FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

Vinicius Alexandre Pereira de Souza

SEGURANÇA EM REDES P2P

Americana, SP

2017

FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

Vinicius Alexandre Pereira de Souza

SEGURANÇA EM REDES P2P

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Segurança da Informação, sob a orientação do Prof. Esp. Marcus Vinícius Lahr Giraldi

Área de concentração: Segurança em Redes

Americana, SP.

2017

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

S719s SOUZA, Vinicius Alexandre Pereira de

Segurança em redes P2P. / Vinicius Alexandre Pereira de Souza. – Americana: 2017.

47f.

Monografia (Curso de Tecnologia em Segurança da Informação) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Esp. Marcus Vinícius Lahr Giraldi

1. Segurança em sistemas de informação 2.Redes de computadores I. GIRALDI, Marcus Vinícius Lahr II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.518.5

Vinicius Alexandre Pereira de Souza

SEGURANÇA EM REDES P2P

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Segurança da Informação, pelo CEETEPS/Faculdade de Tecnologia – Fatec/ Americana.
Área de concentração: Segurança em Redes

Americana, 26 de junho de 2017.

Banca Examinadora:

Marcus Vinícius Lahr Giraldi (Presidente)
Especialista
Fatec Americana

Wagner Siqueira Cavalcante (Membro)
Mestre
Fatec Americana

Paula da Fonte Sanches (Membro)
Mestre
Fatec Americana

AGRADECIMENTOS

Gostaria de agradecer aos meus pais, Isabel e Luiz, que sempre estiveram ao meu lado, me oferecendo conselhos, apoio e carinho, não só na elaboração desse trabalho, mas ao longo de toda minha vida.

Também agradeço a minha namorada Mayara, que sempre esteve ao meu lado, me dando forças sempre me incentivando com muita paciência e carinho.

E ao meu orientador, professor Marcus Lahr, que mesmo com as dificuldades enfrentadas ao longo do projeto, me apoiou, sem medir esforços para me auxiliar sempre que necessário.

DEDICATÓRIA

Dedico este trabalho à minha mãe, Isabel, “In Memorian”, e a meu pai, Luiz, pelo apoio, amor e carinho sem os quais não teria conseguido chegar até aqui.

RESUMO

O presente texto apresenta diversos conceitos sobre redes *Peer-to-peer*, ou P2P, abrangendo aspectos organizacionais e de classificação, propriedades, tais como o controle de entrada e saída de nós da rede, métodos de roteamento, e quais as principais ameaças e formas de se defender de ataques. Ele abrange as principais propriedades das redes P2P não estruturadas, explicando o funcionamento das abordagens centralizadas, distribuídas e híbridas, e mostrando as particularidades de cada abordagem. Também aborda as redes P2P estruturadas, explicando conceitos adotados neste tipo de rede *Peer-to-peer*, como o uso de *hash* consistente e de tabelas de *hash* distribuídas. O texto adota o Chord para exemplificar o funcionamento das redes P2P estruturadas, descrevendo sua geometria, controles de entrada e saída de nós, forma de roteamento e atualização das tabelas de hash. O texto explica quais as principais ameaças às redes P2P, estruturadas e não estruturadas, classificando os ataques de acordo com a camada funcional, efeito nas vítimas, objetivos do atacante e impacto causado, utilizando, como exemplo os ataques Sybil, Eclipse, de envenenamento de índice e tabelas de roteamento, e de inundação, demonstrando quais as maneiras de se defender destes.

Palavras Chave: Overlay, P2P, Chord, Eclipse, Sybil

ABSTRACT

The present text exhibit different concepts about Peer-to-Peer networks, or P2P, including its organizational aspects, classification and properties such as the network's nodes input and output control, routing methods, the main threats and how to defend yourself. It covers the main properties of the unstructured P2P networks, explaining the operation of the centralized, distributed and hybrid approaches, and their particularities. It also approaches the structured P2P networks, explaining the concepts adopted in this kind of Peer-to-peer network, such as the consistent hash and the distributed hash tables. The text approaches Chord to exemplify the operation of structured P2P networks, describing its geometry, nodes' input and output controls, routing, and hash tables update. The text explains which are the threats to both structured and unstructured P2P networks, classifying the attacks based on the caused impact, using for example, Sybil, Eclipse, index poisoning, routing tables poisoning and flooding attacks, demonstrating how to defend yourself.

Keywords: *Overlay, P2P, Chord, Eclipse, Sybil*

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Inúmeras aplicações globais descentralizadas que fazem uso da infraestrutura de Internet surgiram nos últimos anos. Essas aplicações apresentam grande capacidade de processar dados oriundos de milhões de usuários compartilhando conteúdo, recursos e se comunicando simultaneamente. Essas aplicações podem ser classificadas como P2P (*Peer-to-Peer*), pois eliminam a necessidade de servidores para intermediar a conexão entre os membros da rede.

Dada a organização das redes P2P, é possível considerá-las redes sobrepostas, uma vez que geram uma organização de rede virtual sobre uma infraestrutura subjacente já existente.

Utilizar a camada de aplicação para criar redes sobrepostas não é um recurso inédito, e já tem sido muito aplicado ao longo dos anos para criar diversos serviços.

As Redes Sobrepostas Resilientes (*Resilient Overlay Networks* ou RONs) por exemplo, são redes construídas sobre a infraestrutura da Internet com o objetivo de monitorar a vivacidade e a qualidade das rotas entre seus nós. Conforme descrito por Andersen (2001), esse tipo de aplicação pode ser utilizada para criar métricas de rotas.

O *Simple Mail Transfer Protocol* (SMTP) também pode ser enquadrado como uma rede sobreposta, uma vez que clientes e servidores de e-mail estabelecem conexões de rede entre si de maneira direta, através da camada de aplicação.

O Diferencial do P2P em relação a outras aplicações que fazem uso de redes sobrepostas está no fato do P2P utilizar esse recurso para criar serviços finais para usuários.

A popularização dos sistemas P2P se deu com a rede centralizada do Napster, e com outras soluções para o download de conteúdo, tais como o Gnutella, KaZaa, e posteriormente, o *BitTorrent* o *Bitcoin* que revigoraram o interesse da comunidade no estudo e evolução dos sistemas P2P.

1.1 Objetivo do Trabalho

O objetivo deste trabalho consiste em explicar o funcionamento, a organização e a taxonomia das redes *Peer-to-Peer* (P2P, ou Pessoa-para-Pessoa em português), abordando aspectos gerais como a organização, o roteamento, as operações de busca, mecanismos para entrada e saída de nós da rede, entre outras propriedades.

A partir do embasamento inicial acerca do funcionamento das redes P2P de maneira geral, serão abordados os ataques mais comuns a este tipo de rede e formas de remedia-los.

1.2 Justificativa

O uso do P2P ao longo dos últimos anos, em softwares para compartilhamento de arquivos, moedas virtuais, e redes de tráfego anônimo, levou as pessoas acreditar que a tecnologia tem seu uso resumido a aplicações ilícitas, ou no mínimo de legalidade questionável.

Entretanto a tecnologia tem muito a oferecer, algoritmos baseados em Tabelas de *Hash* Distribuídas (DHT), que tiveram suas motivações para pesquisa inicialmente focadas em P2P, atualmente são aplicados à Big Data, como apresentado por Featherston (2010).

Serviços como o *Skype* e *Spotify* utilizaram arquiteturas P2P em seus primórdios, conforme apresentado por Guha e Daswani (2006) e Kreitz e Niemelä (2010) respectivamente.

A compreensão da tecnologia é uma das maneiras de se quebrar o estigma criado, e visualizar os benefícios, os desafios e o futuro da tecnologia para a computação.

2 REDES P2P

2.1 Definição de Redes P2P

Redes *Peer-to-Peer* (P2P) receberam diversas definições em artigos e trabalhos acadêmicos ao longo dos anos, Buford e Yu (2010) apresentam, duas definições que cobrem os conceitos mais comumente encontrados em redes *Peer-to-Peer*, como compartilhamento de recursos, auto-organização, descentralização e conexão entre os nós:

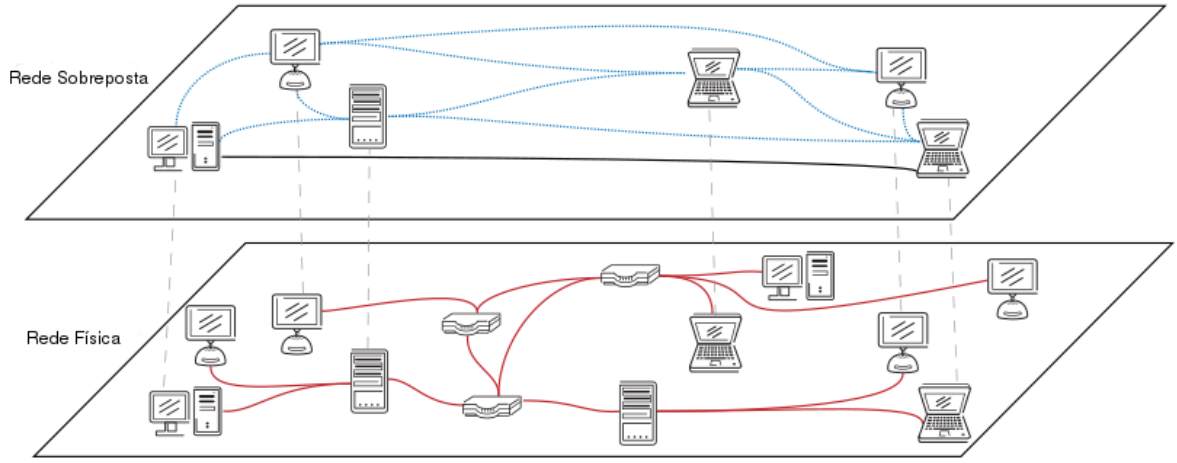
> "Uma rede de arquitetura distribuída pode ser chamada de rede *peer-to-peer*, se os participantes compartilharem parte de seus próprios recursos de hardware (poder de processamento, capacidade de armazenamento, capacidade de conexão de rede, impressoras). Esses recursos compartilhados são necessários para prover o serviço e conteúdo oferecidos pela rede (por exemplo compartilhamento de arquivos ou áreas de trabalho compartilhadas para colaboração). Eles são acessíveis por outras pessoas".
>
> (SCHOLLMEIER, 2001; apud BUFORD; YU, 2010 p. 6)
>
> "Sistemas *Peer-to-peer* são sistemas distribuídos que consistem em nós interconectados capazes de se auto organizar em topologias de rede com os objetivos de compartilhar recursos como conteúdo, ciclos de CPU, armazenamento e largura de banda, capazes de se adaptar a falhas e acomodar populações transientes de nós ao mesmo tempo que mantem conectividade aceitável e performance, sem requerer a intermediação ou o suporte de um servidor global centralizado ou de autoridade".
>
> (ANDROUTSELLIS; SPINELLIS, 2004; apud BUFORD; YU, 2010 p. 6)

2.2 Definição de Redes Sobrepostas (*Overlays*)

Redes sobrepostas são redes virtuais, lógicas, construídas sobre a infraestrutura de uma rede física subjacente, em sistemas distribuídos globalmente, são construídas sobre a infraestrutura da Internet. Nas redes sobrepostas, os participantes usam uns aos outros como roteadores para o tráfego dos dados. Os nós da rede sobreposta podem ou não estar conectados diretamente por meio da rede subjacente, a comunicação se dá por túneis de conexões virtuais entre os nós, conforme observado na figura 1.

