Um serviço para prover autenticação e revogação de nós na rede DHT

Jean T. Garcia[1], **Lucas V. Dias**[1], **Tiago A. Rizzetti** [1]

[1]Curso Superior de Tecnologia em Redes De Computadores
– Universidade Federal de Santa Maria (UFSM)
Av. Roraima nº 1000 Cidade Universitária Bairro Camobi
Santa Maria - RS CEP: 97105-900 +55 (55) 3220-8000

jeangarcia@redes.ufsm.br, lucas_dias@redes.ufsm.br, rizzetti@redes.ufsm.br

Abstract. *This paper proposes a service that provides authentication and revocation of nodes at runtime communications on a DHT network. Authentication is provided through digital certificates coupled with public key infrastructure. Thus messages from compromised nodes will be discarded even if the node joined the network at the time it was valid. Search speed and data success rate tests were performed, considering the authentication verification and message signing before obtaining the information from the network. To implement the tests, the CORE Emulator was used. The results obtained demonstrated that the proposed service provides secure communication on the DHT network, providing performance similar to that of using the DHT network without the reactive authentication service implemented.*

Resumo. *Neste artigo é proposto um serviço que fornece autenticação e revogação de nós em tempo de execução das comunicações em uma rede DHT. A autenticação é provida através de certificados digitais com o acoplamento de uma infraestrutura de chave pública. Assim, mensagens provenientes de nós comprometidos serão descartadas mesmo que o nó tenha ingressado na rede no momento em que era válido. Foram realizados testes de velocidade de busca e taxa de sucesso de obtenção de dados, considerando a verificação de autenticação e assinatura de mensagens antes de obter a informação da rede. Para implementação dos testes utilizou-se a ferramenta CORE Emulator. Os resultados obtidos demonstraram que o serviço proposto fornece uma comunicação segura na rede DHT, fornecendo um desempenho similar ao que se verifica na utilização da rede DHT sem o serviço de autenticação reativa implementado.*

1. Introdução

As redes baseadas em *distributed hash table* (DHT) estão cada vez mais em evidência para implementação de diversos serviços [Rahimi et al. 2016]. Esta atenção especial se deve as suas características como tolerância a falhas, alta disponibilidade, escalabilidade e resiliência. Em uma rede DHT, cada nó armazena uma tabela de *hash*, nesta tabela fica armazenado um par composto por chave e valor. Qualquer nó pode recuperar ou alocar valores na DHT [Tang et al. 2008].

As propriedades da rede DHT fornecem um terreno fértil para que agentes maliciosos explorem diversas vulnerabilidades. Por exemplo, ataques *Sybil*, ataques *Eclipse* e ataques que visam o roteamento e armazenamento da rede [Srinivasan and Aldharrab 2019]. Alguns desses ataques ocorrem com sucesso pelo fato de não haver mecanismos de validação na inserção dos nós na rede.

Existe ainda o problema relativo ao comprometimento de um nó já ingressado na rede e, previamente autêntico. Neste caso não basta garantir que apenas possam ingressar na rede nós autênticos, mas que também todas as mensagens trocadas entre os nós sejam verificadas. Com isso, garante-se que um nó comprometido não possa adulterar informações e republicá-las na rede DHT.

Neste sentido, garantir uma comunicação segura entre os participantes da rede DHT torna-se um grande desafio [Ismail et al. 2016]. Assim, este trabalho concentra-se em propor e implementar um serviço reativo que fornece autenticação e revogação de nós. No que se refere a autenticação, o serviço proposto utiliza certificados digitais para verificar a identidade dos nós. As questões de emissão e revogação dos certificados digitais são tratadas pela infraestrutura de chave pública (ICP). A principal contribuição deste trabalho é oferecer uma rede segura e resiliente, mesmo na presença de nós autênticos que eventualmente sejam comprometidos. Para isso, utiliza um serviço de revogação de nós reativo, fazendo com que a cada comunicação seja verificada a validade do nó, eliminando as mensagens de nós inválidos.

