é o que acontece a partir do instante de tempo 30.

Como o aumento do número de nós interessados implica diretamente no aumento da reputação, a única variação de parâmetros para essa situação é a diminuição do número de nós interessados, conforme descreve o gráfico da Figura 4.8.

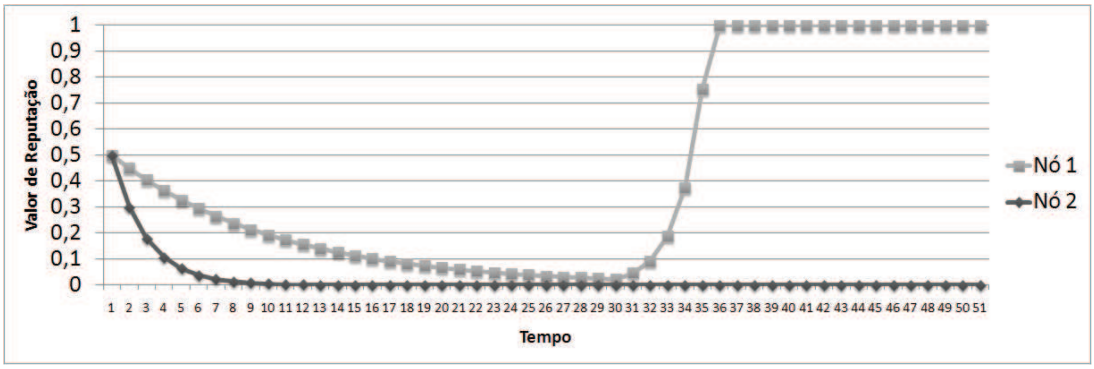

Figura 4.8: Situação 3 - Cenário 1 com diminuição de N_i

A principal diferença descrita no gráfico da Figura 4.8 é o fato da reputação diminuir

e posteriormente aumentar de forma rápida, no instante de tempo em que o nó fornece mais arquivos para compartilhamento e então volta a ser novamente beneficiado. Nesse exemplo percebe-se a maneira dinâmica de como ocorrem as mudanças, onde o nó em poucos instantes de tempo consegue recuperar sua reputação. Essa rapidez e eficiência torna o modelo mais robusto, pois apresenta uma situação próxima à ideal e também estabelece variações que encontram-se dentro do esperado pelo usuário final.

4.1.4 Situação 4: Penalização por excesso de *lists*

A última situação analisada associa os conceitos das demais, por tratar uma funcionalidade do modelo. Considera-se o seguinte exemplo: um nó está compartilhando pouco, tem conhecimento de uma certa quantidade de informações dos arquivos e sua reputação está diminuindo a cada instante de tempo. A partir do momento em que o mesmo parar de receber as mensagens de notificação, será obrigado a fazer operações *list* para obter novas informações sobre arquivos. Como um nó ao fazer uma operação *list* tem sua reputação analisada do cálculo da probabilidade, de pouco adiantará esse nó em questão executar *lists*, já que continuará recebendo pouco ou quase nada de dados sobre arquivos compartilhados. Contudo, caso esse não tome a medida de aumentar o número de arquivos que compartilha para então acrescer sua reputação, ele continuará na tentativa de executar operações *list* para sanar seu problema. Para controlar isso, é verificado se um nó está fazendo esse tipo de operação muito rapidamente, pois então ele é penalizado em caso de resposta afirmativa em -0,05. Na situação 4 são considerados todos os cálculos presentes no modelo: IC, IP, $Ativ_{Ni}$, Qtd_{ir} e Pen_{el}.

O gráfico da Figura 4.9 ilustra uma situação onde o nó é penalizado por excesso de *lists*. Assim como na situação apresentada na subseção 4.1.2, os valores do eixo y do gráfico para a linha da quantidade de informações são estipulados proporcionalmente ao valor de reputação, ou seja, se o nó em questão está recebendo informações sobre todos arquivos compartilhados, o valor representativo será 1.

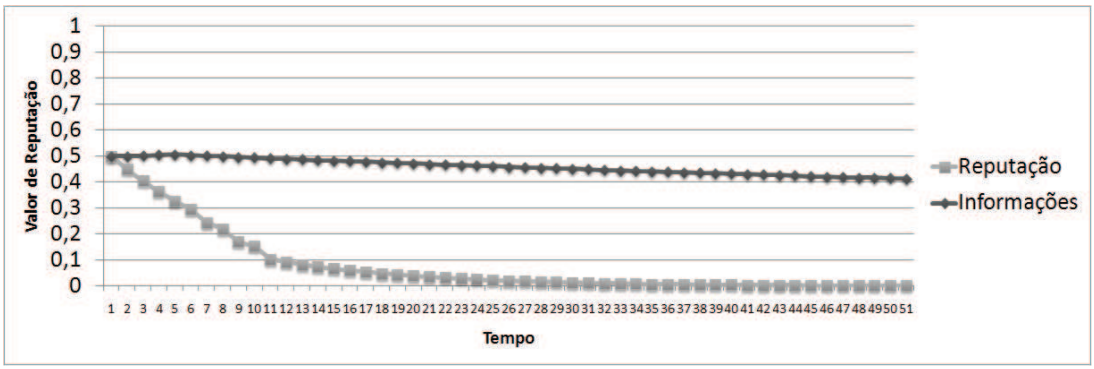

Figura 4.9: Situação 4 - Cenário 1 com parâmetros *default*

Em um primeiro momento, sabe-se que um nó faz uma operação *list* em média a cada 12 horas. Com a reputação diminuindo e o acesso as informações sendo restrito, o nó executa outros *lists* e é penalizado nos instantes de tempo 6, 8 e 10, conforme é apresentado no gráfico da Figura 4.9. Essa penalização se deve ao fato de que o valor da variável Pen_{el} ficou negativo, e passou de -2, limite considerável para a variação de execução de operações de listagem.

Quando diminuída a quantidade de nós interessados, conforme o gráfico da Figura 4.10, o nó tende a executar as operações *list* em excesso mais rapidamente.