Redes sobrepostas são utilizadas como mecanismo para implementar novos serviços não existentes na rede subjacente.

Figura 1 – Conexão lógica entre os nós numa rede sobreposta, e a conexão física na rede subjacente.



Fonte: (HAJIARABDERKANI, 2015, p. 5)

2.3 Propriedades das Redes P2P

Uma série de características comuns às redes P2P podem ser observadas, Buford e Yu (2010) destacam algumas.

2.3.1 Compartilhamento de Recursos

Cada membro da rede contribui com recursos próprios para a manter a operação da rede. Idealmente, cada membro da rede deve contribuir com recursos na mesma proporção que consome, no entanto, existem os *free-riders*, usuários que consomem recursos da rede sem uma contribuição equivalente.

2.3.2 Rede

Todos os membros da rede estão interconectados entre si, num modelo que pode ser representado através de um grafo. Quando um nó ou um grupo de nós fica isolado do resto da rede, pode-se dizer que a rede está particionada, ou segmentada.

2.3.3 Descentralização

O comportamento da rede se dará em função das ações dos nós desta, não existindo um servidor central que controle a rede. Há situações onde existirão servidores centralizados para controle de autenticação ou para executar operações específicas, como a busca por recursos.

2.3.4 Simetria

Em contraste com o sistema de cliente/servidor, onde os papéis dos membros da rede são distintos e bem definidos, as redes P2P de maneira geral, possuem simetria, todos os nós possuem os mesmos papéis dentro da rede sobreposta. Em alguns *designs* de redes *Peer-To-Peer*, essa propriedade não é seguida à risca, principalmente nos casos onde existem super-nós, ou nós de retransmissão.

2.3.5 Autonomia

Cada membro determina suas próprias capacidades de contribuir para a rede de acordo com seus recursos disponíveis. Também fica a critério de cada membro quando este irá se juntar ou deixar a rede. Essa propriedade é responsável em especial pela imprevisibilidade dos serviços oferecidos na rede sobreposta.

2.3.6 Auto-Organização

Diz respeito sobre como cada membro da rede utiliza o conhecimento que possui em relação aos demais para se organizar, e trabalhar em conjunto, suportando a operação da rede. A auto-organização, no entanto, não é uma

propriedade observada em muitas das redes P2P, como nas redes com topologia de estrela ou de propagação.

2.3.7 Escalabilidade

Para um sistema P2P ser considerado escalável, os recursos utilizados por nó devem apresentar uma taxa de crescimento não linear, menor que a taxa de crescimento da rede. O tempo de resposta dentro da rede também não deve apresentar um crescimento linear.

2.3.8 Estabilidade

Possuindo uma taxa máxima de *churn*, a rede deve ser capaz de manter a coesão do grafo, e efetuar o roteamento de forma determinística[1], com um número praticável de saltos durante o roteamento.

2.4 Taxa de Churn e Manutenção do Overlay

Peers[2] podem entrar ou deixar a rede a todo o momento, de acordo com Buford e Yu (2010), devem existir protocolos no *overlay* responsáveis por controlar a entrada e saída de membros do *overlay*, garantindo que vizinhos de membros que saíram sejam informados, e que novos membros consigam se conectar a seus vizinhos.

Quando um membro entra no *overlay*, ele recebe possui um estado inicial de roteamento, esse estado sofre mudanças conforme o funcionamento do *overlay*. Um *peer* ao deixar a rede, pode notificar seus vizinhos, caso essa notificação não seja

[1] Em Ciência da Computação, algoritmos determinísticos são aqueles que produzirão as mesmas saídas sempre que receberem as mesmas entradas, conforme apresentado por Bocchino (2009)
[2] *Peer* é um sinônimo para nó, ou membro

gerada, os demais membros podes verificar a saída do *peer* através de verificações de *heartbeat*[3].

Essas mudanças na população do *overlay*, podem ser identificadas com o termo *churn* de acordo com Buford e Yu (2010), que além de conceituar o termo, também explicitam de forma breve o que é a manutenção do *overlay*.

> "*Churn* é a entrada ou saída de *peers* no/do *overlay*, gerando mudanças na população de *peers* do overlay.
>
> Manutenção do *overlay* é a operação para reparar ou estabilizar o estado de roteamento do overlay em resposta ao *churn*. "
>
> (BUFORD; YU, 2010, p. 19)

2.5 Organização e Taxonomia

As diversas arquiteturas que as redes P2P podem possuir, culminaram na criação de diversos tipos de classificação. Alguns autores e acadêmicos como Xie (2003) propõe a divisão dos sistemas P2P gerações. Buford e Yu (2010) acreditam que essa abordagem, no entanto, é pouco eficaz, uma vez que redes de diferentes gerações estiveram em uso num mesmo momento, além do mais, a simples divisão por gerações não leva em consideração propriedades importantes das redes, como a topologia, métodos de busca e roteamento. Apesar do não consenso sobre a divisão ou não das redes P2P em gerações, existe um consenso sobre a classificação das redes em estruturadas e não estruturadas.

[3] *Hearbeat* é um sinal de checagem periódico gerado para verificar o funcionamento normal de um membro da rede

3 REDES NÃO ESTRUTURADAS

Esse tipo de rede não possui uma forma de organização definida para os *peers*, sua forma de organização se assemelha a de um grafo aleatório. Devido a maneira como os *peers* se organizam em redes deste tipo, são necessários algoritmos específicos para realizar buscas de uma forma otimizada.

Uma definição para Redes não estruturadas:

> "Um *overlay* onde um nó confia apenas em seus nós adjacentes para entregar mensagens para os demais nós do *overlay*. Exemplos de estratégias para a propagação de mensagens são o uso de *flooding* e *random walk.* "
>
> (BUFFORD, YU, LUA; 2009, p. 365)

Jin e Chan (2010, p. 119) classificam quatro abordagens que podem ser adotadas na organização de redes não estruturadas, e citam para cada uma delas um exemplo prático adotado.

3.1 Abordagem Centralizada

Para a abordagem centralizada, o exemplo utilizado é o Napster, que surgiu em 1999, como um *software* para compartilhamento de músicas.

De acordo com Jin e Chan (2010, p. 119-120), a abordagem centralizada, apresenta uma implementação simples e funcional. Nesse modelo, há um servidor central que fica responsável por indexar os arquivos disponíveis na rede, e associar esses arquivos a usuários específicos.

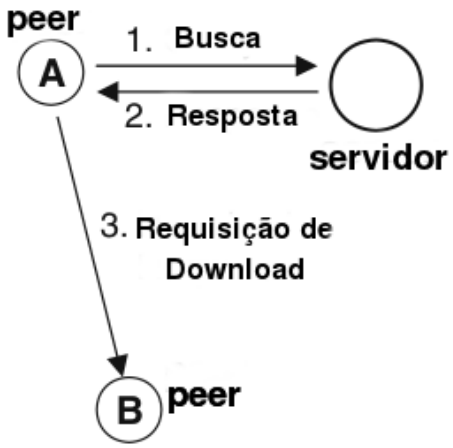
Quando um novo usuário entra na rede, ele contata o servidor central, informando uma lista de arquivos disponíveis, que podem ser solicitados por outros membros da rede.

Quando um usuário busca um arquivo, uma requisição é enviada para o servidor, que retorna uma lista com os usuários que possuem o arquivo. O usuário pode então iniciar o *download* do arquivo solicitado a partir dos membros da rede que o possuam. Esse fluxo pode ser observado na figura 2.

A grande desvantagem do modelo centralizado, é na questão da escalabilidade, para suportar um grande número de nós, o servidor precisa de uma quantidade muito grande de recursos de *hardware* e banda. Além disso, em caso de falha do servidor, a rede entra em colapso.

Por fim, como desvantagem Jin e Chan (2010, p. 120) ainda citam a vulnerabilidade a ataques de negação de serviço.

Figura 2 – Modelo exemplificando o funcionamento da abordagem centralizada.



Fonte: (JIN; CHAN, 2010, p. 120)

3.2 Abordagem Distribuída

No modelo distribuído, quando um *peer* entra na rede, ele se conecta inicialmente há alguns *peers* públicos já existentes[4]. O novo membro da rede então envia para seus vizinhos uma mensagem informando que este entrou na rede, e estes propagam a mensagem para outros membros da rede. Para o novo membro, é retornada uma resposta, com a lista de arquivos dos vizinhos e porta utilizada para estabelecer as conexões. Além disso, é comum que os membros enviem mensagens de checagem para seus vizinhos para verificar se estes ainda são membros da rede.

O Gnutella trabalha desta maneira, sendo um dos melhores exemplos de redes P2P totalmente distribuídas.

> "No protocolo Gnutella, quando um peer entra no sistema, ele primeiramente conecta alguns peers públicos.
>
> […] Um novo peer então envia uma mensagem de PING para qualquer peer ao qual esteja conectado. A mensagem anuncia a existência do novo peer. Ao receber uma mensagem PING, um peer do Gnutella retorna uma mensagem PONG e propaga a mensagem PING para seus vizinhos. Uma mensagem PONG contém o endereço IP e a porta do peer que respondeu, e informações de arquivos sendo compartilhados. Numa rede dinâmica com frequente entrada e saída de peers, periodicamente um peer envia mensagens de PING para seus vizinhos"
>
> (JIN; CHAN, 2010, p. 121)

O Gnutella, assim como outros sistemas P2P que utilizam uma abordagem distribuída, faz buscas na rede por meio de *broadcast* ou propagação. Um *peer* que deseja buscar um determinado arquivo, envia uma mensagem de busca para seus vizinhos, que repassam a consulta para seus vizinhos e assim sucessivamente. Caso algum *peer* possua o arquivo solicitado, ele envia uma mensagem de volta, para o emissor original da mensagem de busca.