O restante deste artigo está organizado da seguinte forma. Na seção, 2 é feita uma breve descrição da bibliografia elencando os trabalhos relacionados. Na seção 3, é abordado o serviço proposto. Após, na seção 4, é descrito o cenário de execução dos testes onde são apresentados os resultados obtidos e realizada uma análise contrapondo eles com o que foi obtido nos trabalhos relacionados. E por fim na seção 5, a conclusão deste trabalho elencando os pontos negativos e positivos da abordagem proposta.

2. Trabalhos Relacionados

No trabalho proposto por [Pecori 2015] é apresentado um mecanismo de confiança aplicado ao protocolo do Kademlia. Este mecanismo proposto baseia-se em uma pontuação de confiança em cada operação realizada, seja de obtenção ou alocação de dados na rede DHT. Vale ressaltar que o objetivo do trabalho é fazer a defesa contra ataques *Sybil*. Nesse tipo de ataque, o agente malicioso insere um conjunto de nós maliciosos na rede para controlar e denegrir o roteamento da rede [Pecori 2015]. Entretanto, essa abordagem é desvantajosa, visto que um nó pode ser malicioso, mas se comportar como um nó confiável e em determinado momento denegrir o comportamento da rede. Diante disso, até que este nó se torne não confiável poderá ainda se comunicar com os demais participantes.

Na arquitetura proposta por [Kohnen et al. 2011] é utilizada a autenticação baseada em certificados digitais também aplicado ao protocolo do Kademlia. O trabalho de [Kohnen et al. 2011] assume que cada nó contém um certificado assinado por uma autoridade de certificação (CA) e que cada nó confia nos demais que também possuem certificados autenticados. Em cada operação do protocolo do Kademlia é aplicada uma pontuação de confiança. Um nó decide se aceita ou não a operação de outro nó com base nessa confiança. A abordagem dos certificados digitais é interessante, visto que atribui um certificado a cada nó por uma CA. Dessa forma, um nó pode ser revogado a qual-

quer momento da rede. Entretanto, ainda há o problema de basear-se na pontuação de confiança para aceitar as mensagens.

Ambos trabalhos propõem serviços voltados ao protocolo Kademlia, sendo assim, em sequência é apresentada uma visão geral dele. Cada nó possui um identificador de 160 bits [Maymounkov and Mazieres 2002a]. Isso serve para que os nós possam ser localizados. Além disso, o protocolo utiliza de mensagens de contato entre si para rotear as mensagens de consulta [Maymounkov and Mazieres 2002a]. As quatro principais funcionalidades do protocolo de acordo com o trabalho [Maymounkov and Mazieres 2002a], são:

- *ping* que serve para verificar se o nó está online;
- *store* que serve para alocação de valores na rede DHT;
- *findnode*, responsável pela busca de um nó por meio de seu identificador;
- *findvalue* na qual serve para buscar de valores na rede DHT.

3. Serviço Proposto

O serviço proposto é constituído de duas partes, o método de autenticação e o de revogação de nós. Em primeiro lugar, a lista de revogação de certificados é periodicamente publicada na rede DHT pela CA, entretanto, a mesma não necessita comunicação em tempo real. Vale ressaltar que a ICP na qual a CA faz parte possui diversos componentes de acordo com [William Stallings 2014], como:

- autoridade registradora: Componente não obrigatório que geralmente está ligada ao registro de entidades finais;
- CA: Componente responsável pela emissão de certificados e da da lista de certificados revogados;
- Entidade Final: Nome genérico para ser identificado no campo de nome do sujeito de um certificado digital;
- Emissor de Lista de Certificados Revogados: Componente opcional que a CA delega para realizar o processo de emissão da lista de certificados revogados;
- Repositório: Termo genérico usado para denotar métodos de armazenamento de certificados e lista de certificados revogados de modo que os clientes possam recuperar.