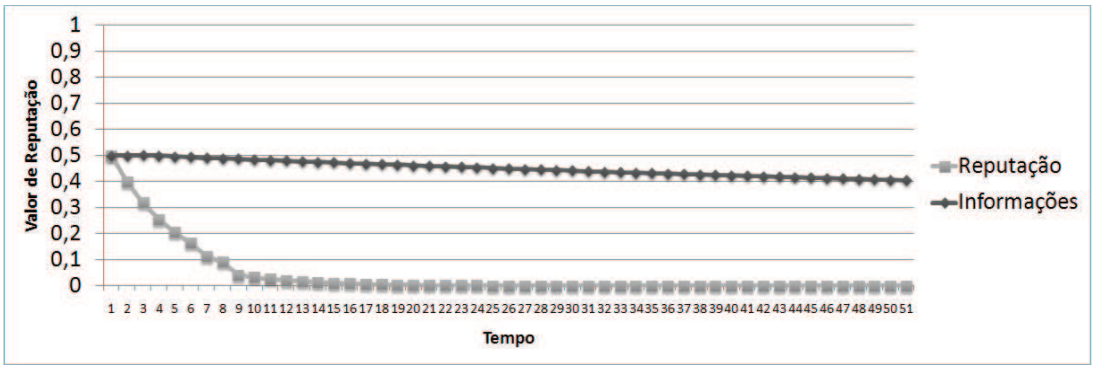

Figura 4.10: Situação 4 - Cenário 1 com diminuição de N_i

O gráfico da Figura 4.10 mostra que, como a execução das operações de listagem foi mais rápida, a penalização se deu nos instantes de tempo 4, 6 e 8, fazendo com que a reputação do nó chegasse ao seu valor mínimo no instante de tempo 22.

4.2 CENÁRIO 2: REDE WIRELESS DE GRANDE PORTE

No cenário 2, onde é especificado a rede wireless de grande porte utilizada por alunos da graduação, descreve-se um ambiente de foco diferente em relação ao cenário 1. É notável que nesse cenário o compartilhamento é realizado com arquivos como músicas, vídeos, imagens e demais tipos variados de documentos.

Realizando a modelagem desse cenário abstrai-se a idéia de um ambiente de alto uso do compartilhamento, o que é justificado pelo fato de que os arquivos são de interesse comum para grande parte dos nós. Além disso, considera-se que o número de arquivos compartilhados também é alto, devido a quantidade de nós que compartilham. A tabela 4 mostra os valores que foram associados a cada parâmetro para este segundo cenário.

Nesse cenário, os valores descrevem um ambiente que contém 300 nós participantes, onde 250 estão compartilhando, 50 não estão, e 150 estão interessados nos arquivos compartilhados, o que entra em acordo com a conceituação anterior. Assim como no cenário 1, define-se que os 300 nós realizam em média 30 operações *get*, onde cada nó executa esse

Tabela 4.3: Valores dos parâmetros de rede do cenário 2

Parâmetro	Valor
N	300
N_c	250
N_{nc}	50
N_i	150
Num_{get}	30
Num_{list}	0,08
Num_{add}	5
Num_{del}	1,5
F	400
T_{get}	0,033
T_{list}	12,5

tipo operação em média a cada 0,033 horas, ou seja, aproximadamente 2 minutos. Por fim, cada nó compartilha em média 400 arquivos, totalizando 100000 arquivos compartilhados na rede.

Assim como na análise do cenário 1, é realizada a mesma análise quanto as situações descritas anteriormente, com o intuito de mostrar a aplicabilidade do sistema em diferentes ambientes LAN.

4.2.1 Situação 1: Análise do valor de reputação

Para a situação 1 que trata o que acontece com o valor de reputação de um nó, foram gerados os gráficos abaixo com suas comparações em relação ao cenário 1 descritas na sequência.

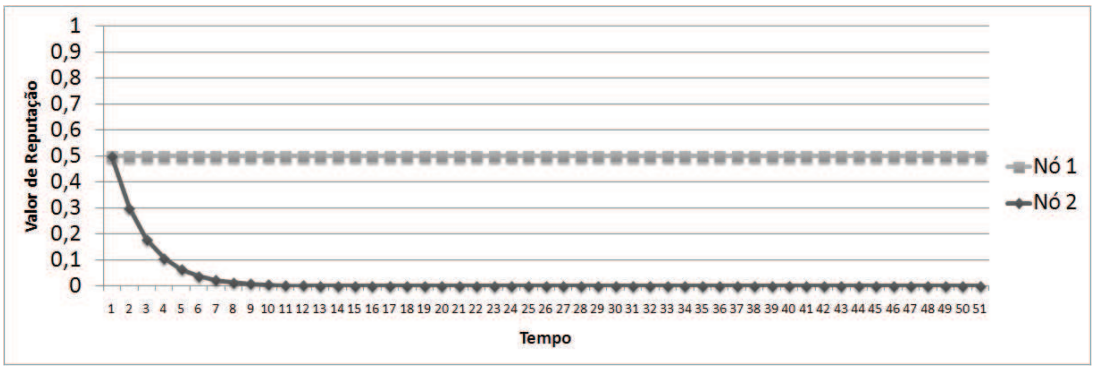

Figura 4.11: Situação 1 - Cenário 2 com parâmetros *default*

O gráfico da Figura 4.11 apresenta o mesmo comportamento da situação 1 quando aplicado no primeiro cenário. Com o Nó 1 compartilhando a média, compreende-se que o mesmo deve manter sua reputação igual a inicial, da mesma forma que o Nó 2 que não compartilha nada tem a sua reputação diminuída rapidamente. Contudo, a principal dife-

rença dessa situação entre os dois cenários está implícita nos valores da taxa de proporção (TP), onde no primeiro cenário o nó que compartilha a média tinha TP resultando em 48,16% e agora nesse segundo cenário a variável TP resultou em 54%. Naturalmente, observa-se que essa diferença vem da estruturação dos parâmetros da rede, devido a sua escala.

