# *PeerRepSim:* um Simulador de Protocolos de Gerenciamento de Reputação para Redes P2P

**Anderson do Nascimento**[1], **Elisângela Carneiro**[2],
**Antônio Coutinho**[1], **Fabíola Greve**[2]

[1]Graduação em Engenharia de Computação
Departamento de Tecnologia – Universidade Estadual de Feira de Santana (UEFS)
Avenida Transnordestina S/N – CEP: 44.036-900 – Feira de Santana – BA – Brasil

[2]Programa de Pós-graduação em Mecatrônica
Departamento de Ciência da Computação – Universidade Federal da Bahia (UFBA)
CEP 40.170.110-10 – Salvador – BA – Brasil

***Abstract.*** *In spite of the popularity of Peer-to-Peer (P2P) systems in the Internet, the research and development of new technologies in the field are limited by the absence of generic P2P simulation and project tools, specially the ones which support the simulation of security P2P protocols and threat models. This paper describes the project of the PeerRepSim, a P2P network simulator which aggregates the simulation of reputation management protocols as well as the simulation of malicious node behaviors. The paper discusses PeerRepSim's main simulation requisites and architectural components, as well as some performance results regarding protocol simulations.*

***Resumo.*** *O modelo de sistemas entre pares (P2P) tornou-se altamente popular na Internet, particularmente em sistemas de compartilhamento e distribuição de conteúdo. No entanto, a pesquisa e o desenvolvimento de novas tecnologias na área são limitadas pela ausência de ferramentas genéricas de projeto e simulação P2P, sobretudo as que oferecem suporte a simulação de protocolos de segurança e modelos de ataque a estes sistemas. Este artigo discute o projeto do PeerRepSim, um simulador de redes P2P que agrega a simulação de protocolos de gerenciamento de reputação, bem como a simulação a ataques e comportamentos de agentes mal intencionados. O artigo aborda os principais requisitos de simulação e componentes arquiteturais do simulador, bem como alguns resultados de simulação de protocolos existentes na literatura.*

## 1. Introdução

Sistemas entre pares (P2P) são sistemas distribuídos de larga-escala, cujo funcionamento envolve numerosas e complexas interações entre usuários e máquinas. As redes P2P são formadas a partir da interconexão entre vários nós independentes (*peers*), com o intuito de compartilhar recursos computacionais (e.g. arquivos, ciclos de processamento, largura de banda) sem o auxílio de entidades intermediárias. São caracterizadas principalmente pela descentralização, autonomia e pelo crescimento em escala. As redes P2P são capazes de se auto-organizar e manter níveis aceitáveis de desempenho e conectividade diante de padrões de comportamento dinâmico de seus participantes. O perfil descentralizado das redes P2P as tornam sistemas altamente robustos e disponíveis, com grande tolerância a falhas de máquinas e rede [Roussopoulos et al. 2004].

As características das redes P2P as tornam modelos bastante atraentes para uma série de aplicações na *Internet*. Arquiteturas P2P são utilizadas em uma grande variedade de aplicações distribuídas, sobretudo em sistemas de compartilhamento e distribuição de conteúdo, sistemas de computação e banco de dados distribuídos, sistemas de transmissão de fluxos de dados multimídia (*streaming*) e programas de trocas de mensagens instantâneas.

Diferente da arquitetura tradicional da *Internet*, baseada no modelo cliente/servidor, em um ambiente P2P cada participante atua simultaneamente como cliente e provedor de recursos. São autônomos e determinam seu próprio comportamento, sem a intervenção de uma entidade centralizada. Cada participante pode entrar ou sair da rede de acordo com seu interesse. A descentralização elimina a presença de pontos centrais de falhas e aumenta a disponibilidade das redes P2P. Apesar de diversas vantagens, a descentralização e a autonomia tornam o sistema vulnerável a ações de nós mal intencionados. A autonomia permite aos usuários desobedecerem políticas e normas de conduta estabelecidas no sistema, como, por exemplo, não compartilhar recursos sobre sua responsabilidade, ou recusar-se a avaliar interações com outros usuários. A descentralização, por sua vez, enfrenta problemas especialmente quanto à autenticidade dos participantes e quanto à confiabilidade das informações transmitidas pela rede.

A compreensão da ação de agentes maliciosos e seu impacto sobre a rede é de fundamental importância para o desenvolvimento de protocolos e mecanismos de segurança para sistemas P2P. Observar o comportamento das arquiteturas sob diferentes formas de ataque e cenários críticos auxilia na identificação de vulnerabilidades, erros de projeto e gargalos operacionais. No entanto, a realização de testes e análises em sistemas reais de larga escala é normalmente impraticável. A dimensão e a complexidade inerentes às redes P2P dificultam o controle e a coleta de resultados de experimentos. Além disso, sistemas P2P normalmente demandam uma vasta quantidade de máquinas e uma grande estrutura de rede, o que torna os custos da realização de testes e alterações elevados.

Neste contexto, o uso de simulações é uma alternativa prática e de menor custo para contornar a dificuldade de estudar sistemas P2P reais em funcionamento. O emprego de simuladores P2P permite a observação das redes de forma controlada e eficiente, com a coleta de diversas estatísticas sobre seu funcionamento. Dentre outros benefícios do uso de simulação estão a capacidade de testar hipóteses acerca do comportamento do sistema, promover e avaliar modificações estruturais e identificar gargalos de operação, tudo isso sem alterar o funcionamento do sistema real [Banks et al. 2000].