Para ingressar na rede, um nó deve contatar outro nó já inserido, chamado de nó de *bootstrap* [Maymounkov and Mazieres 2002b]. Então deve ser realizada uma autenticação inicial onde é estabelecida uma chave de sessão com o algoritmo *diffie-hellman*. Com isso, evita-se mensagens falsificadas durante a comunicação. Também vale dizer que cada nó já ingressa com um certificado válido emitido pela CA. Este processo pode ser melhor visualizado na Figura 1.

Na etapa 1, o nó de *bootstrap* assina o pacote contendo seu certificado e envia para o nó solicitante. Em seguimento, na etapa 2, após receber o certificado, o nó confere a autenticidade do pacote. Em sequência, na etapa 3, o solicitante verifica a assinatura do certificado realizada pela CA e se o certificado não foi revogado. Se o certificado for autêntico, o nó solicitante assina seu certificado e envia-o ao nó de *bootstrap* em 4.

No passo 5, o nó de *bootstrap* verifica o certificado do solicitante. Se o certificado for autêntico, e não revogado (6), é feita a inserção do nó na rede DHT. A verificação

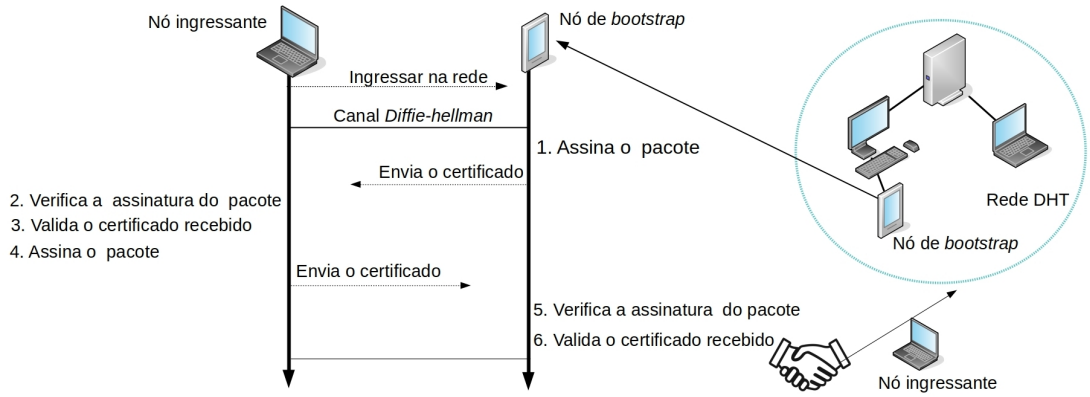

Figura 1. Modelo de autenticação inicial na rede DHT

da integridade e da autenticidade das mensagens trocadas é feita com a chave de sessão do algoritmo *diffie-hellman*. Algumas nomenclaturas importantes usadas neste artigo são apresentadas a seguir.

- *Infohash*: É uma *hash* (chave) de 160 bits gerada pelo protocolo do Kademlia.
- *Get*: Nome da operação de busca de dados da rede DHT.
- *Put*: Nome da operação para alocar uma informação na rede.

No que se refere ao serviço de revogação de nós, cada nó na rede DHT terá um identificador, sendo este uma *Infohash* do número de série do certificado. Tendo como exemplo os nós 10 e 11 da Figura 2, o segundo assinará a mensagem, publicará seu certificado e a mensagem assinada com sua chave privada na rede DHT.

Do outro lado, o nó 10 realizará a busca pela mensagem, obtendo o identificador do nó e partir disso buscará o certificado do nó na rede DHT. Se o certificado for válido (autentico e não revogado), o nó então verifica a assinatura da mensagem com a chave pública do certificado. Para operações futuras o certificado é armazenado localmente.

4. Cenário de Testes

O cenário criado na ferramenta de simulação de redes *Common Open Research Emulator* (CORE) Emulator [Ahrenholz 2010] contém 50 nós. Essa é a quantidade máxima de nós em que a simulação se comportou de maneira adequada na máquina hospedeira. Essa última possui as características computacionais apresentadas abaixo.