O gráfico da Figura 4.12 mostra a situação 1 com o aumento do número de nós interessados.

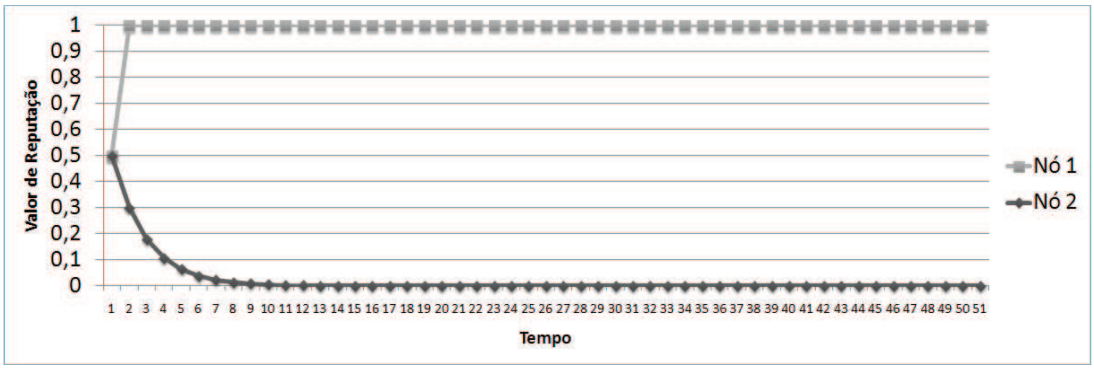

Figura 4.12: Situação 1 - Cenário 2 com aumento de N_i

Na Figura 4.12 nota-se no gráfico que mesmo com um aumento proporcional do número de nós interessados, estabeleceu-se uma diferença nas taxas de proporção se comparados os dois cenários. Enquanto no cenário 1, com o aumento da quantidade de N_i, a taxa de proporção resultou em 96,33%, o cenário 2 teve a variável TP resultando em 72%, o que não indica variação no gráfico já que o valor de reputação chegou ao seu limite máximo logo no início, apesar das diferentes faixas de proporção.

Já com a diminuição da quantidade de nós interessados, a comparação entre os dois cenários não apresentou diferenças notáveis, conforme visto no gráfico da Figura 4.13. Os valores para variável TP resultaram em 24,08% para o cenário 1 e 27% para o cenário 2, ficando então na mesma faixa de proporção, o que implicou no mesmo comportamento para ambos os cenários.

4.2.2 Situação 2: Quantidade de informações recebidas

Para com a segunda situação, que trata a quantidade de informações sobre os arquivos que um nó recebe, são obtidos resultados interessantes no cenário 2 quando comparado ao cenário 1, conforme se pode ver já no gráfico da Figura 4.14.

Inicialmente, nota-se no gráfico da Figura 4.14 resultados semelhantes aos expostos na mesma situação no cenário 1, porém, há diferenças consideráveis. Nos primeiros instantes de tempo o comportamento descrito no gráfico se mantém igual, afinal o nó ainda recebe

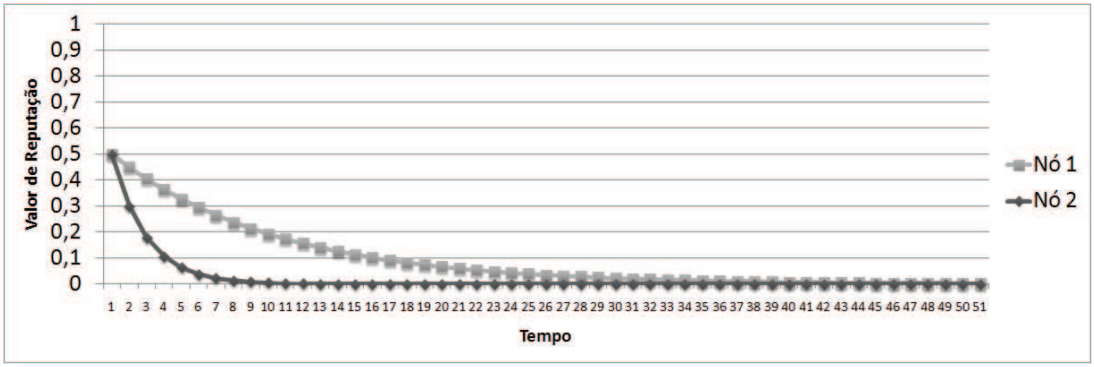

Figura 4.13: Situação 1 - Cenário 2 com diminuição de N_i

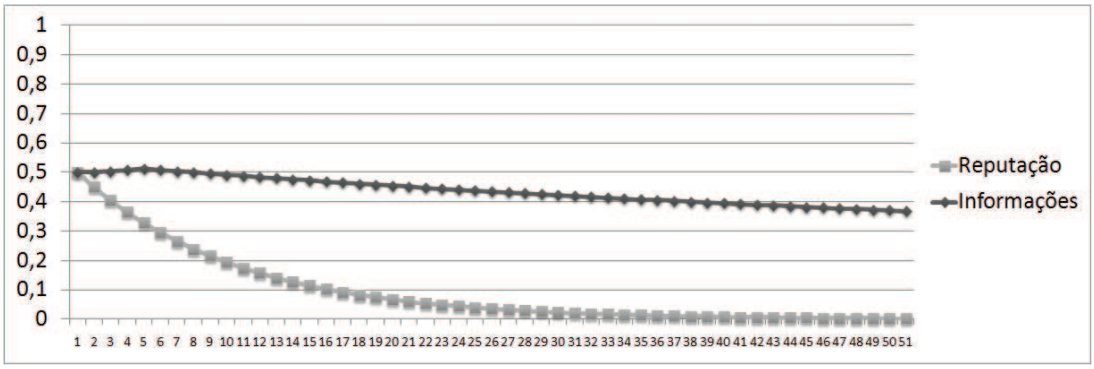

Figura 4.14: Situação 2 - Cenário 2 com parâmetros *default*

as mensagens de notificação. Além disso, é importante ressaltar que as taxas de proporção dos dois cenários estão na mesma faixa, onde TP no cenário 1 é 32,11% e no cenário 2 é 45%. A principal mudança que acontece no cenário 2 é a maior rapidez na diminuição da quantidade de informações recebidas, o que se pode notar no instante de tempo 51, no qual proporcionalmente tem-se o valor de 0,3683 no cenário 2 e 0,4136 no cenário 1. Isso se deve basicamente a grande quantidade de informações que o cenário 2 trata, afinal, quando o nó passa a não receber mais as atualizações de arquivos o total de Qtd_{ir} é 52.625, ao passo que no cenário 1, no mesmo momento, a variável resulta em 772,5.