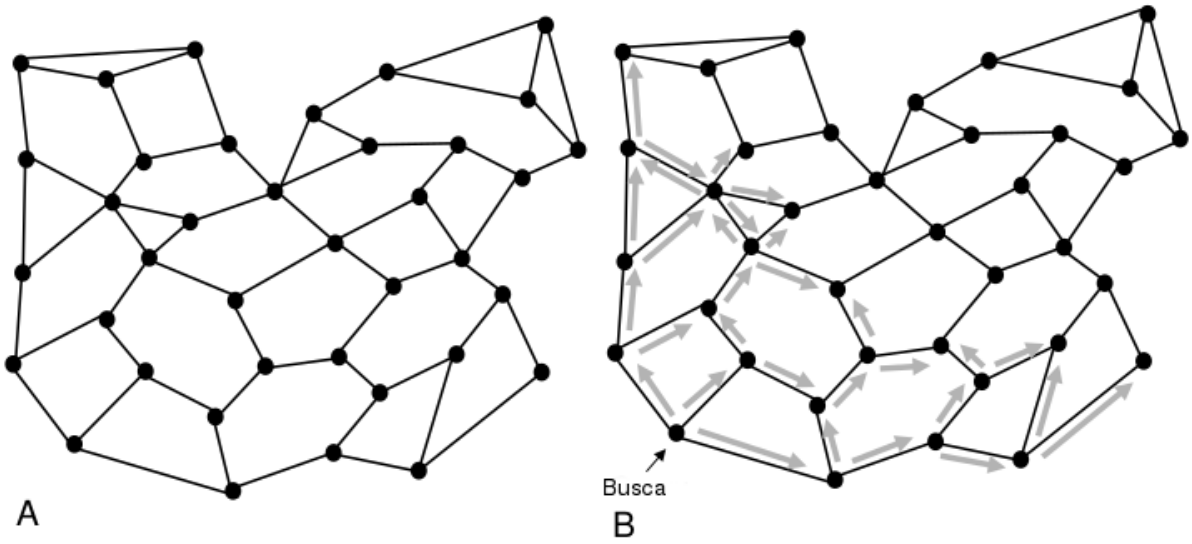
Para reduzir o número de mensagens de busca circulando pela rede, cada mensagem de busca possui um *time-to-live* (TTL) definido. A cada repasse da mensagem por um intermediário, o TTL é reduzido em um. Quando o TTL atinge zero, a mensagem de busca para de ser propagada. Esse método de busca é chamado de *flooding* ou inundação.

---

[4] Essa conexão inicial é conhecida como *bootstrapping* no Gnutella, e pode ser feita obtendo-se uma lista de peers online (Gnutella Web Cache), ou inserindo-se manualmente o endereço de membros da rede.

É possível observar na figura 3, o funcionamento da busca através de inundação com TTL em funcionamento, após quatro *hop counts*[5], a mensagem de busca não é mais propagada.

Figura 3 – A: mostra a organização dos *peers* numa rede totalmente distribuída e B: mostra a busca através de inundação com um TTL igual a 4.



Fonte: (BUFORD, YU, LUA; 2009, p. 47)

Um dos problemas da busca por meio de inundação, é que mesmo que o arquivo desejado seja encontrado logo no começo da busca, ela continuará se propagando entre outros nós até que o TTL atinja zero. Para contornar para esse problema, Buford, Yu e Lua (2009 p. 48) citam o uso do *expanding ring*, um método de busca através de propagação que é uma variação do *flooding*.

No *expanding ring*, a busca é iniciada com um TTL baixo, caso o arquivo não seja localizado, uma nova busca é iniciada com um TTL ligeiramente maior. Esse método é mais eficiente para buscar arquivos populares, que serão encontrados com poucos *hops*, evitando a propagação desnecessária de mensagens de busca

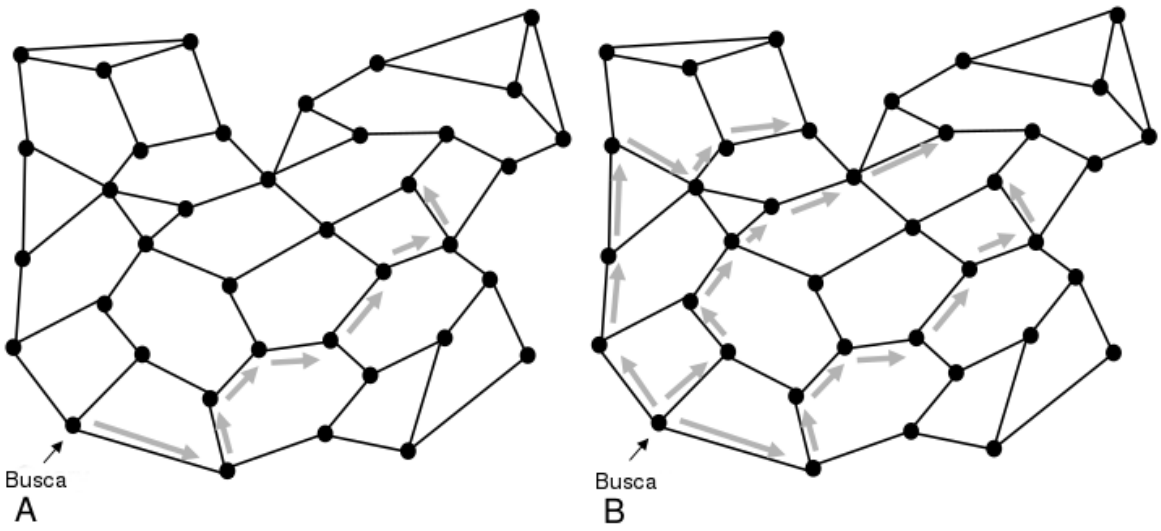
[5] *Hop count* representa o número de dispositivos de roteamento pelo qual um pacote passa (FEI et al, 2008)

pela rede, no entanto, se mostra igualmente ineficaz ao buscar arquivos pouco populares.

Um terceiro método de busca por meio de propagação é citado por Jin e Chan (2010) e Buford, Yu e Lua (2009), conhecido como *random walk*, ou caminhada aleatória. No *random walk*, basicamente um *peer* envia uma mensagem de busca para apenas um de seus vizinhos, de forma aleatória, e este, repassa para outro vizinho de forma também aleatória, e assim sucessivamente, até que o TTL chegue em zero (Figura 4-A).

Uma outra opção para aumentar a performance é executar o *random walk* de maneira paralela, isto é para mais de um vizinho de maneira simultânea (Figura 4-B).

Figura 4 – Demonstração do funcionamento do *random walk*.



Fonte: (BUFORD, YU, LUA; 2009, p. 49)

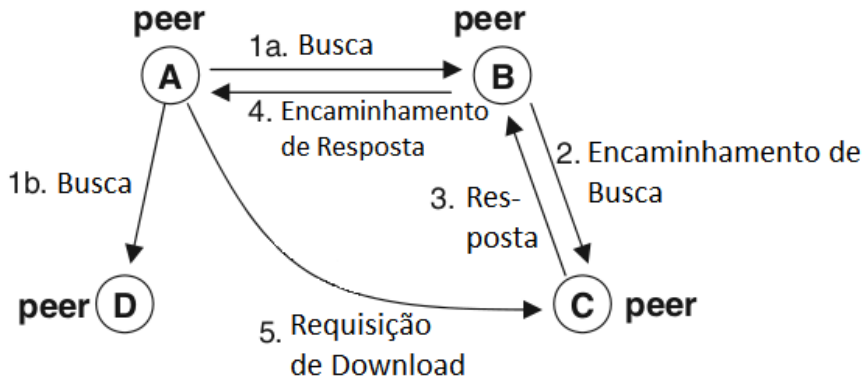
As redes distribuídas de maneira geral, possuem como grande vantagem sua capacidade de crescimento infinita, dinamismo, auto-organização e independência entre os nós, no entanto sua grande deficiência é a capacidade para busca de arquivos.

Utilizar o método de busca adequado e definir um valor para o TTL não é uma tarefa simples. Se o valor for baixo demais, as buscas podem na maior parte das vezes não encontrar os arquivos desejados, se for alto demais, pode inundar a rede

com mensagens de busca afetando a performance do *overlay*, e tornando o tempo de resposta da busca alto demais.

Pode-se observar na Figura 5, uma ilustração exemplificando o funcionamento do Gnutella, desde o envio das mensagens de busca até finalmente localizar o arquivo, estabelecer conexão com o *peer* e iniciar o *download*.

Figura 5 – Fluxo de funcionamento do Gnutella, desde a busca do arquivo até o início do *download*.



Fonte: (JIN; CHAN, 2010, p. 120)

3.3 Abordagem Híbrida

Dadas as limitações das abordagens totalmente distribuídas ou centralizadas, uma abordagem combinando aspectos de ambas é apresentado por Jin e Chan (2010, p. 122), identificada como híbrida, ou parcialmente centralizada.

Há certa controvérsia sobre o uso do termo “rede híbrida” em P2P. Jin e Chan (2010) adotam o termo rede híbrida para descrever redes que são distribuídas, mas fazem uso do recurso de *super-peers* para realizar a busca por arquivos.

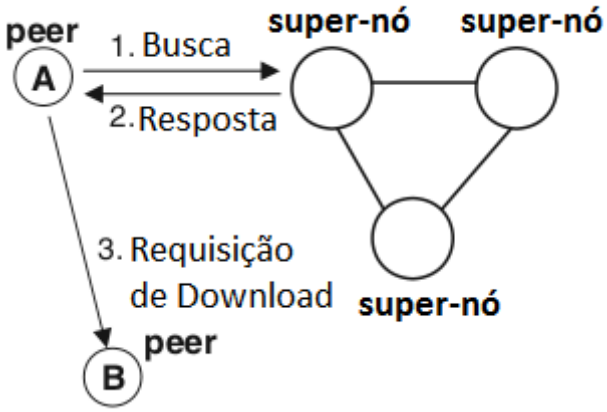
Outros autores como Buford e Yu (2010) e Yang e Garcia-Molina (2003) no entanto usam o termo “redes híbridas” considerando-as tal como modelo do Napster, onde de fato existe um servidor central que mantém metadados e que é responsável por efetuar a busca por arquivos, mas a transferência dos arquivos é feita a partir dos membros da rede.

Para este trabalho foi adotado o uso conforme proposto por Jin e Chan (2010), sendo o modelo do Napster uma rede centralizada, e o modelo baseado em *super-peers* o híbrido.

Nesse tipo de abordagem, *peers* que disponham de mais recursos computacionais e banda são designados como super-nós ou *super-peers*. Os *super-peers* são responsáveis por manter informações sobre recursos dos nós conectados a este, como também manter conexões com outros *super-peers*.