- Processador: Intel® Core® I3;
- Memória RAM: 8 GigaBytes;
- Sistema operacional: Linux mint 19.1 64 bits;
- Versão do CORE *Emulator*: 4.8.

Para implementar o serviço foi desenvolvida uma aplicação na linguagem C++ com a biblioteca OpenDHT. Essa é baseada no Kademlia e fornece um armazenamento de dados na memória distribuído fácil de usar [Savoir-faire Linux Inc 2019]. Com essa biblioteca foi possível colocar a rede DHT em funcionamento.

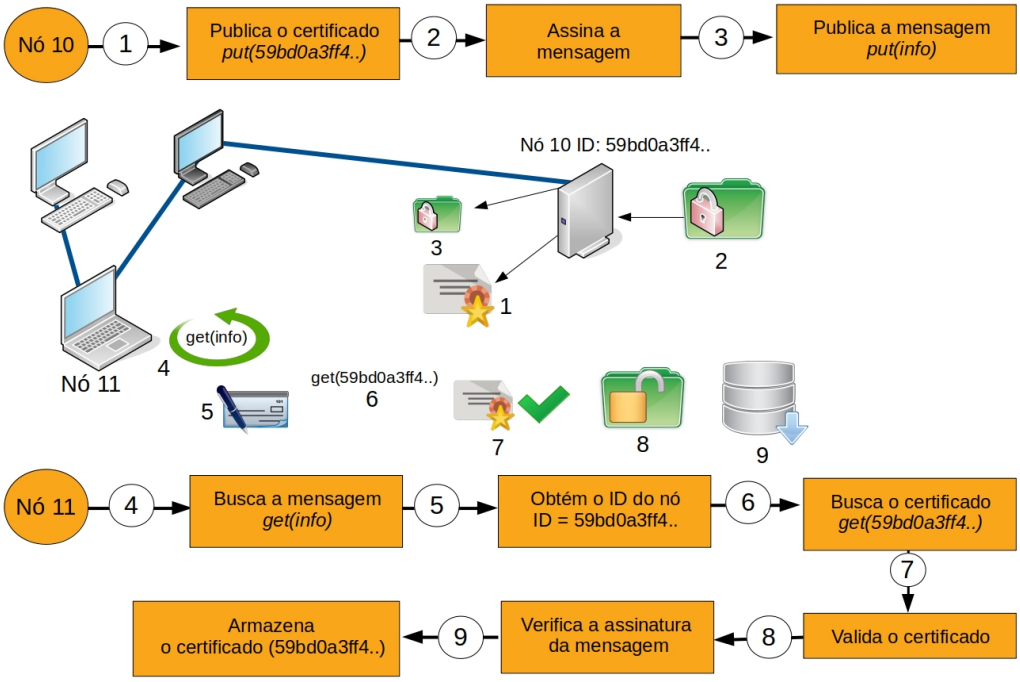

Figura 2. Exemplo do serviço proposto na rede DHT

No que se refere ao processo de teste, foram executados cinco rodadas de testes em cada configuração. As *Infohash*s publicadas pelos nós foram geradas pseudoaleatoriamente dentro do espaço total de nós (50 nós). Isso significa que os nós escolheram a *Infohash* entre (1 - 50). Para garantir a persistência das informações cada nó republicou a informação em um tempo pseudoaleatório no intervalo entre 1 e 10 segundos. Esse tempo foi escolhido com o objetivo de não permitir que um nó inunde a rede com suas *Infohash*s.

Vale ressaltar que o formato de certificados utilizados foi o X509 na versão 3. Dessa forma, os campos do certificado são versão, número de série do certificado, parâmetros de assinatura da CA, período de validade, informações sobre a chave pública do sujeito, identificador único do emissor, do sujeito, as extensões e a assinatura da CA [William Stallings 2014].

Os certificados revogados pela CA também foram gerados de forma pseudoaleatória. Isso fez com que cada nó tivesse a mesma probabilidade de ser revogado. No teste de *get* foi gerada uma lista de revogação com 20% de nós não autênticos.