Considerando essas diferenças proporcionais, percebe-se também no gráfico da Figura 4.15, com o aumento do número de nós interessados, uma situação como a anterior.

Assim como visto anteriormente, a estrutura dos parâmetros da rede implica em uma leve diferença na propoção que lida com valor da variável Qtd_{ir}. Por exemplo, conforme se nota no gráfico da Figura 4.15, o nó no último instante de tempo tem o valor de $Qtd_{ir} = 92.875$ enquanto o padrão da rede está com $Qtd_{ir} = 142.875$, o que é descrito no gráfico com o valor de 0,65. Já no cenário 1, na mesma situação se tem para o nó em análise a variável $Qtd_{ir} = 1.118$ ao passo que para a rede o valor de $Qtd_{ir} = 1.868$, gerando no gráfico o valor de 0,594, menor que em relação a este cenário.

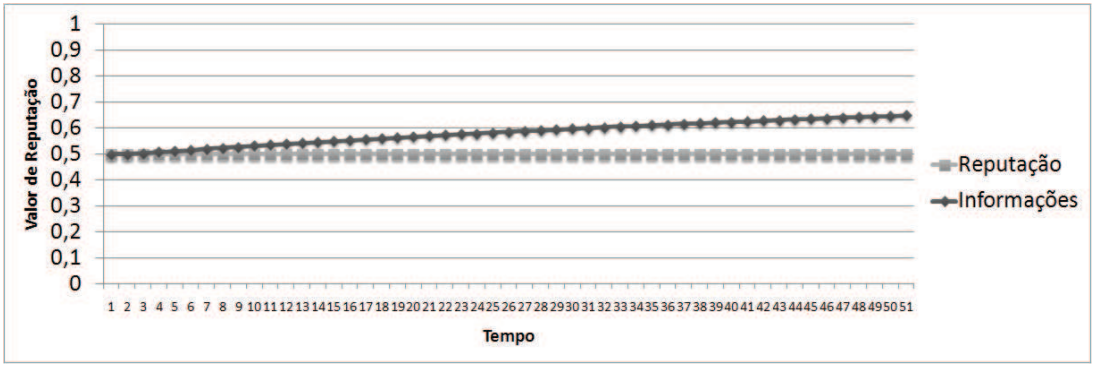

Figura 4.15: Situação 2 - Cenário 2 com aumento de N_i

Realizando essas comparações com a variável Qtd_{ir}, subentende-se que o comportamento com a diminuição da quantidade de nós interessados vai implicar na mesma diferença, conforme exposto no gráfico da Figura 4.16.

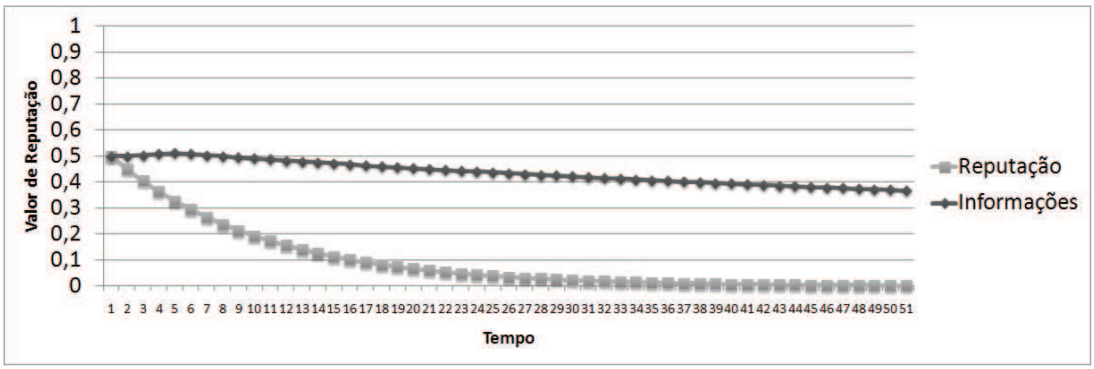

Figura 4.16: Situação 2 - Cenário 2 com diminuição de N_i

Novamente, com a diminuição do número de nós interessados, segundo o gráfico da Figura 4.16, o nó em questão em certo instante de tempo não recebe mais as mensagens de notificação. Com isso, há uma diferença maior no recebimento, já que no último instante de tempo se tem $Qtd_{ir} = 53.500$ enquanto o padrão da rede está com $Qtd_{ir} = 142.875$, descrito no gráfico com o valor de 0,3745. Comparado ao cenário 1, na mesma situação se tem $Qtd_{ir} = 757,5$ ao passo que para a rede o valor de $Qtd_{ir} = 1.868$, exposto no gráfico com o valor de 0,406.

4.2.3 Situação 3: Melhora da reputação

Para a situação 3, que estabelece uma alternativa para um nó de baixa reputação conseguir aumentar a mesma, é importante perceber que no cenário 2 a quantidade de arquivos é quatro vezes maior do que comparado ao cenário 1, mesmo que os demais parâmetros equilibrem a estruturação da LAN e mantenham semelhante a análise do cenário 1.

Trabalhando em torno da variável IC e considerando a situação 1 deste cenário, se nota claramente a facilidade com que um nó que está compartilhando a média tem em aumentar sua reputação, conforme se vê no gráfico da Figura 4.17.