Na Figura 6, fica exemplificado o funcionamento deste tipo de arquitetura: um *peer* convencional envia uma mensagem de busca para o super nó ao qual está conectado, e este pode retornar imediatamente o endereço dos *peers* que possuem o arquivo solicitado caso esteja conectado a algum, ou fazer uma requisição para outros super-nós. Assim que obtiver uma resposta, o super nó retorna a lista de membros que possuem o arquivo solicitado.

Figura 6 – Fluxo de busca por arquivos no modelo híbrido.



Fonte: (JIN; CHAN, 2010, p. 120)

Essa abordagem para a realização de busca por arquivos é incorporada por *softwares* de compartilhamento como o Kazaa, FastTrack, e passou a ser adotada pelo Gnutella a partir da versão 0.6.

Em comparação com o modelo distribuído, a abordagem híbrida se destaca pela sua velocidade de busca, que é muito maior, uma vez que o número de super-nós na rede é muito inferior ao número total de membros, portanto as buscas não inundam a rede inteira. Já em comparação ao modelo centralizado, a vantagem se

dá por conta da inexistência de um ponto único de falha, se um *super-peer* ficar indisponível, basta que os *peers* conectados a este passem a se conectar a um novo super nó.

4 REDES ESTRUTURADAS

As redes estruturadas surgiram como uma proposta de solução para problemas apresentados nas redes não estruturadas no que diz respeito à escalabilidade e à capacidade de efetuar uma busca determinística por recursos na rede.

> "[…] Esses problemas originaram muitos designs de overlays com mecanismos de roteamento que são determinísticos e que podem prover garantias em relação a habilidade de localizar quaisquer objetos armazenados no overlay. A ampla maioria destes designs utilizaram overlays com geometrias de roteamento específicas e são chamados de overlays estruturados."
>
> (BUFFORD, YU, LUA; 2009, p. 75)

Segundo Dhara et al. (2010), para uma melhor compreensão das redes P2P estruturadas, alguns aspectos devem ser observados neste tipo de *overlay*, tais como a geometria apresentada pela rede e as formas de roteamento possibilitadas por esta, além do mecanismo para controle de *churn*, da manutenção do *overlay* em si e do mecanismo de *bootstrapping* adotado.

Buford, Yu e Lua (2009) e Dhara et al. (2010) citam diversos algoritmos que podem ser implementados em redes P2P estruturadas, tais como o Chord, CAN e Tapestry, todos com diferentes características em relação à geometria, particionamento do *overlay*, atribuição de identificadores aos nós, roteamento e controle de *churn*. No entanto, uma similaridade entre estes algoritmos se dá no fato de todos fazerem uso de tabelas de *hash* distribuídas e *hash* consistente[6].

[6] *Hash* consistente é um método de *hashing*, que diferente dos métodos convencionais, não exige que todas as chaves da tabela precisem ser re-mapeadas caso o número de registros na tabela sofra alterações

4.1 Tabelas de *Hash* Distribuídas

A Springer apresenta uma definição de Tabelas de *Hash* Distribuídas (*Distributed Hash Tables* ou DHT):

> “ Uma Tabela de *Hash* Distribuída (DHT) é um sistema descentralizado que prove a funcionalidade de uma tabela de *hash*, em outras palavras: inserção e recuperação de pares chave-valor. Cada nó no sistema armazena uma parte da tabela de *hash*. Os nós são interconectados numa rede estruturadas sobreposta, que possibilita entrega eficiente de requisições de busca e inserção de chaves do requisitante para o nó armazenando a chave. ”
>
> (GALUBA; GIRDZIJAUSKAS, 2009)

Tabelas de *Hash* Distribuídas mantém pares chave-valor assim como tabelas de *hash* convencionais. As redes P2P estruturadas normalmente implementam o algoritmo SHA-1 como função de *hash*.

Meriläinen (2008) destaca quatro algoritmos como os primeiros a ser desenvolvidos com base no conceito de DHT: Chord, CAN, Pastry e Tapestry.

Neste trabalho, serão abordadas as propriedades das redes P2P estruturadas tendo como referência o Chord.

4.2 Chord

O Chord é um algoritmo baseado em DHT e *hash* consistente desenvolvido por pesquisadores do Instituto de Tecnologia do Massachusetts (MIT) e da Universidade de Berkeley.

4.2.1 Geometria e Organização

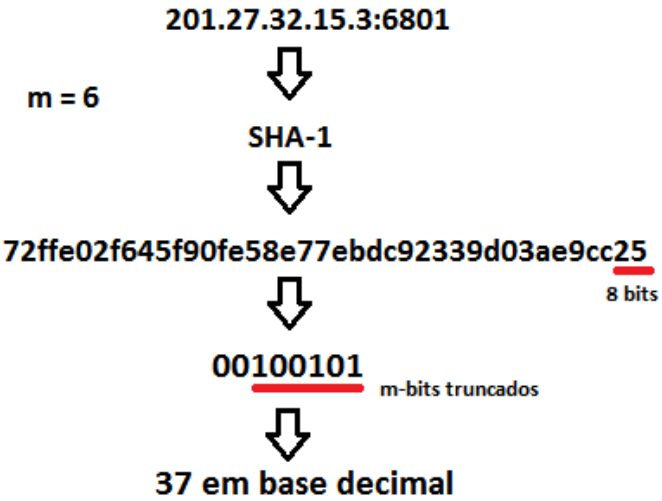
O Chord possui uma geometria circular, de maneira que os nós fiquem todos conectados através de um anel lógico, onde as buscas são realizadas em sentido horário (Stoica et al., 2003).

Cada nó da rede possui um identificador único, seu *NodeId*, que é gerado através da execução de uma função de *hash* SHA-1 em suas informações de

endereço IP e porta. As chaves dos objetos armazenados na rede também são geradas através da função SHA-1, normalmente através do *hash* do nome do arquivo.

Segundo Stoica et. al (2003) Tanto o *NodeId* quanto as chaves dos arquivos possuem um identificador de **m**-bits, onde **m** é um valor pré-definido pela rede que deve ser grande o suficiente afim de minimizar a possibilidade de colisões[7] nas funções de *hash.* O tamanho máximo de **m** é 160, devido a limitações da função de *hash* SHA-1. Podem existir na rede 2^{m} identificadores na rede.

Figura 7 - Processo para obtenção do NodeId numa rede onde m = 6.



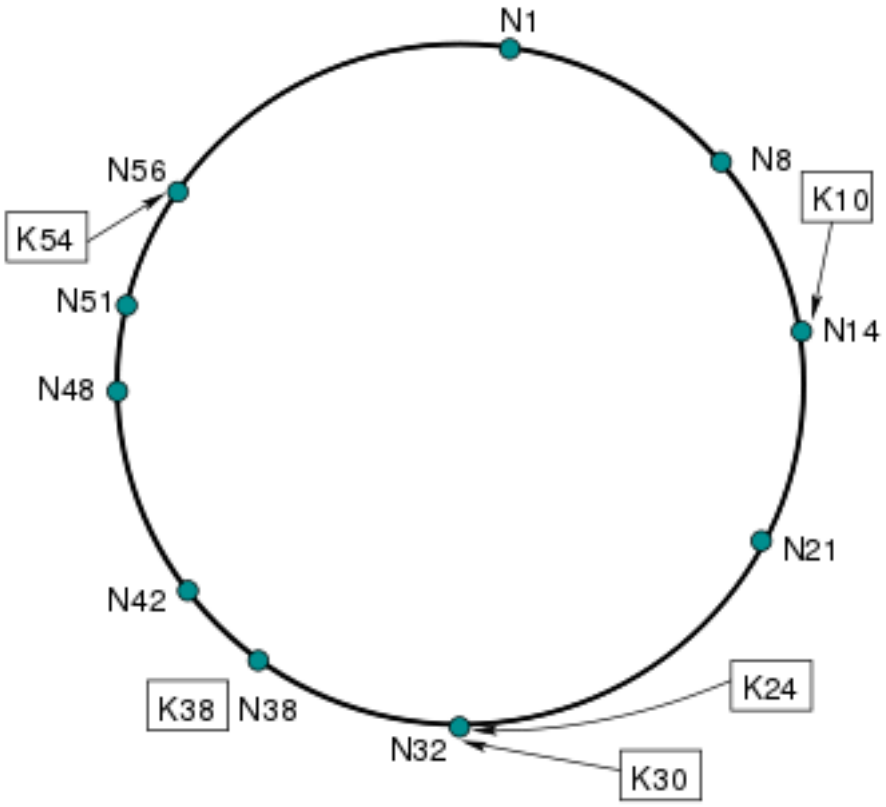
Fonte: Próprio Autor

Na figura 7, é possível observar os passos até a obtenção do *NodeId*, inicialmente gerando o *hash* a partir do IP e porta do nó, truncando esse valor em **m** bits e convertendo-os para decimal. A obtenção da chave de um arquivo segue a mesma linha de raciocínio, apenas substituindo o IP e porta pelo nome do arquivo.

Os nós e arquivos são distribuídos na rede de forma crescente de acordo com seus identificadores, em sentido horário. Pode-se observar na figura 8, uma rede com 5 chaves e 10 nós, onde m = 6.

[7] Uma colisão ocorre quando duas entradas distintas gera o mesmo valor de saída em uma função de *hash*

Figura 8 - Organização dos nós no Chord.



Fonte: (STOICA et. al, 2003)

No Chord, cada nó fica responsável pelos arquivos disponíveis numa fração do *overlay*, onde as chaves dos arquivos sejam menores ou iguais ao seu identificador de nó, e maiores que o identificador do seu vizinho em sentido anti-

horário, essa partição do *overlay* é denominada espaço identificado, de acordo com Stoica et al. (2003).

Seguindo essa regra, na figura 8, as chaves 24 e 30 serão localizados no nó de Id 32 e a chave 38 no nó de Id 38. Caso um novo nó de Id 26 se unisse a rede, o objeto de chave 24 passaria a ser localizado nesse novo *peer*.

4.2.2 Busca e Roteamento

Cada nó dentro do overlay possui três informações básicas para efetuar buscas e roteamento de mensagens de busca: seu nó predecessor, seu nó posterior, e sua tabela de roteamento, denominada *finger table*.

Tabela 1 – Informações de Roteamento de um nó

NOTAÇÃO	DEFINIÇÃO
Finger	[n]((x + $2^{(n-1)}$) mod 2^m), 1 <= n <= m
Sucessor	ó seguinte no círculo. *finger*[1].*node*
Predecessor	O nó anterior no círculo.