Entretanto, existe uma carência significativa de simuladores P2P para o meio acadêmico, sobretudo os que suportam a simulação de protocolos de segurança e modelos de ataque. Alguns estudos como em [Naicken et al. 2007] e [Baker and Rahim 2007] comprovaram que a maioria dos simuladores disponíveis no mercado não atende a uma série de requisitos básicos para uso em pesquisa e o desenvolvimento de novas arquiteturas e protocolos P2P. A falta de simuladores suficientemente genéricos e de fácil adaptação leva grande parte dos pesquisadores a implementarem seus próprios ambientes de simulação de acordo com suas necessidades específicas. Em conseqüência do uso de simuladores de propósito específico, há uma dificuldade na verificação e reprodução dos resultados obtidos em diferentes pesquisas. Algumas exceções são os simuladores PeerSim[Jelasity et al. 2010] e P2PSim[Gil et al. 2010], que são ferramentas genéricas,

amplamente adotadas pela comunidade para testes comparativos. O PeerSim tem a característica de ser altamente escalável e adaptável ao alto dinamismo proporcionado por entradas e saídas frequentes de nós. Já o P2PSim maximiza a concorrência através da adoção de múltiplos *threads*. Entretanto, nenhum destes simuladores tem facilidades específicas para os aspectos de segurança e reputação dos protocolos.

Visando contribuir para a pesquisa de novas soluções em segurança de sistemas P2P, bem como para a análise de protocolos e arquiteturas existentes, este trabalho discute o projeto do *PeerRepSim*: um simulador de redes P2P que possibilita a simulação de protocolos de gerenciamento de reputação e de modelos de comportamentos maliciosos de nós. Os mecanismos de reputação agregam opiniões sobre o comportamento passado das entidades com o objetivo de estimar o seu comportamento futuro. Essa estimativa é baseada em valores que exprimem o grau de confiança nas entidades [Liu and Qiu 2007].

O *PeerRepSim* oferece suporte a simulação de redes sobrepostas estruturadas sob o modelo *Chord*, e não estruturadas sob o modelo *Gnutella* original [Gnutella 2010]. As redes sobrepostas, por sua vez, servem de base para a simulação dos cinco protocolos de gerenciamento de reputação integrados ao *PeerRepSim*, são eles: o P2PRep [Cornelli et al. 2002], o XRep [Damiani et al. 2002], a Abordagem Bayesiana Combinada [Buchegger and Le Boudec 2004], o EigenTrust [Kamvar et al. 2003] e o STORM [Ravoaja and Anceaume 2007]. Além disso, o *PeerRepSim* implementa modelos de ameaças baseados em comportamentos maliciosos dos usuários que fazem parte da rede, com o intuito de possibilitar a avaliação dos protocolos simulados sob diversas formas de ataque e ambientes críticos.

O restante deste artigo está organizado da seguinte maneira. A Seção 2 aborda os principais requisitos de *software* estabelecidos para o *PeerRepSim*. A Seção 3 apresenta o modelo arquitetural do *PeerRepSim* e descreve a estrutura e o comportamento de seus principais componentes. A Seção 4 apresenta as características funcionais da ferramenta, uma análise da performance do sistema e alguns resultados de simulação dos protocolos implementados. Por fim, a Seção 5 apresenta considerações sobre os resultados obtidos e indicações de trabalhos futuros com o simulador.

## 2. Requisitos de Simulação

O funcionamento das redes P2P apresenta um considerável número de variáveis interdependentes e normalmente demanda uma grande infraestrutura de máquinas e rede. Diante disso, o uso de simuladores é bastante útil para observar o comportamento destes sistemas, tanto em sua fase de projeto quanto em sua manutenção.

Em [Naicken et al. 2006] é estabelecido um conjunto de critérios de avaliação para os simuladores P2P, com base nos principais requisitos encontrados em diversas pesquisas no setor. Os critérios dizem respeito aos seguintes aspectos: a arquitetura do sistema, usabilidade, estratégia de simulação da rede subjacente, estatísticas geradas e capacidades de simulação. Estes critérios, por sua vez, foram empregados como diretrizes para o estabelecimento dos requisitos do *PeerRepSim*.

### 2.1. Requisitos Gerais de Simulação

De maneira geral, um simulador P2P deve refletir as complexas interações entre usuários, máquinas, protocolos e redes num ambiente controlado, gerando um conjunto significa-

tivo de estatísticas para a observação e análise do modelo implementado. A arquitetura deve ser flexível para permitir a fácil adaptação do sistema à diferentes necessidades de pesquisa, ou seja, o sistema deve possibilitar ao usuário a fácil integração e remoção de componentes específicos como protocolos e modelos de comportamento. Além disso, o simulador deve oferecer ao usuário uma ampla capacidade de configuração dos diversos parâmetros de rede e de protocolos, possibilitando a realização de experimentos num grande número de cenários.

Quanto às características internas do simulador, um aspecto importante de projeto diz respeito ao modo de simulação da rede subjacente. O sistema pode simular a transmissão de cada pacote individual, introduzir atrasos e falhas em nível de camada física, ou simplesmente abstrair estes detalhes e focar-se exclusivamente na manutenção da rede sobreposta (*overlay network*). Ainda no contexto de rede, devem ser considerados os mecanismos de endereçamento e geração de identificadores implementados, bem como a capacidade de simulação de transações simultâneas entre os nós.

A disponibilização de uma interface gráfica de usuário (*Graphical User Interface*, GUI) amigável, de fácil manuseio e que ofereça ampla possibilidade de configuração é uma característica bastante desejável em simuladores P2P. De certa forma, uma GUI facilita a realização de experimentos individuais e estimula o uso do simulador, embora possa diminuir a eficiência da simulação em lote de um grande número de cenários. O ideal é que o usuário possa interagir com o sistema, fornecendo parâmetros de configuração e coletando resultados tanto por meio de uma GUI, em caso de simulações individuais, quanto por linha de comando e arquivos de configuração, em caso de simulações em lote.