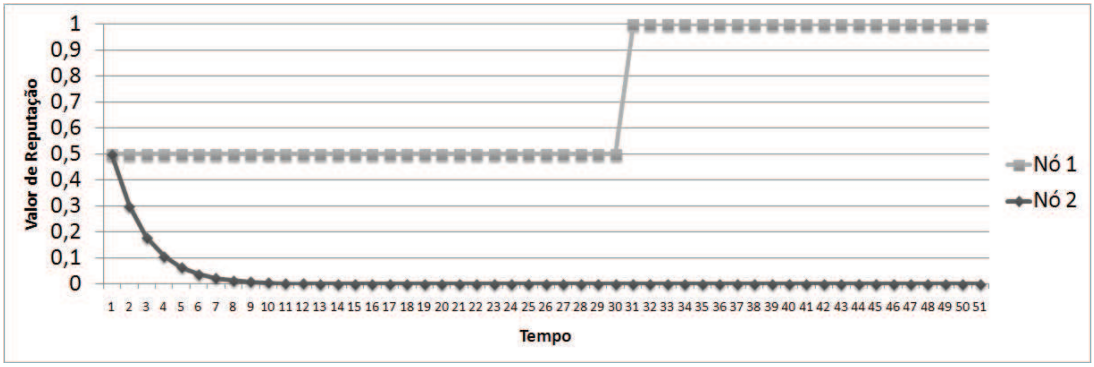

Figura 4.17: Situação 3 - Cenário 2 com parâmetros *default*

O nó compartilhando a média mantém sua reputação em 0,5, conforme o gráfico da Figura 4.17, e no instante de tempo 20 passa a compartilhar mais arquivos e por consequência aumenta sua reputação. Neste cenário 2, o nó começa então compartilhando 400 arquivos, se mantendo com a média. A partir desse instante 20, ele passa a compartilhar apenas 35 arquivos a mais, em um total de 435. Esse número de arquivos é definido nessa quantidade pois a partir desse valor, a proporção que era de 54% quando o nó compartilhava a média e passou a ser de 55,431% depois, alterando assim o nível da faixa de proporções e naturalmente, começando a aumentar.

Assim como descrito anteriormente no cenário 1, a única variação de parâmetros para situação 3 é a diminuição do número de nós interessados, apresentada no gráfico da Figura 4.18.

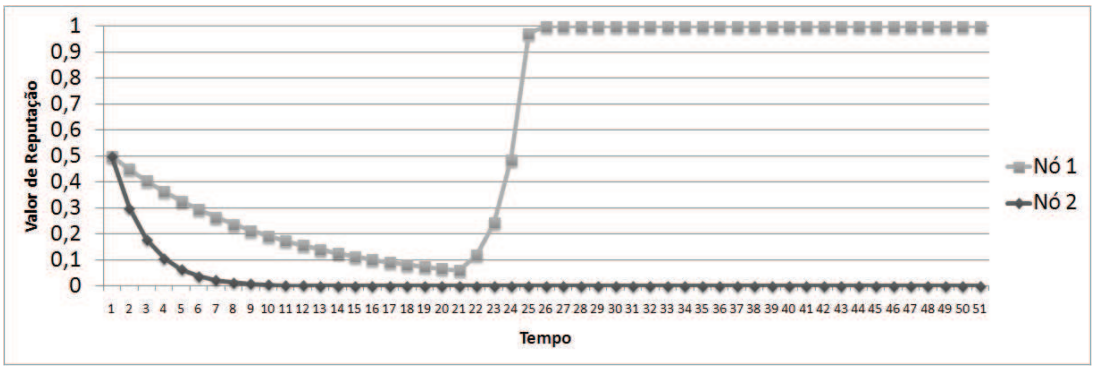

Figura 4.18: Situação 3 - Cenário 2 com diminuição de N_i

Nesse caso ilustrado no gráfico da Figura 4.18, mesmo com o nó compartilhando a quantidade média de arquivos da rede sua reputação cai devido ao baixo número de nós interessados, situação que já foi ilustrada anteriormente. Inicialmente então, o valor da

taxa de proporção é 27%, até o instante de tempo 20 quando o nó passa a compartilhar mais arquivos. Para que a reputação volte a aumentar, considerando os parâmetros da rede, é necessário que o nó compartilhe no mínimo 470 arquivos, e então a taxa de proporção passará a ser 55,027%. Em comparação ao cenário 1, o valor da taxa de proporção é maior do que os 24,08% que resulta nesse cenário inicial, mesmo que não existam diferenças consideráveis, já que a faixa de proporção que contém 24,08% e 27% é a mesma.

4.2.4 Situação 4: Penalização por excesso de *lists*

Na situação 4, que trata a penalização de um nó por esse executar muitas operações *list*, o cenário 2 não apresenta variações quanto a variável Pen_{el}, e sim da penalização em si atuando juntamente com a queda mais rápida do valor de reputação em função dos parâmetros da rede. Considerando a situação padrão da rede, o gráfico da Figura 4.19 apresenta a idéia de penalização para o cenário 2.