Fonte: (STOICA et. al, 2003)

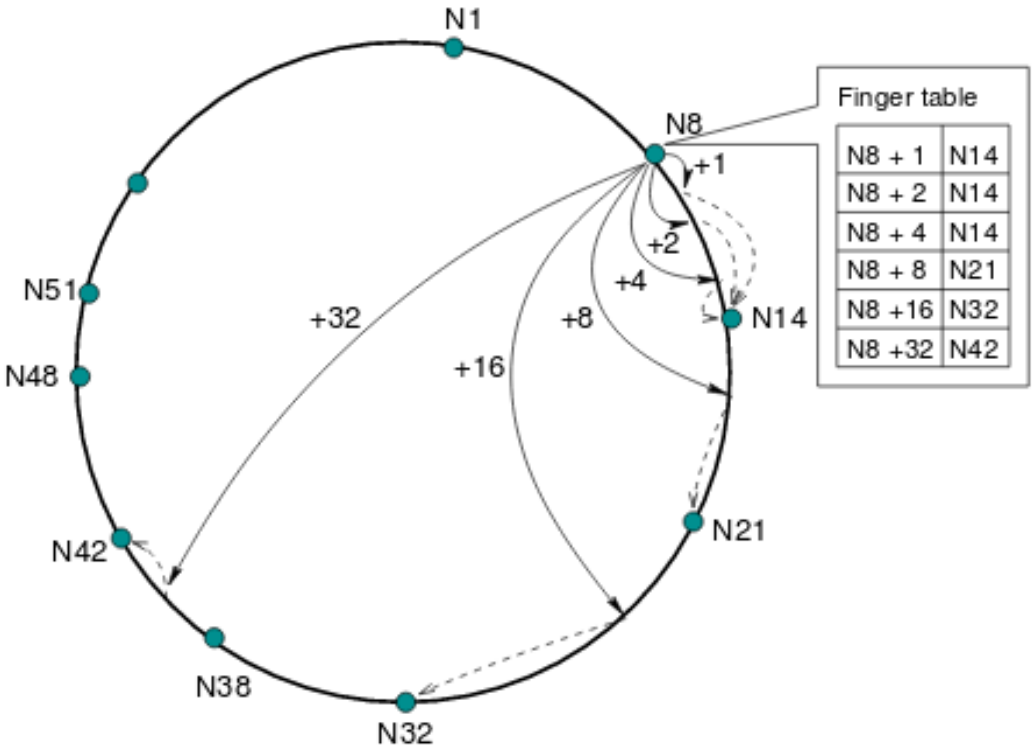
A criação do *finger table* apresentada na tabela 1 segue a notação ((x + $2^{(n-1)}$) mod 2^m), onde m, conforme abordado anteriormente é um valor pré-determinado pela rede, x é identificador do nó, e n refere-se ao n-ésimo valor da *finger table*, 1 <= n <= m. Cada nó da rede possuirá m registros em sua tabela de roteamento (Stoica et al. 2001).

Por meio das informações da *finger table*, um nó é capaz de localizar na rede arquivos, ou rotear requisições para outros nós. Como cada nó possui determinados

objetos de acordo com seu *NodeId*, através da chave de um objeto é possível determinar exatamente em qual nó este deve estar armazenado (Stoica et al. 2003).

É possível observar por meio da figura 9, que na tabela de roteamento de um nó de *Id* 8, que os três primeiros valores são preenchidos com o *Id* do nó 14. De acordo com a fórmula, o primeiro valor da tabela deveria possuir o Id do nó 9 ($8 + 2^{(1-1)} \bmod (2) = 9$), no entanto como não existe um nó na rede com esse identificador, é inserido o nó mais próximo desse valor que exista na rede, que é o *Id* 14. O mesmo se repete para as duas entradas seguintes da *finger table*, os nós de *Id* 10 e 12 não existem, logo é acrescentado o nó mais próximo, de *Id* 14. Nos dois registros seguintes, observa-se situação semelhante e são acrescentados os nós de *Id* 32 e 42 à tabela de roteamento.

Figura 9 - Tabela de roteamento no Chord onde m = 6.



Fonte: (STOICA et. al, 2003)

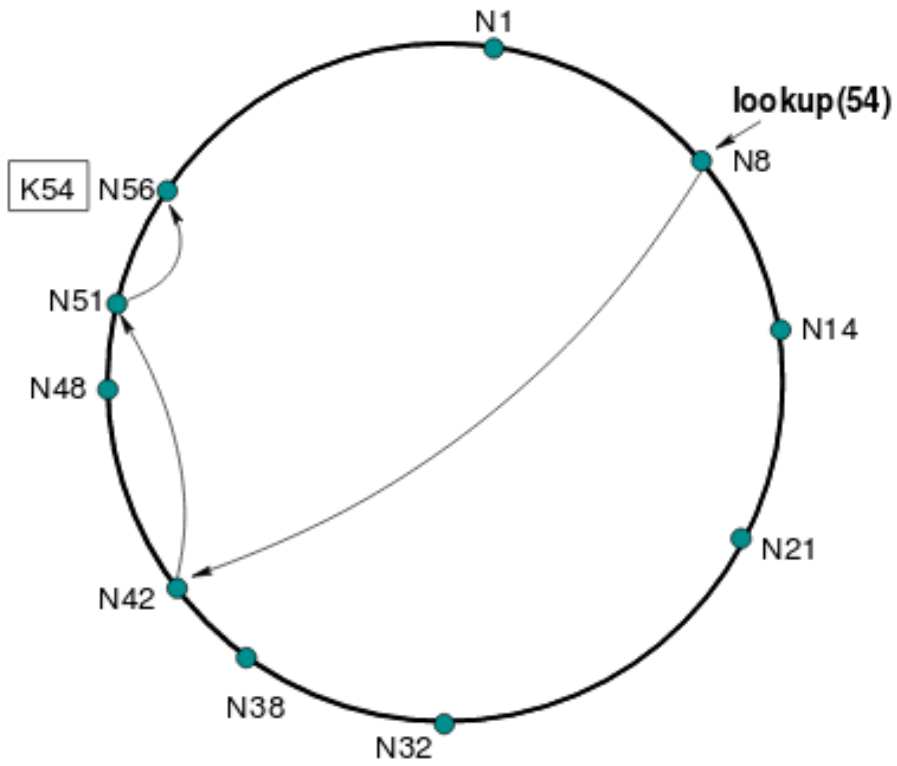
Por meio do *finger table*, é possível realizar buscas efetuando O(log N) *hops*, onde N é o número total de nós na rede.

Com a divisão que o Chord faz do *overlay*, distribuindo objetos entre os nós de acordo com seus *Ids*, é possível localizar um arquivo na rede com poucos *hops*, sem que seja necessário um índice de arquivos, ou extensas tabelas de roteamento.

As buscas no Chord são sempre executadas no sentido horário, quando o nó solicitante não possui em sua tabela de roteamento o *peer* responsável pela chave buscada, ele roteia a requisição para o nó com o *Id* mais próximo à chave que existir em sua *finger table*. A requisição é roteada até chegar o nó responsável pela chave, que então entrará em contato com o nó solicitante (Stoica et al. 2003).

Na Figura 10 observa-se o fluxo de uma busca no Chord, o nó de *Id* 8 inicia uma busca por um objeto na rede de chave 54. O *peer* com *Id* mais próximo que o nó 8 possui em sua tabela de roteamento é o nó 42, portanto a requisição é roteada, o nó 42 por sua vez possui o nó 51 em sua tabela de roteamento como o nó com *Id* mais próximo ao da chave, a requisição sofre mais um roteamento até finalmente chegar ao nó com o identificador 56, que é o responsável pelo objeto solicitado.

Figura 10 - Busca por um objeto com a chave 54 a partir do nó de Id 8.



Fonte: (STOICA et. al, 2003)

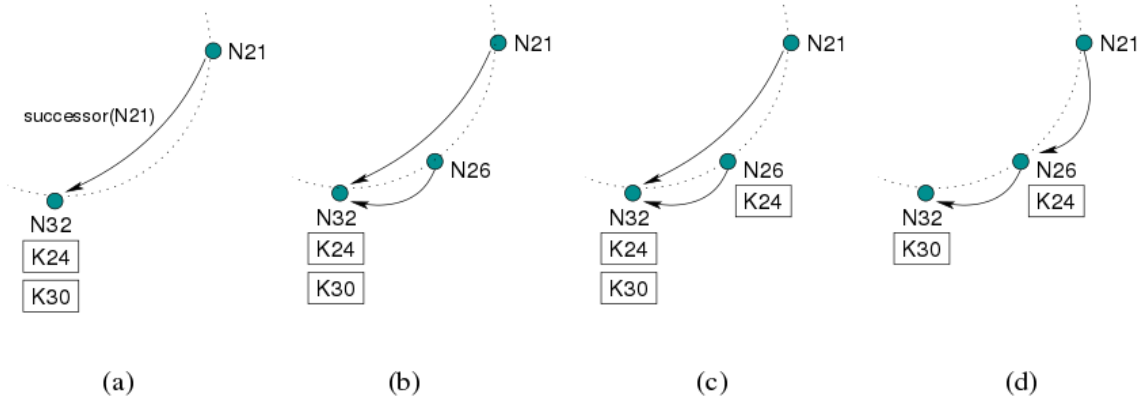
4.2.3 Controle de *Churn*

Por meio de um protocolo de estabilização, o Chord atualiza as *finger tables* e as informações de nó predecessor e posterior dos *peers* afetados por um novo membro se unindo ou deixando o *overlay*.

4.2.3.1 Entrada de nó na rede

Após adquirir um identificador e contatar um nó de *bootstrap* da rede, o novo membro da rede deve entrar em contato com o nó que será seu sucessor, e copiar todas as chaves que que estejam dentro de seu espaço identificado. Após esse processo, um protocolo de estabilização deve ser executado corrigindo as informações de nó predecessor e posterior e de *finger tables* dos vizinhos do novo *peer*, no sentido horário e anti-horário.

Figura 11 - Exemplo do processo de entrada de um nó na rede.



Fonte: (STOICA et. al, 2003)

Na Figura 11 é exemplificado o procedimento de entrada de um novo nó na rede: 11a é o estado inicial da rede, com o nó 32 possuindo em seu espaço identificado as chaves 24 e 30, e sendo o sucessor do nó 21, em 11b o novo nó de *Id* 26 localiza seu sucessor na rede, em 11c o nó 26 copia todas as chaves menores

ou iguais ao seu *Id*, e na figura 11d são executados os protocolos de estabilização, corrigindo as informações de nó sucessor, predecessor e *finger tables.*

4.2.3.2 Saída de Nó da Rede

Segundo Stoica et al. (2003) ”A precisão protocolo Chord confia no fato de cada nó conhecer seu sucessor”. Isso significa que em situações onde os nós falhem, a consistência do roteamento pode ser afetada.

Na Figura 10 por exemplo, caso os nós 14, 21 e 32 ficassem indisponíveis simultaneamente, o nó 8 não saberia que o nó 38 é seu sucessor, uma vez que ele não existe em sua tabela de roteamento. Para contornar esse problema, de acordo com Stoica et. al (2003), “Cada nó no Chord mantém uma lista de tamanho **r** com os seus r sucessores”, dessa maneira, se o sucessor de um nó fica indisponível, o nó pode substituir a informação do sucessor com o próximo registro da lista e então atualizar sua *finger table*.