É fundamental que os resultados gerados pelo simulador possam ser exportados para outras ferramentas de análise e geração de estatísticas. A utilização de um formato de dados intermediário e de simples manuseio é uma estratégia atraente para este requisito. Uma alternativa viável é o uso de arquivos no formato **.csv** (*Comma Separated Values*) para armazenamento das séries temporais geradas pelo simulador. Estes arquivos podem ser utilizados como fontes de dados em uma grande variedade de aplicações estatísticas por ser de simples manuseio.

### 2.2. Modelos de Ameaças

Como forma de avaliar o comportamento dos protocolos diante de cenários agressivos e ataques mal intencionados, um simulador P2P com foco em segurança deve implementar modelos de ameaças e falhas de dispositivos.

O *PeerRepSim* implementa cinco modelos de ameaças baseadas no modelo proposto em [Schlosser et al. 2005]. Os modelos de comportamento são descritos a seguir.

**Honestos.** Comportam-se de maneira correta no sistema. Colaboram com recursos sobre sua responsabilidade e avaliam seus pares de forma adequada, ou seja, emite avaliações positivas quando recebem recursos legítimos e avaliações negativas caso contrário.

**Maliciosos.** Têm como objetivo minar o sistema com avaliações incorretas. Possui comportamento arbitrário, emite avaliações corretas e incorretas de acordo com uma função interna de probabilidade.

**Conspiradores.** Tentam acumular uma alta reputação formando um grupo (conluio) onde todos se reconhecem e se avaliam positivamente. Sempre que um nó conspirador encontra outro nó conspirador ele interage corretamente (e.g., repassa recursos, encaminha mensagens, etc.) e dá sempre uma avaliação positiva. Ao interagir com outros tipos de nós, ele fornece recursos e avaliações neutras.

**Egoístas.** Os nós egoístas buscam usufruir o máximo dos serviços oferecidos na rede, mas contribuem o mínimo possível, ou até mesmo não contribuem. Assim, não encaminham mensagens, não fornecem arquivos sob o seu domínio e não avaliam transações. Um exemplo de tais nós são os chamados *free-riders* em redes de compartilhamento de arquivos como o *Kazaa* e *Gnutella*. Eles normalmente não compartilham arquivos com o objetivo de reduzir o consumo de largura de banda e processamento em suas máquinas.

**Perturbadores.** Agem como nós honestos até alcançarem altos valores de reputação. Em seguida, passam a assumir comportamento semelhante aos nós maliciosos. Ele permanece neste estado até que sua reputação atinja um valor mínimo, então sai da rede e retorna com um novo identificador. Ao retornar à rede a sua reputação possui um valor inicial e seus repositórios de experiências são vazios. Esse modelo implementa características dos ataques do tipo *whitewashers* [Feldman et al. 2004].

Para simular o comportamento dinâmico dos usuários no sistema real, o *PeerRepSim* provê um mecanismo de falhas baseado na entrada e saída inesperada de nós (*churn*). A entrada e saída dos nós é realizada de forma aleatória no sistema, obedecendo um percentual pré-estabelecido nos parâmetros iniciais de configuração do simulação. A saída e entrada é empregada de forma que se assemelhe ao comportamento real dos usuários, que saem da rede no momento que acham conveniente ou por eventuais falhas de *hardware* e rede.

O simulador proposto é capaz de receber novas implementações de comportamento, ambientes P2P e protocolos de gerenciamento de reputação de forma fácil, por se tratar de uma API simples e portável o suficiente para que outros usuários possam testar seus próprios modelos no *PeerRepSim*.

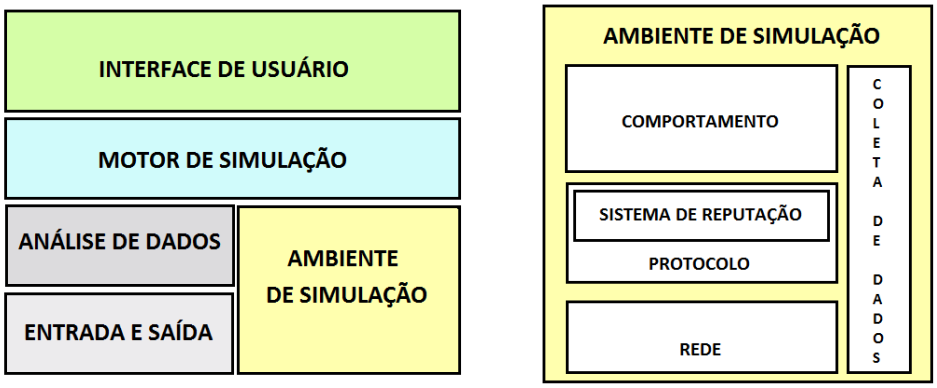
## 3. Modelo Arquitetural

A arquitetura de um sistema é sua organização fundamental representada por componentes e princípios que guiam o seu desenvolvimento e evolução. A arquitetura compreende também os relacionamentos dos componentes entre si e deles com o ambiente. Um modelo arquitetural, por outro lado, é o conjunto de artefatos que documentam a arquitetura de um sistema [Hilliard 2000].

O funcionamento geral do sistema é baseado no modelo de ciclos de busca (*query cycles*) descrito em [Schlosser et al. 2003]. Um ciclo de busca engloba as operações realizadas entre a requisição e a transferência de um recurso. Um ciclo se incia após um nó requisitar um recurso a seus pares e termina com o recebimento ou não do mesmo. No *PeerRepSim*, a transação é caracterizada pelo recebimento do recurso solicitado, o que corresponde a um ciclo. A avaliação desta transação pode ser: boa (recebimento de um recurso legítimo), ruim (recebimento de um recurso corrompido) ou neutro (se o recurso recebido não é o esperado, mas não caracteriza um recurso corrompido).