Ainda assim, a taxa de sucesso de *get*, por exemplo com 50 nós e 5000 *Infohash*s publicadas foi de 79,4%. Isto significa que o restante (20.6%) das informações foram descartadas, visto que não eram válidas. Isto comprova a eficácia da abordagem reativa, em que a autenticação é verificada em tempo de execução das comunicações na rede. Em alguns casos a porcentagem de não sucesso de *get* foi maior que o número de nós maliciosos na rede. Isso se deve ao fato de que os nós publicavam a informação mais de uma vez.

O gráfico geral pode ser visualizado na Figura 3. A não linearidade no resul-

tado se dá pelo fato de que o intervalo de valores de identificação das *Infohash*s é muito próximo. Com isso, por muitas vezes ela foi armazenada no mesmo nó. A rede DHT tem uma relação entre *Infohash* e identificador dos nós. Ou seja, em alguns momentos, as informações foram armazenadas em um dispositivo autêntico e em outros momentos em um nó inválido. Com o aumento de *Infohash*s, a probabilidade da escolha de um dispositivo inválido diminui.

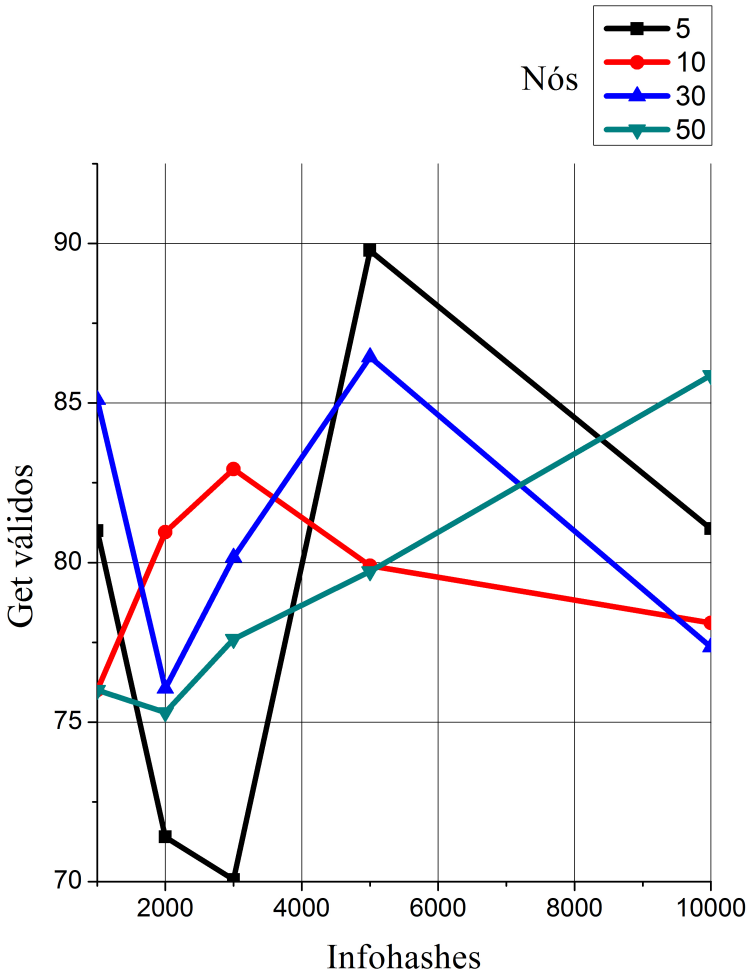

Figura 3. Resultados de operações *get* válidas com 20% de nós maliciosos.

Com o intuito de avaliar o desempenho do serviço proposto em relação à rede DHT em sua versão original, sem a validação reativa, foi realizado um teste de velocidade de busca das *Infohash*s com o serviço proposto implementado. O gráfico da Figura 4 demonstra que o serviço não degradou de forma significativa o desempenho da rede DHT. Com uma quantidade menor de nós na rede, o tempo para recuperar informações é proporcional. Isso se dá pelo fato da rede DHT estar em uma topologia plana e os nós precisarem trocar informações para montar a tabela de roteamento [Tang et al. 2008].