Todos os nós na lista de sucessores teriam de falhar para afetar as operações de busca e roteamento, evento pouco provável, considerando uma lista de sucessores de tamanho razoável. De acordo com Dhara et. al (2010), numa lista de sucessores de tamanho S(log N), onde N é o número total de nós, há uma enorme probabilidade das buscas serem bem sucedidas, mesmo que cada nó possua 50% de chances de falha.

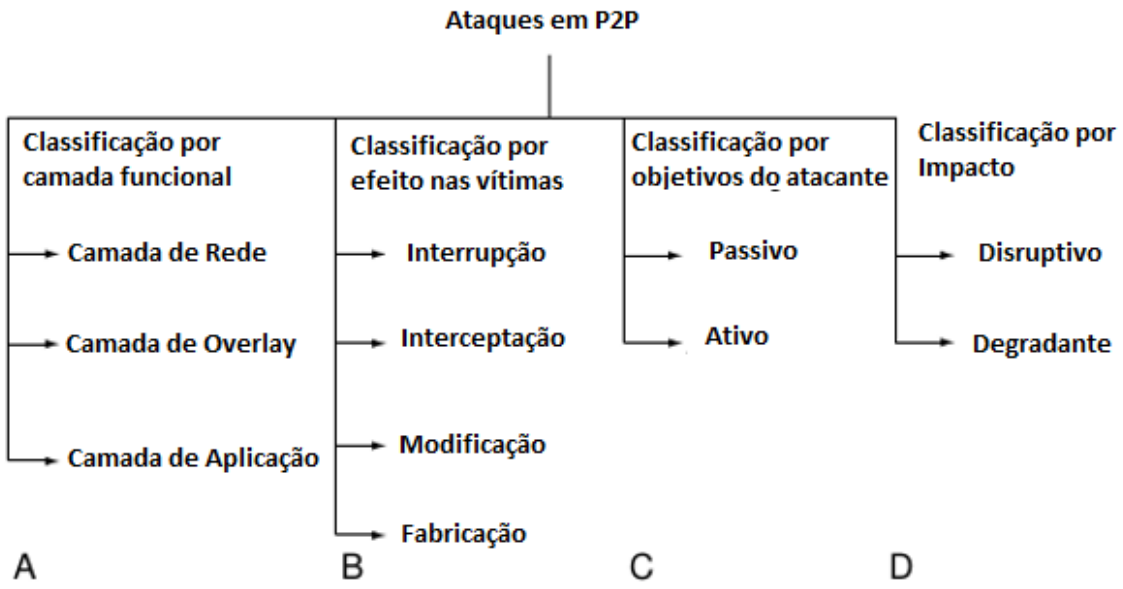
5 SEGURANÇA EM REDES P2P

Segurança é uma preocupação existente em todos os sistemas e meios de comunicação existentes, inclusive nas redes P2P.

5.1 Classificação dos Ataques

Na figura 12 é apresentada uma classificação dos ataques em redes P2P de acordo com a camada atacada (Figura 12A), e com o efeito causado nas vítimas (Figura 12B), com os objetivos do atacante (Figura 12C) e com o impacto do ataque (Figura 12D).

Figura 12 - Tipos de ataques a redes P2P.

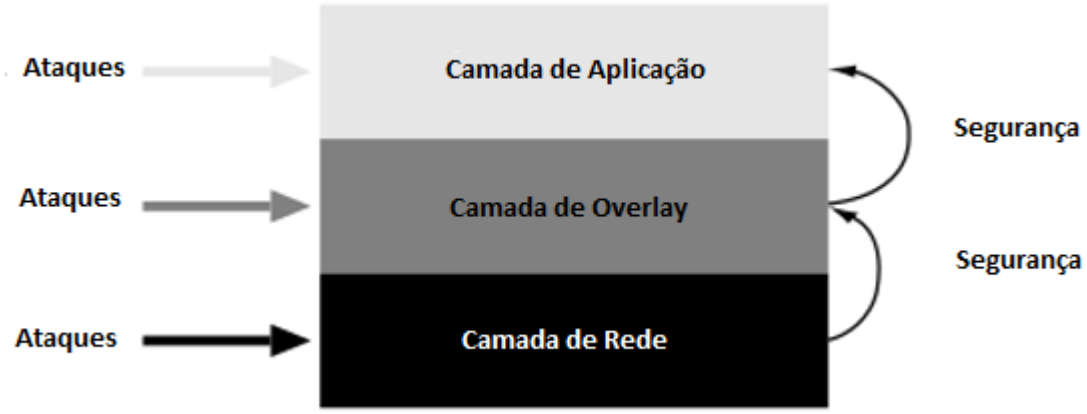


Fonte: (BUFORD; YU; LUA, 2009, p. 321)

5.1.1 Classificação por Camada Funcional

De acordo com Buford, Yu e Lua (2009), os ataques a redes P2P podem ter três camadas funcionais diferentes como alvo: a camada de rede, camada de *overlay* ou sobreposição, e a camada de aplicação. Além disso, a segurança das camadas superiores depende diretamente das camadas inferiores, portanto uma falha na camada de rede por exemplo, compromete as demais camadas acima, conforme observado na figura 13.

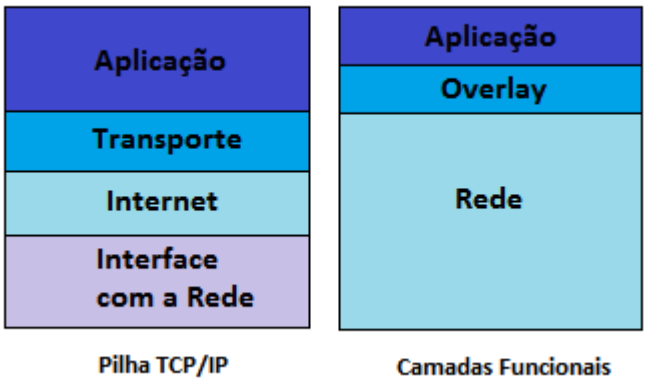
Figura 13 - Ataques às diferentes camadas funcionais.



Fonte: (BUFORD; YU; LUA, 2009, p. 321)

É importante ressaltar que as três camadas apresentadas por Buford, Yu e Lua (2009) não têm relação direta com as camadas da pilha TCP/IP, as camadas de aplicação e *overlay* propostas por Buford, Yu e Lua (2009) estão ambas na camada de aplicação da pilha TCP/IP, ao passo que a camada de rede se refere às demais camadas da pilha TCP/IP, conforme observado na figura 14.

Figura 14 - Relação entre as camadas funcionais e a pilha TCP/IP.



Fonte: Próprio Autor

Falhas de segurança exploradas na camada de aplicação dependem de uma interação direta do usuário com a interface do aplicativo. Buford, Yu e Lua (2009) destacam que, apesar da maior parte das ameaças ter origem em aplicações de compartilhamento de arquivos, riscos e ameaças também existem e podem ser explorados em outros tipos de *softwares* P2P.

Na camada de *overlay*, as ameaças têm como alvo a operação básica da rede, nas redes P2P estruturadas, visto que o maior ponto fraco se dá em relação ao roteamento com DHT. Uma vez que os nós confiam uns nos outros para manter suas tabelas de roteamento atualizadas, um *peer* malicioso que forje informações de roteamento pode degradar ou até mesmo interromper o funcionamento da rede.

Já as ameaças na camada funcional de rede são as que comprometem quaisquer outras aplicações que fazem uso de recursos de rede, e não somente as redes P2P, pois, “Esses ataques incluem interceptação de pacotes, manipulação do

conteúdo dos pacotes, ou roteamento incorreto de pacotes” (BUFORD; YU; LUA, 2009).

5.1.2 Classificação por Efeito nas Vítimas

Assim como as demais aplicações de rede, a comunicação no overlay das redes P2P está suscetível a quatro classes de ataques, segundo Stallings (1995, apud BUFORD; YU; LUA, 2009, p. 322):

- Interrupção: interrupção não autorizada
- Interceptação: acesso não autorizado
- Modificação: alteração não autorizada
- Fabricação: criação não autorizada

Seja interceptando comunicações, ou inserindo código malicioso na aplicação P2P em si, é possível expor os membros da rede à uma série de ameaças, como vírus, invasões, espionagem, trojans, roubo de informações, entre inúmeras outras.

5.1.3 Classificação Baseada nos Objetivos do Atacante

De acordo com Buford, Yu e Lua (2009, p. 322), um atacante em uma rede P2P tem sempre um propósito ativo ou passivo. No ativo, o invasor rouba dados e informações, pode alterar arquivos, forjar informações e até causar a interrupção do serviço. No passivo, o ataque consiste basicamente na análise do tráfego de rede.

5.1.4 Classificação por Impacto

Um ataque bem-sucedido por fim tem somente duas classificações de impacto possíveis: disruptivo, ou degradante. O serviço fornecido pela rede P2P pode ser completamente interrompido, ou pode funcionar de maneira precária após um ataque.

5.2 Ataques na Camada de *Overlay*

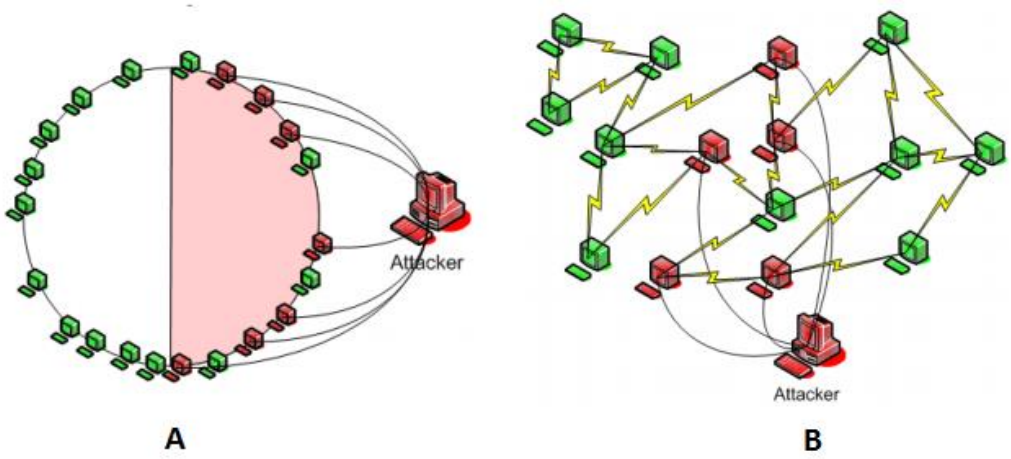
Conforme abordado anteriormente, as redes P2P são redes sobrepostas, ou seja, construídas sobre uma infraestrutura de rede já existente. De acordo com Buford, Yu e Lua (2009), essa camada adicional possui segurança limitada em relação à disseminação de mensagens, o que cria oportunidades para que *peers* maliciosos comprometam o funcionamento da rede.