O *PeerRepSim* possui uma arquitetura semelhante ao modelo 3-LS descrito em [San Ting 2003]. Conceitualmente, o sistema também é dividido em camadas básicas que se comunicam entre si, cada uma fornece serviços para as camadas adjacentes. A Figura 1 apresenta um esboço da arquitetura geral do simulador com os seus componentes básicos. No entanto, diferentemente do modelo 3-LS, os mecanismos de troca de mensagens entre os nós, encapsulados na camada Rede da arquitetura, baseiam-se no modelo de emissores e ouvintes de eventos e não no esquema de "caixas de correio"do modelo original. Diferentemente do modelo de "caixas-de-correio", a arquitetura baseada em eventos amplia a capacidade de simulação de transações concorrentes no sistema.

O modelo reúne cinco grandes componentes: *interface de usuário*, *motor de simulação*, *ambiente de simulação*, *análise de dados* e *entrada e saída*. Cada um destes componentes encapsula atributos e responsabilidades específicas, sendo que suas interfaces oferecem serviços aos componentes adjacentes. Uma descrição de cada componente é feita a seguir.




(a) Modelo Conceitual (b) Ambiente de Simulação

**Figura 1. Arquitetura geral do PeerRepSim: Modelo Conceitual e Ambiente de Simulação**

**Interface de Usuário.** É responsável por reunir as entidades responsáveis pela coleta dos parâmetros inicias da simulação. A configuração dos parâmetros iniciais podem ser realizadas através da interface gráfica de usuário e por arquivos de propriedades. A configuração por interface gráfica oferece ao usuário a possibilidade de supervisão do experimento pela própria interface gráfica, baseada em *Swing* [Sun 2010]. A interface disponibiliza formulários de configuração da simulação, gráficos dinâmicos com estatísticas sobre o experimento e controles de início e de interrupção do experimento. A configuração por arquivos de propriedade é destinada aos usuários que desejam realizar um grande número de simulações em lote.

**Motor de Simulação.** Uma vez recebidos os parâmetros de configuração do usuário, o Motor de Simulação é responsável por criar as entidades necessárias para a simulação (e.g., rede, nós, protocolos, repositórios, etc.) e iniciar a execução. O conjunto destas entidades compõe um Ambiente de Simulação. O Motor pode executar um ou mais Ambientes de Simulação ao mesmo tempo, porém cada um gera

seus próprios resultados. O Motor de Simulação também é responsável pela coordenação dos passos da execução (*ticks*), realiza periodicamente limpezas de memória, esvaziamento de *buffers*, sincronização entre as transações e chamadas para a organização de dados do experimento. Durante um passo de simulação, o Motor escolhe aleatoriamente um número *n* de nós que realizarão ciclos de busca, onde *n* é um parâmetro de simulação fornecido pelo usuário. Após a realização dos ciclos de busca, o componente aciona o módulo de Análise de Dados para realizar a organização dos dados gerados até então. Os nós escolhidos para realização de ciclos de busca em um passo de simulação são acionados concorrentemente pelo Motor de Simulação.

**Ambiente de Simulação.** Este componente reúne as entidades básicas do simulador, conforme apresentado na Figura 1(a). É composto por três camadas fundamentais (semelhantes às do modelo 3-LS) e um subsistema de coleta de dados. A *camada de Rede* é responsável pelo encaminhamento das mensagens entre os pares. A *camada Protocolo* inclui as implementações de protocolos e mensagens, além de estruturas de rede e reputação. Sua entidade principal é o *Peer*, o qual representa um nó da rede. Por fim, a *camada Usuário* contém as implementações dos modelos de comportamentos dos usuários, necessárias nas tomadas de decisões dos protocolos. O subsistema de coleta de dados é distribuído pelas três camadas básicas, sendo responsável pela coleta de estatísticas da simulação, como por exemplo, a contagem de mensagens na rede e o cálculo de reputações globais. O subsistema de coleta de dados comunica-se com módulo de Análise de Dados. A reputação global ($M_p$) de um *Peer p* é a média aritmética de todas as avaliações que ele recebeu na rede.

**Análise de Dados.** É responsável pela organização e classificação dos dados coletados, bem como pela elaboração de gráficos e conjuntos de saída. Os gráficos gerados por este módulo são armazenados em arquivo através da interface oferecida pelo componente de entrada e saída. A classificação dos dados depende da natureza das amostras coletadas. Amostras de mensagens podem ser agrupadas por tipo ou por comportamento do remetente/destinatário. Médias globais de reputação são agrupadas por grupos de comportamento. Uma vez organizados, os dados podem ser exportados em gráficos e/ou arquivos separados por vírgula, para análises posteriores em outras ferramentas. A adição de um novo tipo de gráfico para a análise envolve a implementação de interfaces definidas neste componente, assim como a implementação de rotinas no subsistema de coleta de dados. O Motor de Simulação é programado para exportar cada um dos gráficos definidos no módulo de Análise de Dados.

**Entrada e Saída.** Este componente contém classes que auxiliam a leitura e escrita de arquivos separados por vírgula para entrada e saída de dados. Além de fornecer interfaces para a leitura de arquivos de propriedades e conjuntos de comportamentos de nós.