Comparando os resultados na Figura 4 com 30 nós utilizando o serviço proposto e não utilizando, pode ser visto a tendência do tempo de busca ser o mesmo. Isso também ocorre quando comparado os gráficos de 50 nós. Esse resultado reforça a utilização da

solução proposta, demonstrando que não degrada de forma significativa o desempenho da rede de maneira significativa.

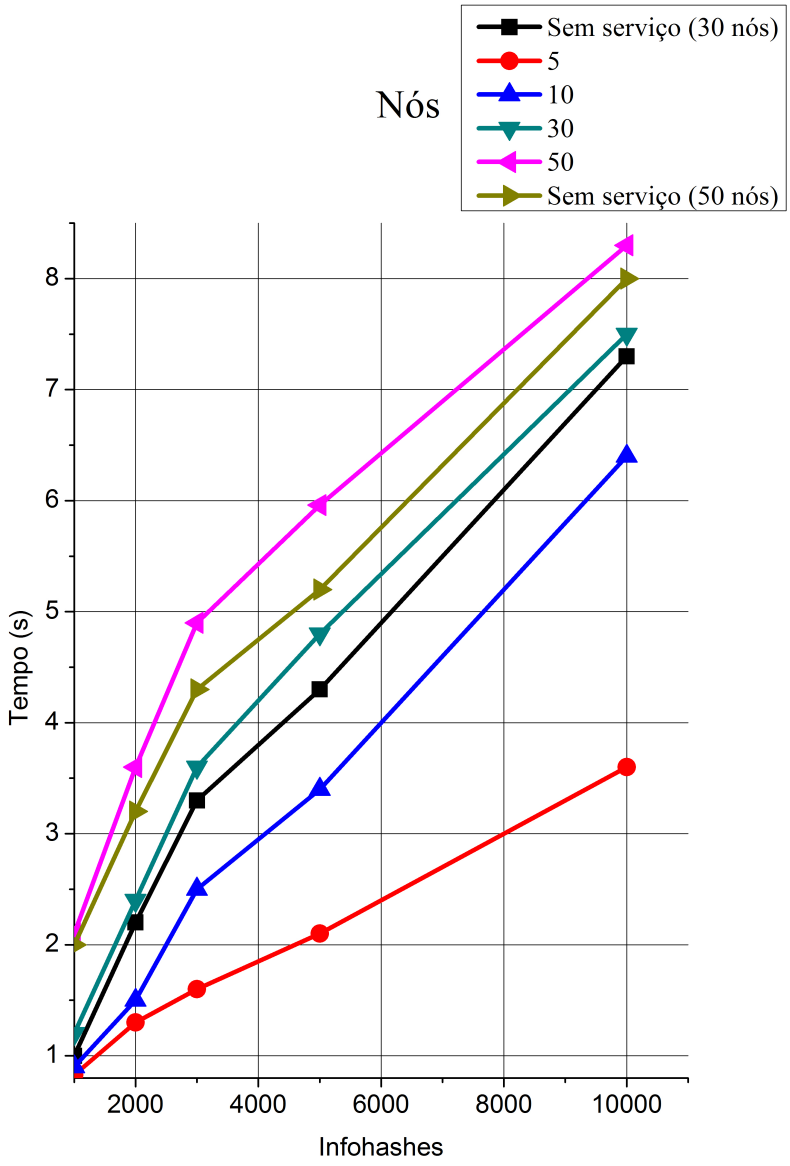

Figura 4. Resultados do teste de velocidade de busca das *Infohash*s

4.1. Análise com os trabalhos relacionados

No que se refere ao trabalho de [Pecori 2015], o resultado de operações de (*get*) bem sucedidas é cerca de 80%, com nenhum nó falso na rede. Isso significa que as comunicações estão sendo ignoradas pela relação de confiança abordada.