5.2.1 Ataques *Sybil*

Segundo Douceur (2002), ataques Sybil podem ser definidos como “(...) um pequeno número de entidades forjando identidades de *peers* a fim de comprometer uma parte considerável do sistema. ”

Nas redes P2P estruturadas, cada nó deve ser único e ter um único identificador. Em um ataque *Sybil*, um mesmo nó assume várias identidades, atuando como se fosse vários *peers* na rede. De acordo com Buford, Yu e Lua (2009), ao assumir várias identidades distintas, um atacante passa a ter grande influência na rede, podendo restringir acesso de *peers* a certo conteúdo da rede, interceptando mensagens de roteamento, e inclusive podendo assumir o controle do *overlay*.

Figura 15 - Ataque Sybil em redes P2P estruturadas (A) e não estruturadas (B).



Fonte: (ENGLE; KHAN, 2006)

Na imagem 15 é ilustrado o ataque *Sybil*, em 15A, pode-se observar um ataque que comprometeu metade de uma rede P2P estruturada baseada no Chord, e na figura 15B, o ataque em uma rede P2P descentralizada, onde a rede ficou fragmentada.

Existem diversas propostas para minimizar a ameaça em relação aos ataques *Sybil*, Dinger e Hartenstein (2006) sugerem o uso de um processo chamado “Auto-registro”, onde o identificador de cada nó seria gerado à partir de seu endereço IP e porta utilizados para estabelecer conexão, após gerar seu registro, o nó solicita a entrada na rede, e membros da rede já registrados ficam incumbidos de validar a identidade do novo *peer*.

Outra proposta, de acordo com Mauch et al (2010), consiste no uso de desafios computacionais adaptativos. Um nó que solicitasse uma identidade, teria de resolver um desafio computacional antes. O servidor de *bootstrap* receberia a requisição de identidade, analisaria a taxa de recorrência da rede e da fonte, ou seja, com que frequência são solicitadas novas identidades, e então requisitaria a solução de um problema computacional. Redes ou *peers* que solicitem identidades com muita frequência teriam de solucionar problemas mais complexos, o que dificultaria a solicitação de muitas identidades.

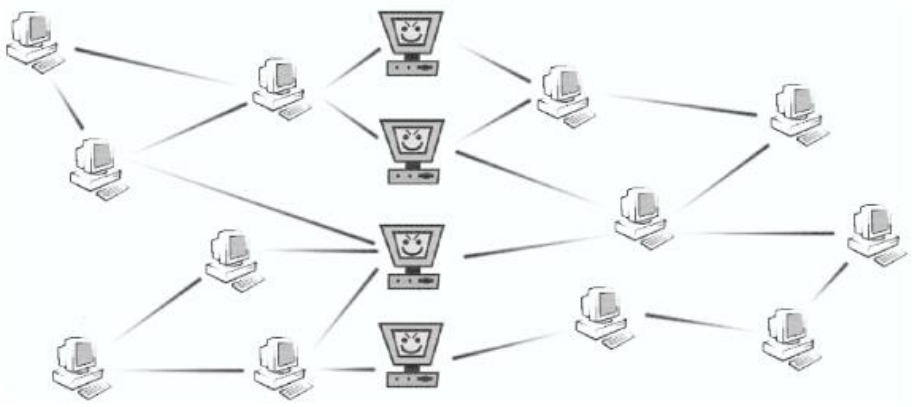
Deuceur (2002), acredita que não seja possível impedir ataques *Sybil* sem a existência de uma autoridade certificadora central, que seria responsável por atribuir às identidades aos membros da rede, e evitar que um mesmo host assumisse diferentes *Ids*.

5.2.2 Ataques Eclipse

Num ataque Eclipse, de acordo com Wang (2006) e Prêtre (2005), um atacante controla a maior parte dos nós vizinhos de um nó não malicioso. Nesta situação, um grupo de nós maliciosos age em conjunto, com o intuito de se tornarem vizinhos de um alvo, durante o processo de união deste à rede. Um ataque eclipse pode causar a fragmentação da rede, cessando a comunicação entre certas partes do *overlay*.

Um ataque Eclipse pode ser conduzido tanto em redes P2P estruturadas como não estruturadas. De acordo com Barcellos e Gaspary (2006), existem duas formas de lançar um ataque Eclipse: por meio de um ataque *Sybil*, e através da manipulação do algoritmo de roteamento. Devido à possibilidade de lançar um ataque através da manipulação do *overlay*, López-Fuentes, Eugui-de-Alba e Ortíz-Ruiz (2012) ressaltam que a defesa de um ataque *Sybil* não impede que um ataque Eclipse seja efetuado.

Figura 16 - Exemplo de um ataque Eclipse em uma rede P2P não estruturada, fragmentando o *overlay*.



Fonte: (PRETRÊ, 2005)

Algumas propostas para se defender de ataques Eclipse são abordadas por Singh et al. (2006). Redes P2P que utilizam algoritmos de roteamento mais rígidos, com regras de roteamento mais bem definidas como o Chord e o CAN devem estar protegidas, desde que possuam mecanismos de defesa à ataques *Sybil*.

Para Prêtre (2005), uma forma de defesa em redes P2P não estruturadas, consiste em não utilizar o modelo híbrido, e distribuir os *peers* de forma aleatória durante o processo de *bootstrapping*, reduzindo a probabilidade que um grupo de nós consiga dividir a rede. É importante frisar que mecanismos de defesa à ataques *Sybil* também se fazem necessários neste caso.

5.2.3 Ataques de Roteamento

Em redes P2P estruturadas, cada nó mantém uma tabela de roteamento própria, utilizada para realizar a busca por recursos no *overlay*. Um membro mal-intencionado da rede pode efetuar o roteamento de mensagens de busca para nós incorretos.

De acordo com Wang (2006) esse tipo de ataque é fácil de ser contornado, devido ao funcionamento do DHT utilizado nas redes P2P estruturadas, a cada roteamento, a chave do nó deve se aproximar da chave do arquivo solicitado, caso o *peer* que iniciou a busca perceba esse comportamento anormal, basta solicitar ao *peer* anterior por uma rota alternativa.

5.2.4 Ataques de Negação de Serviço

Ataques de negação de serviço (*Denial-of-Service* ou DoS) são uma forma comum de ataque, difíceis se precaver, tanto em P2P quanto em outros sistemas.

Nesse tipo de ameaça, os atacantes fazem requisições para um determinado serviço em um *host* de forma exaustiva, até que este não possa mais atender a requisições legítimas. Existem ainda os ataques de negação de serviço distribuídos

(*Distributed Denial-of-Service* ou DDoS), que são versões amplificadas do DoS, onde são utilizados vários *hosts* no ataque.

Redes P2P que utilizam uma abordagem centralizada são mais suscetíveis a ataques DDoS de acordo com Wang (2006) isso se dá devido ao ponto único de falha que é o servidor centralizado, responsável por indexar os arquivos existentes na rede.

Em redes P2P descentralizadas ou híbridas, os impactos são menores, normalmente reduzindo a performance da rede, mas sem causar indisponibilidade. Uma forma de ataque DDoS que pode ser observado em redes P2P é o ataque por *Query Flooding*, ou Inundação.

5.2.4.1 Ataques por Inundação

Os ataques por inundação também podem ser conduzidos a partir da camada de aplicação em redes P2P descentralizadas. De acordo com Wang (2006) no Gnutella por exemplo, para efetuar buscas por arquivos na rede, as queries são propagadas entre os vizinhos de um determinado nó. Um membro mal-intencionado da rede poderia ficar efetuando uma série de buscas com o intuito de inundar a rede com requisições.

Em redes P2P descentralizadas, o impacto de um ataque por inundação seria relativamente pequeno, uma vez que as buscas adotam um mecanismo de TTL, que evitaria à propagação das queries por toda a rede. No entanto, em redes P2P centralizadas, o impacto é enorme. Devido ao ponto único de falha, o servidor responsável por indexar os arquivos na rede poderia ficar indisponível no caso de um ataque por inundação efetuado por um número elevado de *peers*.

Daswani e Garcia-Molina (2002) apontam como solução aos ataques por inundação, que os nós da rede aceitem um número máximo de queries oriundas de um mesmo *peer*, e após atingir esse limite, as requisições passem a ser descartadas.

5.2.5 Ataques de Envenenamento

Segundo Wang (2006), redes P2P estão suscetíveis a ataques de envenenamento, como por exemplo: falsos endereços de IP, tabelas de roteamento adulteradas e falsos índices para arquivos.

5.2.5.1 Envenenamento de Índice

Conforme abordado anteriormente, redes P2P centralizadas utilizam um servidor para efetuar a busca por arquivos. Liang, Naoumov e Ross (2006) e Wang (2006) afirmam que é possível inserir informações falsas no índice do servidor.

Ao receber informações sobre novos arquivos, o servidor responsável pela indexação não verifica a autenticidade dos dados: se o endereço IP informado aceita conexões na porta indicada, e se possui o arquivo informado.

Num ataque de envenenamento de índice, um atacante poderia inserir uma série de registros falsos no índice do servidor, indicando que um nó na rede possui um conjunto de arquivos de grande popularidade. Nessa situação, o nó alvo receberia um grande número de conexões de outros membros da rede, solicitando arquivos. Seriam afetados tanto o nó alvo, que devido ao grande número de requisições poderia ficar sobrecarregado, quanto os demais membros da rede, que tentariam acessar arquivos inexistentes.

Liang, Naoumov e Ross (2006) acreditam que tentar contornar o problema através da validação dos registos inseridos no índice não seria uma solução bem-sucedida, invés disso, ele propõe a adoção de um sistema de autenticação na rede, ou então, a adoção de um sistema de reputação, e de listas negras.

5.2.5.2 Envenenamento da Tabela de Roteamento

Os ataques de envenenamento de tabela de roteamento são executados em redes P2P estruturadas que utilizam tabelas de *hash* distribuídas para localizar conteúdo no *overlay*, tais como o Chord. De acordo com Wang (2006) nestes

ataques, um nó malicioso pode enviar mensagens para atualização da tabela de roteamento de membros da rede indicando a entrada de um novo nó, que será o alvo do ataque.

Durante o processo de busca por objetos na rede, pacotes podem ser roteados para o alvo, numa rede extensa com muitos nós, um número elevado de requisições pode ser gerado, ao ponto de alvo ficar indisponível.