O sistema de troca de mensagens do *PeerRepSim* é inspirado no modelo de ouvintes de eventos (*event listeners*) do *Swing* [Sun 2010]. A Figura 2 ilustra o funcionamento do modelo de eventos. A idéia central deste modelo é permitir o tratamento ins-

tantâneo de eventos de forma concorrente. Uma fonte de eventos é responsável por encaminhar os eventos a ouvintes específicos, filtrando o fluxo e evitando que os demais ouvintes sejam acionados sem necessidade. Cada evento possui um identificador do ouvinte, ou grupo de ouvintes, ao qual ele é destinado. Internamente, as fontes de eventos mantém uma lista de ouvintes cadastrados, sendo que cada um deles pode ser associado a diferentes fontes.

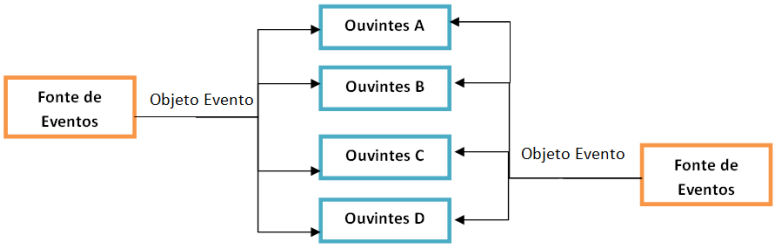



**Figura 2. Modelo de ouvintes de eventos**

A rede é considerada uma fonte de eventos, os nós representam os ouvintes e as mensagens transmitidas são os eventos em si. Cada evento é encaminhado a um nó específico, identificado por uma chave única de destino na mensagem. Somente o nó de destino é acionado pela rede. Ao receber uma mensagem, o destinatário a encaminha para tratamento em alguma instância de protocolo previamente configurada.

Um último aspecto a ser considerado na arquitetura do simulador proposto é o tratamento do grande volume de dados gerados durante uma simulação. Uma característica importante para arquitetura do sistema é a coleta e organização destes resultados de forma simples e eficiente. Os dados são agrupados e classificados em conjuntos de séries temporais com base nos passos da simulação (*ticks*). As estatísticas geradas são então apresentadas de forma clara para os usuários, preferencialmente em forma de gráficos e/ou histogramas. Outro requisito importante é a possibilidade de reutilização dos dados gerados em outras ferramentas de geração de análise. Para isso, as saídas são também exportadas em arquivos *.csv*. As estatísticas oferecidas pelo simulador são:

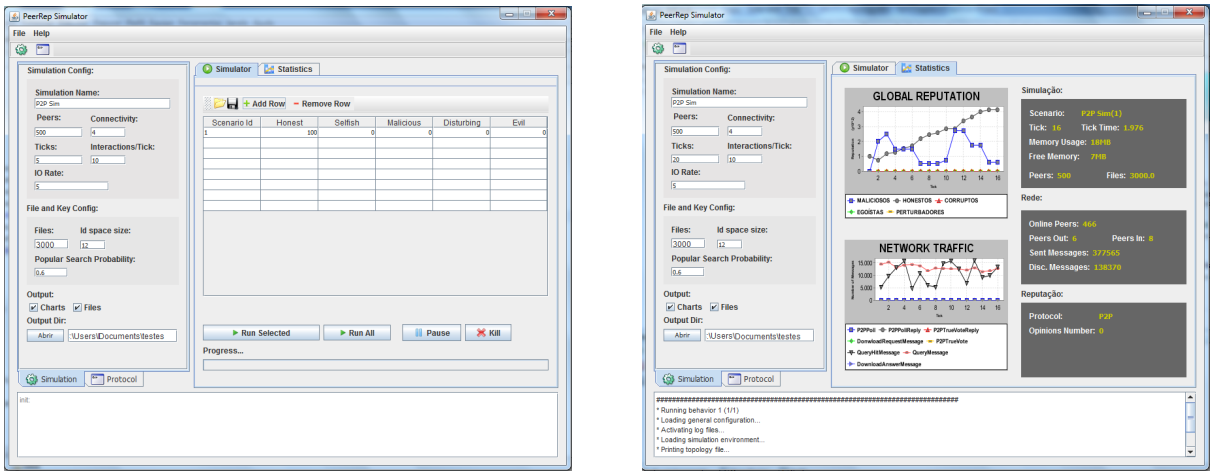
1. *Média global de reputação:* é a média aritmética da reputação alcançada por tipo de nós, calculada através da equação **??**. Pode-se verificar se os nós maliciosos ganham ou perdem reputação ao longo dos ticks.
2. *Mensagens total trafegadas na rede:* quantidade de mensagens trafegadas na rede em cada *tick*, agrupadas por tipo de mensagem. A partir desta informação é possível verificar qual protocolo emite mais mensagens para realizar as ações de cada protocolo.
3. *Mensagens transmitidas na rede por tipo de nó:* é a média aritmética das mensagens emitidas na rede agrupadas por tipo de comportamento do nó emissor. Assim, pode-se avaliar qual tipo de nó é mais ativo em cenários específicos.
4. *Total de mensagens descartadas por tick:* quantidade de mensagens perdidas na rede por tipo.
5. *Percentual de mensagens descartadas em relação às demais mensagens:* a partir desta informação pode-se perceber se houve alteração no comportamento dos nós e a verificação de perda de mensagens de cada protocolo em função do cenário.

6. *Distribuição de arquivos por quantidade de Peers:* mostra a quantidade de arquivos armazenadas em cada nó, distribuição de arquivos segue a função *zipf*.
7. *Contagem das ofertas e pedidos de recursos por tipo de Peer:* quantidade de pedidos realizados e recursos ofertados por tipos de nós. Ao longo dos *ticks* pode-se avaliar se os nós escolhem os melhores provedores de recursos (nós honestos).
8. *Tempo de execução de cada tick:* o tempo gasto para executar $n$ ciclos de busca. Pode-se avaliar a complexidade dos protocolos.
9. *Número de transações bem e mal sucedidas:* somatório das transações com suas respectivas avaliações. Pretende-se avaliar os cenários e a sua robustez diante dos ataques maliciosos e a efetividade de cada protocolo de reputação.
10. *Número de mensagens exclusivas de sistemas de reputação na rede:* pode-se avaliar a complexidade de mensagens de cada sistema de gerenciamento de reputação. As mensagens são agrupadas por tipo.