No que se refere a proposta de [Kohnen et al. 2011], considerando cerca de 40% de nós maliciosos, foi obtida uma taxa de sucesso de *get* de cerca de 70%. Porém, novamente a questão da confiança leva aos participantes considerarem nós maliciosos como confiáveis.

A abordagem proposta no presente artigo trás contribuições neste sentido. Mesmo com a presença de uma significativa porcentagem de nós falsos tentando se comunicar na rede DHT. O sucesso de busca de uma informação na rede DHT foi significativo conforme apresentado na Figura 3. Salienta-se também o fato que a abordagem não impactou significativamente o desempenho da sobreposição com o serviço proposto.

5. Conclusão

Atualmente (2019), a rede do Mainline é um bom exemplo do uso das vantagens da rede DHT baseadas no Kademlia implementadas na Internet com milhões de usuários ativos [Xinxing et al. 2016]. Nesse sentido, a segurança da informação em especial a prerrogativa de autenticação tem sido um grande desafio [Shin et al. 2019].

Diante disso, o respectivo artigo trouxe um serviço para prover autenticação e revogação de nós em uma rede DHT. Foi realizada a implementação de uma aplicação na linguagem C++ com a biblioteca OpenDHT para validar o serviço proposto. Para a validação do serviço foi utilizado meios computacionais simulando um ambiente de rede com o CORE Emulator. Dessa forma, os resultados obtidos demonstraram que o mecanismo de revogação e autenticação é vantajoso para garantir esses aspectos de segurança, sem degradar o desempenho de modo significativo.

Por fim, vale ressaltar que o trabalho apresenta um método de revogação e autenticação dos nós, a abordagem é voltada as mensagens na rede. Entretanto, há questões como proteção a ataques de negação de serviço que podem serem explorados como trabalhos futuros. Com o intuito de melhorar o desempenho da arquitetura proposta, como trabalho futuro fica a implementação em uma rede DHT hierárquica na qual faz segmentação da rede. A DHT plana tem a desvantagem de escalabilidade reduzida, já que todos os nós ficam no mesmo nível. Isso acarreta no crescimento da tabela de roteamento dos nós, impossibilitando sua segmentação. [Tang et al. 2008].

Referências

Ahrenholz, J. (2010). Comparison of core network emulation platforms. In *2010-Milcom 2010 Military Communications Conference*, pages 166–171. IEEE.

Ismail, H., Germanus, D., and Suri, N. (2016). Malicious peers eviction for p2p overlays. In *2016 IEEE Conference on Communications and Network Security (CNS)*, pages 216–224.

Kohnen, M., Gerbecks, J., and Rathgeb, E. P. (2011). Applying certificate-based routing to a kademlia-based distributed hash table.

Maymounkov, P. and Mazieres, D. (2002a). Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer.

Maymounkov, P. and Mazieres, D. (2002b). Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer.

Pecori, R. (2015). Trust-based storage in a kademlia network infected by sybils. pages 1–5.

Rahimi, N., Sinha, K., Gupta, B., Rahimi, S., and Debnath, N. C. (2016). Ldepth: A low diameter hierarchical p2p network architecture. pages 832–837.

Savoir-faire Linux Inc (2019). Opendht. Acesso em 20 jul. 2019.

Shin, J., Islam, M. R., Rahim, M. A., and Mun, H.-J. (2019). Arm movement activity based user authentication in p2p systems. *Peer-to-Peer Networking and Applications*, pages 1–12.

Srinivasan, A. and Aldharrab, H. (2019). Xtra—extended bit-torrent protocol for authenticated covert peer communication. *Peer-to-Peer Networking and Applications*, 12(1):143–157.

Tang, X., Xu, J., and Lee, W. (2008). Analysis of ttl-based consistency in unstructured peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(12):1683–1694.

William Stallings, L. B. (2014). *Segurança de Computadores: Princípios e Práticas*. Rio de Janeiro.

Xinxing, Z., Zhihong, T., and Luchen, Z. (2016). A measurement study on mainline dht and magnet link. In *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*, pages 11–19.