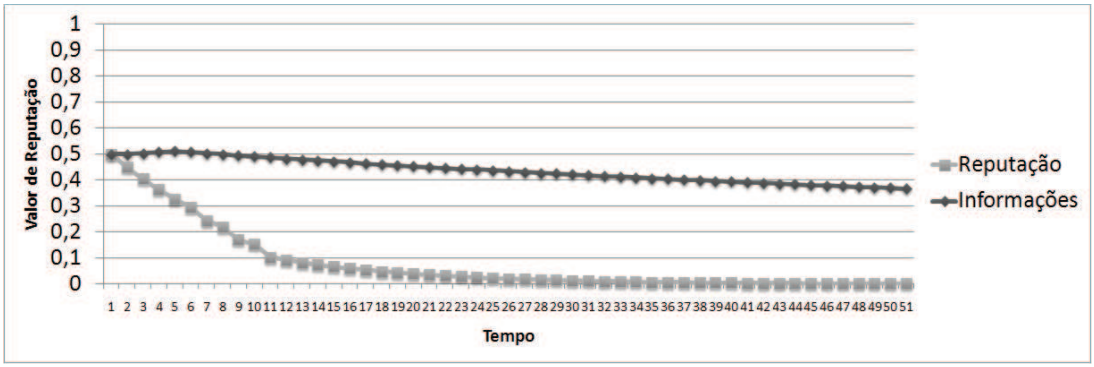

Figura 4.19: Situação 4 - Cenário 2 com parâmetros *default*

Nota-se no gráfico da Figura 4.19 que a reputação está diminuindo em virtude da taxa de proporção ser de 45% e que o nó parou de receber as mensagens de notificação no instante de tempo 5. No momento 6, 8 e 10, o nó em questão executa a operação *list* para logo obter novas informações sobre os arquivos, devido ao ambiente de rede que é bastante dinâmico. Por consequência, nesses três instantes de tempo o nó é penalizado em -0,05 fazendo com que sua reputação diminua mais rapidamente.

Já em uma outra análise, a alteração na quantidade de nós interessados apresentou mudanças mais relevantes, conforme o gráfico da Figura 4.20.

Nesse caso, foi estipulado um número mais baixo de nós interessados do que o normalmente avaliado nas outras situações descritas anteriormente, assim definindo o valor de N_i em 50, para que se obtivesse uma taxa de proporção 19,5%. Com isso, logo no tempo 4 o nó já executou uma operação *list* por não estar recebendo as atualizações, e assim também fez nos tempos 5, 6 e 8. Na ilustração percebe-se que não há uma variação

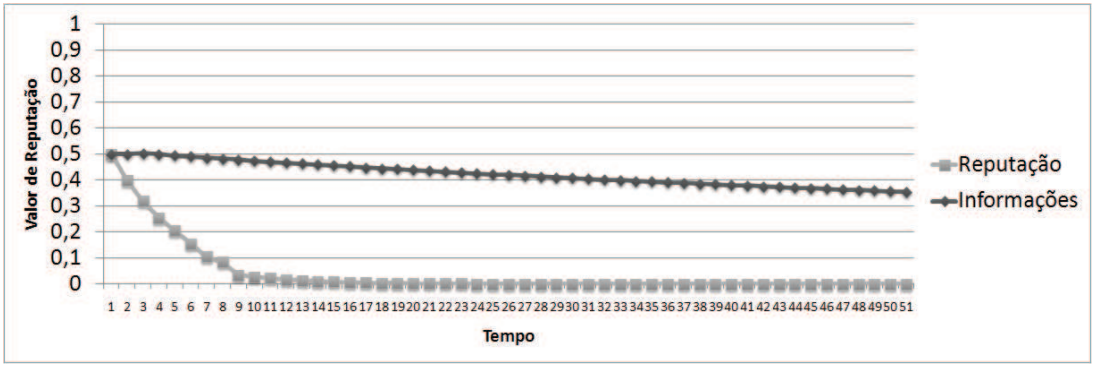

Figura 4.20: Situação 4 - Cenário 2 com diminuição de N_i

tão grande entre a penalização por excesso de *lists* e a própria diminuição da reputação. Contudo, com a penalização, logo no instante de tempo 12 o valor de reputação chegou em 0, muito mais rapidamente comparado ao cenário 1.

4.3 SÍNTESE DA VALIDAÇÃO

A etapa de validação do sistema de reputação TrustLP2P consiste em uma modelagem através de parâmetros para que fosse possível simular possíveis situações de comportamento dos nós. Essa definição de modelagem é um aspecto notoriamente relevante devido as análises efetuadas, considerando que algumas delas necessitariam de longo tempo para serem realizadas caso fosse implementado um protótipo do sistema. Com a estruturação correta da avaliação analítica, percebe-se maior consistência nas definições basilares do sistema de reputação, ressaltando os devidos cuidados tomados para que fossem tratados grande parte dos problemas e características do ambiente em que o mesmo se aplica.

As situações analisadas são dispostas na devida ordem com o intuito de apresentar o que acontece de acordo com as ações dos nós participantes. Em um primeiro instante verifica-se a variação do valor de reputação em função da quantidade de arquivos que o nó compartilha. Essa situação trata um aspecto crucial dentro do sistema que é o próprio valor de reputação. Contudo, as mudanças realizadas nas variáveis tem como seu principal foco a alteração na quantidade de nós interessados, afinal isso ilustra que o aumento/diminuição do valor de reputação não depende única e exclusivamente do número de arquivos que o nó fornece para rede, pois esse número varia com o cenário de participantes interessados e que estão compartilhando.

Já a análise da quantidade de informações recebidas pelo nó e a forma de melhorar a reputação são situações descritas para explicar que um nó que pouco compartilha não terá acesso as informações sobre demais arquivos, porém a alternativa de ação rápida para sanar esse problema é apenas o mesmo passar a compartilhar a quantidade adequada de

arquivos para o cenário disposto. Condiz com a questão de recompensa e penalização que se toma como um dos objetivos centrais do trabalho, ao passo que o controle sobre a baixa reputação limita o conhecimento do nó sobre certos arquivos e também o aumento da reputação fornece subsídio para que o nó passe a receber as mensagens de notificação.

Por fim, a análise de penalização por excesso de operações *list* é uma tratativa que engloba todo o funcionamento do sistema de reputação assim como as demais situações anteriormente analisadas. Tem-se a limitação do acesso as informações sobre os arquivos, devido a baixa reputação do nó, a verificação do valor de reputação durante o processo dos *lists* e ainda a atuação de uma forma de controle sobre o excesso das mensagens de listagem, realizadas a fim de se obter as informações antes não recebidas pelo nó egoísta.

Assim, considerando que as situações ilustram e descrevem adequadamente o modelo, percebe-se naturalmente também que a atuação do mecanismo de recompensa e penalização se mostrou notável dentro do contexto analisado. A opção por não limitar a vazão de dados e sim o acesso as informações sobre os arquivos tornou-se a alternativa correta que é evidenciada pelos gráficos apresentados nos dois cenários. Dessa forma, a etapa de validação ratifica a importância do objetivo do sistema de reputação e seus conceitos dentro do ambiente de redes locais.