De posse das informações coletadas é possível inferir sobre a eficiência dos sistemas de gerenciamento de reputação, avaliar a sua robustez, verificar a complexidade de mensagem e o armazenamento requerido. Além das informações sobre os sistemas de reputação pode-se avaliar o comportamento da rede e como são realizadas as ações básicas de cada protocolo, como busca e obtenção de recursos.

## 4. Implementação e Resultados

O principal resultado deste trabalho é a construção do *PeerRepSim*, enquanto *software* de simulação P2P, e suas possíveis aplicações na pesquisa e desenvolvimento de protocolos de gerenciamento de reputação. A Figura 3(a) apresenta a interface do simulador e mostra a interface para configuração dos parâmetros iniciais e dos cenários a serem simulados. Na Figura 3(b) é apresentada a coleta de estatísticas em tempo de simulação.




(a) *PeerRepSim:* Configuração de Simulação (b) *PeerRepSim:* Estatísticas em Tempo de Simulação

**Figura 3. Arquitetura geral do PeerRepSim**

Foram realizados dois experimentos com o simulador: o primeiro, com intuito de observar o comportamento dos protocolos implementados diante de diferentes cenários de ameaças; e o segundo, com o objetivo de verificar a complexidade das mensagens geradas pelos protocolos P2P: *Gnutella* original e o *Chord*.

### 4.1. Experimento I

O objetivo deste experimento foi verificar o impacto da atuação dos diversos tipos de nós maliciosos sobre a reputação dos nós honestos. A Tabela 4 mostra os parâmetros inciais da simulação, como: a quantidade de nós na rede, a quantidade de arquivos distribuídos nos nós, a quantidade de *ticks* de simulação, quantidade de transações realizadas por *tick* e probabilidade da busca popular. Na Tabela 2 são apresentados os cenários das simulações. Os cenários são estabelecidos a partir da quantidade de nós presentes na rede em cada simulação, alterando as quantidades de nós pode-se perceber cenários críticos onde o sistema não suporta determinada quantidade de nós mal intencionados.

**Tabela 1. Parâmetros de Simulação do Experimento I**

| Parâmetros | Valores |
|---|---|
| Número de nós na rede | 500 |
| Número de *ticks* | 50 |
| Número de transações por *tick* | 20 |
| Taxa de entrada/saída dos nós | 5% |
| Número de arquivos dispersos na rede | $2^{13}$ |
| Probabilidade da busca popular | 60% |

**Tabela 2. Cenários do experimento I**

| | *Nós* | | | | |
|---|---|---|---|---|---|
| Cenários | honestos | egoístas | maliciosos | corruptos | perturbadores |
| Cenário 01 | 100% | – | – | – | – |
| Cenário 02 | 50% | 50% | – | – | – |
| Cenário 03 | 50% | – | 50% | – | – |
| Cenário 04 | 50% | – | – | 50% | – |
| Cenário 05 | 50% | – | – | – | 50% |

O principal instrumento para esta análise é o gráfico da média de reputação dos nós gerados pelo *PeerRepSim*. No entanto, esta informação não é suficiente para afirmar que este comportamento é o mais prejudicial ao sistema, do ponto de vista da qualidade das transações. Uma vez que a rede simulada modela um ambiente de troca de arquivos, é preciso observar a ação de cada ataque sobre a qualidade das transações efetivadas.

A qualidade de cada transação realizada na rede é medida através da avaliação feita pelos pares. No entanto, as avaliações são uma visão parcial da rede e depende do comportamento do nó avaliador, que nem sempre é confiável. A robustez dos protocolos de reputação é avaliada em função da reputação dos nós honestos, pois estes nós sempre fornecem recursos bons e fazem boas avaliações. Se a reputação dos nós honestos é baixa e a dos nós mal intencionados é alta, isto significa que os nós maliciosos conseguem subverter o sistema. Como a reputação dos nós honestos é baixa, eles não serão escolhidos como provedores de recursos, assim, a qualidade dos recursos fornecidos na rede não será boa.

Através da análise dos resultados obtidos é possível determinar o tipo de ataque mais danoso ao funcionamento de cada protocolo de reputação. Os danos estão relacionados à concepção de cada protocolo e ao modo como coletam, agregam e distribuem informações na rede. A Tabela 3 apresenta os resultados obtidos neste experimento. Foram

analisados o tempo médio de execução dos cenários apresentados na Tabela 2 e qual modelo de ameaça interfere mais na reputação dos nós honestos neste cenário.

O protocolo STORM apresenta um maior tempo de execução, isto se dá por causa da manutenção da arquitetura estruturada da rede, além da arquitetura do próprio sistema de gerenciamento de reputação que mantém uma estrutura de *clusters* dos provedores de recursos organizadas em anel, semelhante ao *Chord*. O protocolo Abordagem Bayesiana Combinada tem o menor tempo de execução. Este protocolo tem uma abordagem proativa, o que reduz bastante o tempo necessário para localizar informações sobre o comportamento dos nós na rede.

Os protocolos STORM e EigengTrust são os mais sensíveis aos ataques de nós egoístas. O sistema de gerenciamento de ambos os protocolos criam uma arquitetura estruturada para gerenciar as informações. Caso não haja colaboração, a manutenção da rede é comprometida. Além disso, os autores do protocolo EigenTrust admitem a existência de um percentual de nós honestos presentes na rede, e que seriam os nós dispostos a construir a rede. Os protocolos XREP e P2PRep apresentam maior vulnerabilidade aos ataques dos nós maliciosos.