Este tipo de ataque não é “letal” para redes P2P, segundo Wang (2006) pois mesmo que valores falsos sejam inseridos nas tabelas de roteamento dos nós, as rotinas de estabilização do *overlay* devem se encarregar da remoção dos registros falsos.

6 CONSIDERAÇÕES FINAIS

Por meio desta pesquisa foi possível demonstrar o funcionamento das redes P2P, suas propriedades, formas de classificação e organização, além de ameaças mais comuns a este tipo de rede sobreposta.

O trabalho apresentou a divisão das redes P2P em redes estruturadas e não estruturadas, divisão esta, sendo a mais aceita no meio acadêmico. Foram apresentadas as vantagens e desvantagens de cada modelo arquitetural, e suas particularidades.

Foi abordado o conceito de tabelas de hash distribuídas, que é utilizado em redes P2P estruturadas, por meio do Chord. Existem diversos outros algoritmos baseados em DHT, conforme citado por Buford, Yu e Lua (2009) e Dhara et al. (2010).

O trabalho também levantou alguns dos ataques mais comuns a redes P2P estruturadas ou não, e mecanismos para defesa.

Em trabalhos futuros é possível abordar outros algoritmos baseados em DHT, e aprofundar os conceitos de segurança em P2P, através de dados quantitativos e demonstrações práticas. Além disso, é possível explorar outras áreas de pesquisa em P2P, como as criptomoedas por exemplo.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ANDERSEN, G. David. **Resilient Overlay Networks**. 2001. p. 33. Tese (Mestrado)- Department of Electrical Engineering and Computer Science, Massachusetts Institute Of Technology, [S.l.], 2001. Disponível em: < http://repository.cmu.edu/cgi/viewcontent.cgi?article=1082&context=compsci >. Acesso em: 17 mai. 2017.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - ABNT. **NBR 10520**: informação e documentação: citações em documentos: apresentação. Rio de Janeiro: ABNT, 2002. 7p.

BARCELLOS, M. P. Antonio; GASPARY, P. Luciano. **Segurança em redes P2P : princípios, tecnologias e desafios**. In: Simpósio Brasileiro de Redes de Computadores. 24. , 2006, Curitiba, Anais... Curitiba: Sociedade Brasileira de Computação, 2006. Disponível em: < http://www.lume.ufrgs.br/handle/10183/7510> . Acesso em: 22 mai. 2017.

BOCCHINO JR., Robert L. et al. **Parallel Programming Must Be Deterministic by Default**. [S.l.]: University Of Illinois, 2009. 1 p. Disponível em: . Acesso em: 22 nov. 2016.

BUFORD, John; YU, Heather; LUA, Eng Keong. **P2P Networking and Applications**. 1. ed. Burlington, EUA: Morgan Kaufman, 2009.

BUFORD, John; YU, Heather. Peer-to-Peer Networking and Applications: Synopsis and Research. In: BUFORD, John et al. (Org.). **Handbook of Peer-To-Peer Networking**. 1. ed. New York: Springer, 2010. cap I, p. 3-37.

DASWANI, Neil; GARCIA-MOLINA, Hector. **Query-Flood DoS Attacks in Gnutella**. In: ACM conference on Computer and communications security, 9., 2002, Washington. Nova Iorque: ACM, 2002. p. 181-192 Disponível em: < http://ilpubs.stanford.edu:8090/572/1/2002-65.pdf >. Acesso em: 17 mai. 2017.

DEUCEUR, R. John. **The Sybil Attack. In: International Workshop on Peer-to-Peer Systems**, 1., 2002, Berlin. Londres: Springer, 2002. p. 251-260. v. 2429. Disponível em: < http://ilpubs.stanford.edu:8090/572/1/2002-65.pdf >. Acesso em: 17 mai. 2017.

DHARA, Krishna et al. Overview of Structured Peer-to-Peer Overlay Algorithms. In: BUFORD, John et al. (Org.). **Handbook of Peer-To-Peer Networking**. 1. ed. Nova Iorque: Springer, 2010. cap. III, p. 223-254.

DINGER, Jochen; HARTENSTEIN, Hannes. **Defending the Sybil Attack in P2P Networks: Taxonomy, Challenges, and a Proposal for Self-Registration**. In: First International Conference on Availability, Reliability and Security (ARES'06), 1., 2006, Viena. Viena: IEEE, 2006 Disponível em: < https://gnunet.org/sites/default/files/10.1.1.60.8756.pdf >. Acesso em: 17 mai. 2017.

ENGLE, Marling; KHAN, I. Javed. **Vulnerabilities of P2P Systems and a Critical Look at their Solutions**. [S.l], 11 nov. 2006. Disponível em: < http://medianet.kent.edu/techreports/TR2006-11-01-p2pvuln-EK.pdf>. Acesso em: 17 mai. 2017

FEATHERSTON, Dietrich. **Cassandra: Principles and Application**. In International Conference on Computing, Engineering and Information, 2010, Urbana-Champaign. Urbana-Champaign: University of Illinois, 2010. Disponível em: < http://disi.unitn.it/~montreso/ds/papers/Cassandra.pdf>. Acesso em: 17 mai. 2017.

FEI, Aiguo et al. **MEASUREMENTS ON DELAY AND HOP-COUNT OF THE INTERNET**. [S.l.]: University Of California, 2008. 7 p. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.30.6547> . Acesso em: 22 nov. 2016.

GALUBA, Wojciech; GIRDZIJAUSKAS, Sarunas**. Distributed Hash Table**. 2009. Disponível em: <https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_1232>. Acesso em: 01 jun. 2017.

GUHA, Saikat; DASWANI, Neil. **An Experimental Study of the Skype Peer-to-Peer VoIP System**. In: International Workshop on Peer-to-Peer Systems, 5., 2006, Santa Barbara, EUA. p. 1-6. Disponível em: < https://ecommons.cornell.edu/bitstream/handle/1813/5711/TR2005-2011.pdf?sequence=1 >. Acesso em: 17 mai. 2017.

HAJIARABDERKANI, Masih. **Adaptative Dissemination of Network State Knowledge In Structured Peer-To-Peer Networks**. 2014. p. 33. Tese (PhD)-School of Computer Science, St. Andrews University, [S.l.], 2015. Disponível em: <https://core.ac.uk/download/pdf/30319034.pdf>. Acesso em: 22 nov. 2016.

JIN, Xing; CHAN, Gary. Unstructured Peer-to-Peer Network Architectures. In: BUFORD, John et al. (Org.). **Handbook of Peer-To-Peer Networking**. 1. ed. Nova Iorque: Springer, 2010. cap II, p. 117-124.

KREITZ, Gunnar; NIEMELÄ, Frederik. **Spotify -- Large Scale, Low Latency, P2P Music-on-Demand Streaming**. IEEE International Conference on Peer-to-Peer Computing, Delft, Holanda, p. 1-10, 2010. Disponível em: < https://pdfs.semanticscholar.org/26a6/a5795e5006aef31642d03e5944167f8c630e.pdf>. Acesso em: 24 maio 2017.

LIANG, Jian; NAOUMOV, Naoum, ROSS, W. Keith. **The Index Poisoning Attack in P2P File Sharing Systems**. In: IEEE International Conference on Computer Communications, 25., 2006, Barcelona, Barcelona: IEEE, p. 1-12. Disponível em: < http://www.csie.ntu.edu.tw/~azarc/poison.pdf>. Acesso em: 17 mai. 2017.

LÓPEZ-FUENTES, A. Francisco; EUGUI-DE-ALBA, Inãki; ORTÍZ-RUIZ, M. Otoniel. **Evaluating P2P Networks against Eclipse Attacks**. In: Iberoamerican Conference on Electronics and Computer Science. 2012, Guadalajara, Guadalajara: Elsevier Ltd, 2012. <http://www.sciencedirect.com/science/article/pii/S2212017312002368>. Acesso em: 01 jun. 2017.

MAUCH, G. H. et al. **Dois pesos, duas medidas: gerenciamento de identidades orientado a desafios adaptativos para contenção sybils**. In: SIMPÓSIO BRASILEIRO DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 28., 2010, Gramado. Anais... Porto Alegre: Sociedade Brasileira de Computação, 2010. p. 7-30. Disponível em: <http://www.inf.ufrgs.br/~frsantos/files/sybil_sbrc2010.pdf>.Acesso em: 17 mai. 2017.

MERILÄINEN, Markus. **Survey of DHT Evaluation Methods**. In: TKK T-110.5190 Seminar on Internetworking, 2008, Helsinki, Finlândia. Helsinki, Finlândia: TKK, 2008 Disponível em: < http://www.cse.hut.fi/en/publications/B/1/papers/Merilainen_final.pdf>. Acesso em: 17 mai. 2017.

PRÊTRE, Baptiste. **Attacks on Peer-to-Peer Networks**. 2005. p. 14-15. Tese - Department of Computer Science, Swiss Federal Institute of Technology, [S.l.], 2005. Disponível em: < http://repository.cmu.edu/cgi/viewcontent.cgi?article=1082&context=compsci >. Acesso em: 17 mai. 2017.

SINGH, Atul et. al. **Eclipse Attacks on Overlay Networks: Threats and Defenses**. In: IEEE International Conference on Computer Communications. 25., 2006, Barcelona, Barcelona: IEEE. Disponível em: < https://www.eecs.harvard.edu/~mema/courses/cs264/papers/eclipse-infocom06.pdf>. Acesso em: 17 mai. 2017

STOICA, Ion et al. **Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications**. IEEE/ACM Transactions on Networking (TON), Piscataway, EUA, v.

11, n. 1, p. 17-32, fev. 2003. Disponível em: <https://pdos.csail.mit.edu/papers/ton:chord/paper-ton.pdf>. Acesso em: 24 maio 2017.

WANG, Lin**. Attacks Against Peer-to-peer Networks and Countermeasures**. In: TKK T-110.5290 Seminar on Network Security, 2006, Helsinki, Finlândia. Helsinki, Finlândia: TKK, 2006 Disponível em: < http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.476.6803&rep=rep1&type=pdf>. Acesso em: 17 mai. 2017.

XIE, Ming. **P2P Systems Based on Distributed Hash Table**. [S.l.]: University Of Ottawa, 2003. 3 p. Disponível em: < https://pdfs.semanticscholar.org/1473/1e50e6572b0db7de5e0a0f82b7d22b22b46b.pdf> . Acesso em: 22 nov. 2016.

YANG, Beverly; GARCIA-MOLINA, Hector. **Designing a Super-Peer Network**. In: International Conference on Data Engineering, 19., 2003, Bangalore, Bangalore: IEEE. Disponível em: < http://infolab.stanford.edu/~byang/pubs/superpeer.pdf> . Acesso em: 22 nov. 2016.