5 CONSIDERAÇÕES FINAIS

O TrustLP2P é um sistema de reputação que visa recompensar os nós por compartilharem arquivos, auxiliando a rede através de seus bons comportamentos. Adequar essa recompensa com algum processo, de maneira eficaz, é o principal objetivo do sistema para estabelecer uma justificativa interessante para seu desenvolvimento. Delimitada a questão de pesquisa e levando em conta que a probabilidade de respostas para as respectivas consultas afeta diretamente a percepção do usuário, a atribuição da reputação com esse aspecto torna-se devidamente notável. Como tradicionalmente em sistemas P2P a penalização se dá pela limitação de banda, e isso não é interessante quando se trata de um ambiente local, limitar o acesso as informações sobre os arquivos acabou sendo uma alternativa adequada e eficiente como objetivo do modelo de reputação. Então, a modelagem e aplicação do TrustLP2P tem por intuito preencher a falta de mais pesquisas em torno do uso de sistemas P2P em redes locais, e detalhadamente, de formas de controle e segurança dentro desse contexto.

Com a devida padronização dos parâmetros, tanto os que descrevem a rede e os cenários como os que fazem parte do TrustLP2P, a análise do valor de reputação através de gráficos ilustra bem o funcionamento do sistema de reputação. Já a variação do parâmetro que representa os nós interessados (N_i) fornece conclusões adicionais acerca desse funcionamento, em virtude desse parâmetro atuar diretamente nos cálculos estabelecidos. Os cálculos foram projetados no intuito de tratar essas questões de variação dos parâmetros de forma a retratar o que realmente acontece com a mudança das variáveis. Por exemplo, nota-se que quando o número de nós interessados é modificado, tem-se mudanças relevantes já que essa variável atua diretamente no cálculo da taxa de proporção (TP).

Já a divisão de dois cenários para as análises se mostra interessante devido a questões de grandeza, justificado pelo fato de que, no cenário 1 se tem apenas 30 nós participantes, enquanto o cenário 2 é formado por 300 nós participantes. Isso serve para descrever que o TrustLP2P se torna adaptável de acordo com a correta atribuição dos parâmetros, atuando tanto em uma rede local com muitos usuários como em uma com poucos usuários. Além disso, percebe-se que essas variações da reputação implicam diretamente na necessidade de colaboração dos nós na rede, para que seja mantida a utilização adequada da mesma.

Com as situações apresentadas em todos os gráficos percebe-se claramente que o TrustLP2P trata os diversos comportamentos de nós, tanto os que compartilham e são nós generosos como os nós egoístas. Isso pode ser verificado a medida que um nó egoísta que compartilha menos que a média da rede começa a ter sua reputação diminuída, passa a não receber mais as mensagens de notificação com as atualizações dos demais participantes da rede, começa a executar operações *list* para sanar esse problema e então tem sua reputação

analisada através do cálculo de probabilidade. Ou seja, ná há alternativa evidente para um nó egoísta ter acesso as informações completas da rede e participar ativamente se esse não contribuir com sistema e compartilhar seus arquivos.

A idéia de desenvolver um sistema de reputação engloba questões de segurança mas principalmente lida as inúmeras ações que um nó participante pode executar. Compreende-se que para o usuário final sem um maior entendimento sobre o objetivo do LP2P, que busca um determinado arquivo, pouco importa a iniciativa de participar do compartilhamento fornecendo seus arquivos, e sim apenas o fato do mesmo conseguir obter o arquivo desejado. Dessa forma, o TrustLP2P limita e controla adequadamente o funcionamento do sistema em geral, para que esse usuário tenha interesse em compartilhar ao mesmo tempo que, enquanto sua reputação é baixa, ele não desista de fazer uso do mesmo. Assim, mostra-se importante a utilização do TrustLP2P dentro do devido ambiente P2P, de forma a incentivar a participação dos nós constantemente e fazer com que todo o funcionamento seja devidamente controlado em diversos aspectos de segurança.

BIBLIOGRAFIA

ADAR, E.; HUBERMAN, B. A. Free riding on gnutella. *First Monday*, v. 5, n. 10, 2000.

ANDROULAKI, E. et al. Reputation systems for anonymous networks. In: *PETS '08: Proceedings of the 8th international symposium on Privacy Enhancing Technologies*. Berlin, Heidelberg: Springer-Verlag, 2008. p. 202–218. ISBN 978-3-540-70629-8.

CARUSO, C. A. A.; STEFFEN, F. D. *Segurança em informática e de informações*. São Paulo: SENAC, 1999.

CHEN, H.; CHEN, G. A resource-based reputation rating mechanism for peer-to-peer networks. In: *GCC '07: Proceedings of the Sixth International Conference on Grid and Cooperative Computing*. Washington, DC, USA: IEEE Computer Society, 2007. p. 535–541. ISBN 0-7695-2871-6.

CIGNO, R. L. et al. Peer-to-peer beyond file sharing: Where are p2p systems going? *Parallel and Distributed Processing Symposium, International*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 1–8, 2009.

COHEN, B. Incentives build robustness in bittorrent. In: *Proc. of First Workshop on Economics of Peer-to-Peer Systems*. New York, NY, USA: [s.n.], 2003.

CORNELLI, F. et al. Choosing reputable servents in a p2p network. In: *WWW '02: Proceedings of the 11th international conference on World Wide Web*. New York, NY, USA: ACM, 2002. p. 376–386. ISBN 1-58113-449-5.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. *Distributed Systems: Concepts and Design (4th Edition) (International Computer Science)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321263545.

DAMIANI, E. et al. 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*. New York, NY, USA: ACM, 2002. p. 207–216. ISBN 1-58113-612-9.

DIAS, C. *Segurança e auditoria da tecnologia da informação*. Rio de Janeiro: Axcel Books, 2000.