Os nós honestos conseguem uma maior reputação no protocolo EigenTrust e no Abordagem Bayesiana. Apesar da vulnerabilidade aos ataques dos nós egoístas e maliciosos, o Abordagem Bayesiana apresenta características que impedem que os nós maliciosos alcancem elevados valores de reputação.

**Tabela 3. Parâmetros de Simulação do Experimento II**

| Protocolo | Ataque | Tempo de Simulação (s) |
|---|---|---|
| A. Bayesiana | Maliciosos/Egoístas | 221,77 |
| EigenTrust | Egoístas | 305,20 |
| P2PRep | Maliciosos | 228,32 |
| STORM | Egoístas | 357,57 |
| XREP | Maliciosos | 240,01 |

## 4.2. Experimento II

Além da reputação média dos nós e o tempo de execução dos protocolos, é possível verificar a quantidade de mensagens trafegadas na rede pelos protocolos P2P: *Gnutella* na sua versão original e o *Chord*. Na Tabela 4 são apresentados os parâmetros da simulação para este experimento.

**Tabela 4. Parâmetros de Simulação do Experimento II**

| Parâmetros | Valores |
|---|---|
| Número de nós na rede | 250 |
| Número de *ticks* | 100 |
| Número de transações por *tick* | 50 |
| Taxa de entrada/saída dos nós | 5% |
| Número de arquivos dispersos na rede | $2^{13}$ |

Na Figura 4 é apresentada a quantidade de mensagens disseminadas na rede *Gnutella* original para realizar ações de busca (`Query`), respostas às buscas (`QueryHits`), requisição do recurso (`DownloadRequest`), envio do recurso

(`DownloadAnswer`) e manutenção da rede. A quantidade de mensagens apresentadas no gráfico é um valor médio calculado em cada *tick*.

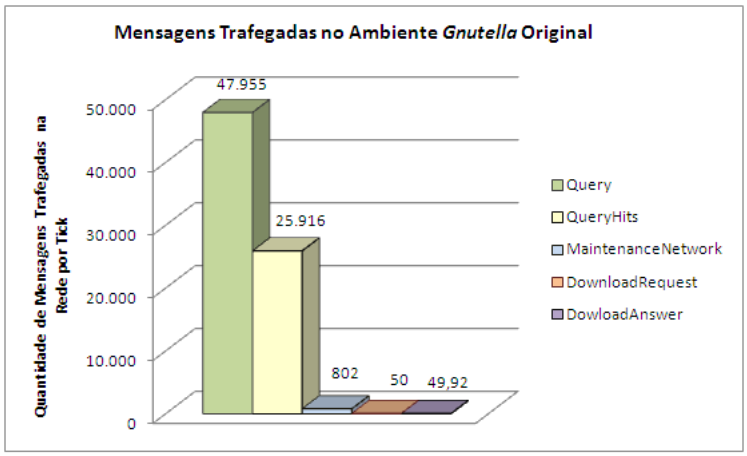



**Figura 4. Mensagens trafegadas no ambiente *Gnutella* original.**

Na Figura 5 pode-se observar a quantidade de mensagens trafegadas no sistema sob a arquitetura estruturada *Chord*. Ao comparar as quantidades das mensagens trafegadas nos ambientes *Gnutella* original e *Chord* pode-se perceber que a quantidade de mensagens do tipo `Query` e `QueryHit` no ambiente *Gnutella* é bastante superior a quantidade das mesmas mensagens no ambiente *Chord*. No entanto, a quantidade de mensagens para manutenção da rede (*MaintenanceNetwork*) no *Chord* é maior do que o mesmo tipo de mensagem na rede *Gnutella* original. Isto se dá por causa do custo da manutenção da rede estruturada.

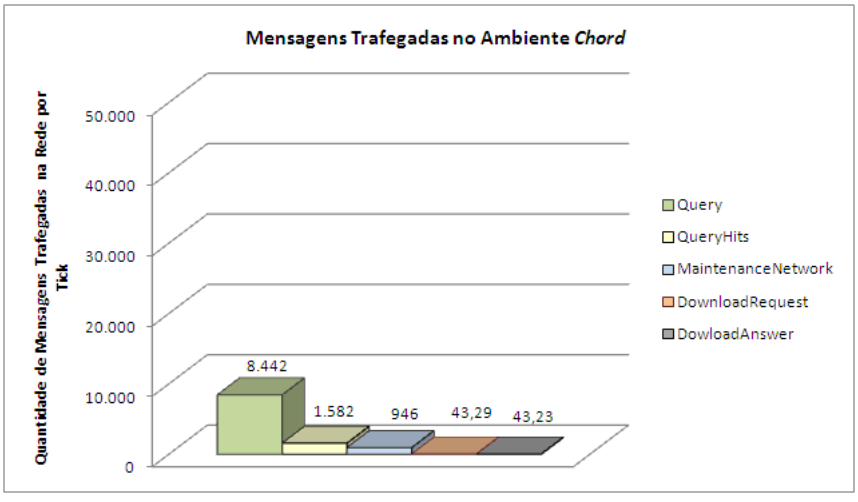



**Figura 5. Mensagens trafegadas no ambiente *Chord*.**

O ambiente *Gnutella* original dissemina uma maior quantidade de mensagens na rede, principalmente para requisitar recursos. Na arquitura estrutura como o *Chord* a busca é realizada de forma mais eficiente e menos custosa.

## 5. Conclusões e Trabalhos Futuros

Este artigo apresentou o *PeerRepSim*, um simulador de redes P2P que agrega a simulação tanto de protocolos de gerenciamento de reputação, quanto de ataques e comportamentos de agentes mal intencionados. O artigo abordou os principais requisitos para a simulação, aspectos do modelo arquitetural do simulador, bem como alguns resultados de simulação de protocolos existentes na literatura. As estatísticas geradas possibilitam uma análise criteriosa dos protocolos sob vários aspectos, desde o tráfego na rede até as médias de reputação acumuladas por cada grupo de nós. O *PeerRepSim* é relativamente simples de utilizar e gera um conjunto variado de estatísticas de forma clara para análise sobre o comportamento da rede.

O *PeerRepSim* atende aos critérios de avaliação de simuladores estabelecidos por [Naicken et al. 2006] e apresenta as seguintes características: (i) uma arquitetura baseada em ciclos de eventos, que implementa uma estratégia de simulação da rede adjacente com troca efetiva de mensagens entre os nós; (ii) uma interface gráfica de usuário, que facilita o seu manuseio, bem como um conjunto diversificado de estatísticas de saída; e, por fim, (iii) uma implementação de transações concorrentes. Esta última característica o distingue de simuladores populares como o *PeerSim*[Jelasity et al. 2010], que implementa uma fila sequencial de eventos de simulação. Tal fator amplia o número total de transações simuladas no sistema.

Como forma de aumentar a capacidade de simulação e reduzir o tempo de execução de experimentos no PeerRepSim, está sendo estudada a sua adaptação para execução em grade computacional P2P. O objetivo é estabelecer uma estrutura que possibilite a paralelização não somente das simulações sobre grandes conjuntos de dados, mas também que adapte o Motor de Simulação para execução em paralelo de suas rotinas. Com isto, espera-se aumentar consideravelmente o desempenho do simulador. A integração de modelos de mobilidade e consumo de energia ao sistema também está sendo estudada. O objetivo é aumentar as possibilidades de simulação do PeerRepSim, oferecendo aos usuários um modelo de rede mais realístico.

## Referências

Baker, M. and Rahim, L. (2007). Peer-to-peer simulators. *Technical Reports*.

Banks, J., Carson, J. S., Nelson, B. L., and Nicol, D. M. (2000). *Discrete-Event System Simulation (3rd Edition)*. Prentice Hall, 3 edition.

Buchegger, S. and Le Boudec, J. Y. (2004). A robust reputation system for p2p and mobile ad-hoc networks. In *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*.

Cornelli, F., Damiani, E., Capitani di Vimercati, S. D., Paraboschi, S., and Samarati, P. (2002). Choosing reputable servants in a p2p network. In *In Proceedings of the 11th International World Wide Web Conference*. ACM.

Damiani, E., Vimercati, D. C. D., Paraboschi, S., Samarati, P., and Violante, F. (2002). A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *In Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 207–216. ACM Press.

Feldman, M., Papadimitriou, C., Chuang, J., and Stoica, I. (2004). Free-riding and whitewashing in peer-to-peer systems. In *PINS '04: Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pages 228–236, New York, NY, USA. ACM.

Gil, T., Kaashoek, F., Li, J., Morris, R., and Stribling, J. (2010). P2psim: A simulator for p2p protocols. Disponível em: http://pdos.csail.mit.edu/p2psim/. Acesso em: 30/08/2010.

Gnutella (2010). The gnutella protocol development website. Disponível em: http://rfc-gnutella.sourceforge.net/.

Hilliard, R. (2000). Ieee recommended practice for architectural description of software-intensive systems. Technical report.

Jelasity, M., Montresor, A., Jesi, G. P., and Voulgaris, S. (2010). The Peersim simulator. Disponível em: http://peersim.sourceforge.net. Acesso em: 30/07/2010.

Kamvar, S. D., Schlosser, M. T., and Molina, H. G. (2003). The eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA. ACM Press.

Liu, H. and Qiu, Y. (2007). A reputation model base on transactions in peer-to-peer networks. In *SKG '07: Proceedings of the Third International Conference on Semantics, Knowledge and Grid*, pages 398–401, Washington, DC, USA. IEEE Computer Society.

Naicken, S., Basu, A., Livingston, B., and Rodhetbhai, S. (2006). A survey of peer-to-peer network simulators. *Proceedings of the 7th Annual Postgraduate Symposium (PGNet '06)*.

Naicken, S., Livingston, B., Basu, A., Rodhetbhai, S., Wakeman, I., and Chalmers, D. (2007). The state of peer-to-peer simulators and simulations. *SIGCOMM Comput. Commun. Rev.*, 37(2):95–98.

Ravoaja, A. and Anceaume, E. (2007). Storm: A secure overlay for p2p reputation management. *Self-Adaptive and Self-Organizing Systems, International Conference on*, 0:247–256.

Roussopoulos, M., Giuli, T., Baker, M., Maniatis, P., Rosenthal, D. S. H., and Mogul, J. (2004). 2 p2p or not 2 p2p? The 3rd Int'l Workshop on Peer-to-Peer Systems.

San Ting, N. (2003). A generic peer-to-peer network simulator. In *Proceedings of the 2002-2003 Grad Symposium, CS Dept, University of Saskatchewan*.

Schlosser, A., Voss, M., and Brückner, L. (2005). On the simulation of global reputation systems. *Journal of Artificial Societies and Social Simulation*, 9.

Schlosser, M. T., Condie, T. E., and Kamvar, S. D. (2003). Simulating a file-sharing p2p network. In *In First Workshop on Semantics in P2P and Grid Computing, 12th.WWWConference*.

Sun, M. (2010). The java swing api. Disponível em: http://java.sun.com/docs/api/javax/swing/package-summary.html. Acesso em: 30/07/2010.