DINGLEDINE, R.; FREEDMAN, M. J.; MOLNAR, D. The free haven project: distributed anonymous storage service. In: *International workshop on Designing privacy enhancing technologies*. New York, NY, USA: Springer-Verlag New York, Inc., 2001. p. 67–95. ISBN 3-540-41724-9.

FELDMAN, M. et al. Robust incentive techniques for peer-to-peer networks. In: *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*. New York, NY, USA: ACM, 2004. p. 102–111. ISBN 1-58113-711-0.

FIGG, W.; ZHOU, Z. A computer forensics minor curriculum proposal. *J. Comput. Small Coll.*, Consortium for Computing Sciences in Colleges, , USA, v. 22, n. 4, p. 32–38, 2007. ISSN 1937-4771.

GUPTA, M.; JUDGE, P.; AMMAR, M. A reputation system for peer-to-peer networks. In: *NOSSDAV '03: Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*. New York, NY, USA: ACM, 2003. p. 144–152. ISBN 1-58113-694-3.

HAM, M.; AGHA, G. Ara: A robust audit to prevent free-riding in p2p networks. In: *P2P '05: Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing*. Washington, DC, USA: IEEE Computer Society, 2005. p. 125–132. ISBN 0-7695-2376-5.

HOFFMAN, K.; ZAGE, D.; NITA-ROTARU, C. A survey of attack and defense techniques for reputation systems. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 42, n. 1, p. 1–31, 2009. ISSN 0360-0300.

KAMVAR, S. D.; SCHLOSSER, M. T.; GARCIA-MOLINA, H. The eigentrust algorithm for reputation management in p2p networks. In: *WWW '03: Proceedings of the 12th international conference on World Wide Web*. New York, NY, USA: ACM, 2003. p. 640–651. ISBN 1-58113-680-3.

KUBIATOWICZ, J. et al. Oceanstore: an architecture for global-scale persistent storage. *SIGPLAN Not.*, ACM, New York, NY, USA, v. 35, n. 11, p. 190–201, 2000. ISSN 0362-1340.

LEE, S. Y. et al. A reputation management system in structured peer-to-peer networks. In: *WETICE '05: Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise*. Washington, DC, USA: IEEE Computer Society, 2005. p. 362–367. ISBN 0-7695-2362-5.

LEIBOWITZ, N.; RIPEANU, M.; WIERZBICKI, A. Deconstructing the kazaa network. In: *WIAPP '03: Proceedings of the The Third IEEE Workshop on Internet Applications*. Washington, DC, USA: IEEE Computer Society, 2003. p. 112. ISBN 0-7695-1972-5.

LIAU, C. Y. et al. Efficient distributed reputation scheme for peer-to-peer systems. In: *In Proceedings of the 2nd International Human.Society@Internet Conference (HSI), volume LNCS 2713*. [S.l.]: Springer, 2003. p. 54–63.

MARTI, S.; GARCIA-MOLINA, H. Limited reputation sharing in p2p systems. In: *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*. New York, NY, USA: ACM, 2004. p. 91–101. ISBN 1-58113-711-0.

MARTI, S.; GARCIA-MOLINA, H. Taxonomy of trust: categorizing p2p reputation systems. *Comput. Netw.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 50, n. 4, p. 472–484, 2006. ISSN 1389-1286.

OOI, B. C.; LIAU, C. Y.; TAU, K.-L. Managing trust in peer-to-peer systems using reputation-based techniques. In: *In Proc. of WAIM*. [S.l.: s.n.], 2003. p. 2–12.

PIATEK, M. et al. One hop reputations for peer to peer file sharing workloads. In: *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2008. p. 1–14. ISBN 111-999-5555-22-1.

RESNICK, P. et al. Reputation systems. *Commun. ACM*, ACM, New York, NY, USA, v. 43, n. 12, p. 45–48, 2000. ISSN 0001-0782.

ROCHA, É. d. S.; MARCON, D.; ÁVILA, R. B. Comunicação peer-to-peer aplicada a redes locais. In: *ERRC10: Anais da 8° Escola Regional de Redes de Computadores*. Alegrete, RS, Brasil: [s.n.], 2010.

ROSENTHAL, D. S. H. et al. Economic measures to resist attacks on a peer-to-peer network. In: *In Proceedings of the Workshop on Economics of Peer-to-Peer Systems*. [S.l.: s.n.], 2003. p. 2–12.

SELCUK, A. A.; UZUN, E.; PARIENTE, M. R. A reputation-based trust management system for p2p networks. In: *CCGRID '04: Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid*. Washington, DC, USA: IEEE Computer Society, 2004. p. 251–258. ISBN 0-7803-8430-X.

SÊMOLA, M. *Gestão da segurança da informação: uma visão executiva*. Rio de Janeiro: Campus, 2003.

SRIVATSA, M.; XIONG, L.; LIU, L. Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks. In: *WWW '05: Proceedings of the 14th international conference on World Wide Web*. New York, NY, USA: ACM, 2005. p. 422–431. ISBN 1-59593-046-9.

STALLINGS, W. *Criptografia e segurança de redes*. São Paulo: Pearson Prentice Hall, 2008.

TANENBAUM, A. S.; STEEN, M. V. *Distributed Systems: Principles and Paradigms*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN 0130888931.

XIONG, L.; LIU, L. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. on Knowl. and Data Eng.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 16, n. 7, p. 843–857, 2004. ISSN 1041-4347.

ZHOU, R.; HWANG, K. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Trans. Parallel Distrib. Syst.*, IEEE Press, Piscataway, NJ, USA, v. 18, n. 4, p. 460–473, 2007. ISSN 1045-9219.

ZHU, B.; JAJODIA, S.; KANKANHALLI, M. S. Building trust in peer to peer systems: a review. *Int. J. Secur. Netw.*, Inderscience Publishers, Inderscience Publishers, Geneva, SWITZERLAND, v. 1, n. 1/2, p. 103–112, 2006. ISSN 1747-8405.