



XXVIII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2024

Integração da Tecnologia Tangle/IOTA ao Ambiente de Emulação Fogbed

Weslei Santos Pinheiro¹; Antônio Augusto Teixeira Ribeiro Coutinho²

1. Estagiário PEVIC/UEFS, Graduando em Engenharia da Computação, Universidade Estadual de Feira de Santana,
e-mail:weslei.eng.comp@gmail.com
2. Orientador, Departamento de tecnologia (DTEC), Universidade Estadual de Feira de Santana,
e-mail: augusto@ecomp.ufes.br

PALAVRAS-CHAVE: Emulação; Névoa; Borda; Blockchain; Livro-razão distribuído.

INTRODUÇÃO

A Internet das Coisas (*Internet of Things*, IoT) (Aazam et al., 2014) continua a se expandir por meio da rede global de computadores, com a expectativa de que até 2025 mais de 30 bilhões de dispositivos inteligentes estarão conectados através da Internet (Lueth, 2021). Este crescimento vem gerando desafios envolvendo o gerenciamento e a segurança dos dados produzidos pelos dispositivos IoT (Aazam et al., 2014). Soluções baseadas em Computação em Borda/Névoa (Coutinho et al., 2018), agregadas às tecnologias de livro-razão distribuído (*Distributed Ledger Technologies*, DLT), como o Blockchain (Baniata & Kertesz, 2020), podem oferecer baixa latência e promover a segurança em ambientes voltados à IoT (Hassija et al., 2019). Este trabalho envolve a integração da DLT Tangle/IOTA¹ ao Fogbed (Coutinho et al., 2018), um emulador de Borda/Névoa, visando à prototipagem rápida de sistemas de Borda/Névoa-DLT. A solução empregou uma metodologia baseada em *plugins* de software para facilitar a integração da DLT Tangle/IOTA ao ambiente Fogbed. Para demonstrar a solução, foi realizada uma avaliação de desempenho de uma rede Tangle/IOTA emulada sobre o ambiente no Fogbed, focando no tempo médio necessário para escrever/ler transações usando os métodos HTTP/RESTful e MQTT/ZMQ.

MATERIAL E MÉTODOS OU METODOLOGIA (ou equivalente)

A integração e avaliação da DLT Tangle/IOTA sobre o ambiente Fogbed requereu a execução de um conjunto de etapas fundamentais, listadas a seguir:

1. **Implementação do plugin FogLedger/IOTA** - Para a criação de um *plugin* FogLedger para rede Tangle/IOTA foram inseridos os seguintes componentes:
 - Imagens de contêineres Docker de nós *Hornet*² (v1.2.4) da rede Tangle/IOTA disponibilizadas pelo projeto IOTA;
 - Um *script* em Bash de configuração a rede Tangle/IOTA;

¹ <http://www.iota.org>

² <https://wiki.iota.org/hornet/welcome/>

- Uma classe em Python chamada *IotaBasic*, seguindo boas práticas de desenvolvimento de software, além de outras classes estendidas para lidar com configurações específicas da rede.

2. Implementação da Rede IOTA – A Figura 1 ilustra os componentes da rede privada Tangle/IOTA emulada através do *plugin* FogLedger. Este processo envolve a sequência de passos específicos listados de forma resumida a seguir:

- Configuração dos nós da rede:** A rede Tangle/IOTA é mantida pelos nós *Coordinator* e *Spammer*, enquanto os nós *Hornet* utilizam os arquivos *config.json* e *peering.json*, que são criados e atualizados pela classe *IotaBasic*, para a configuração inicial dos nós participantes da rede;
- Criação das identidades par-a-par (peer-to-peer, P2P):** Identidades criptográficas são geradas para cada nó na rede e registradas no arquivo *peering.json* de cada nó, possibilitando a identificação e a comunicação entre os nós através da rede Tangle/IOTA;
- Bootstrap do coordenador:** Este processo de inicialização (*bootstrap*) envolve a obtenção do *snapshot*, a validação dele, a sincronização com a rede Tangle/IOTA e a subsequente participação na rede;
- Execução dos nós:** Após a configuração inicial, os nós são executados na rede Tangle/IOTA, conforme a arquitetura apresentada na Figura 1.

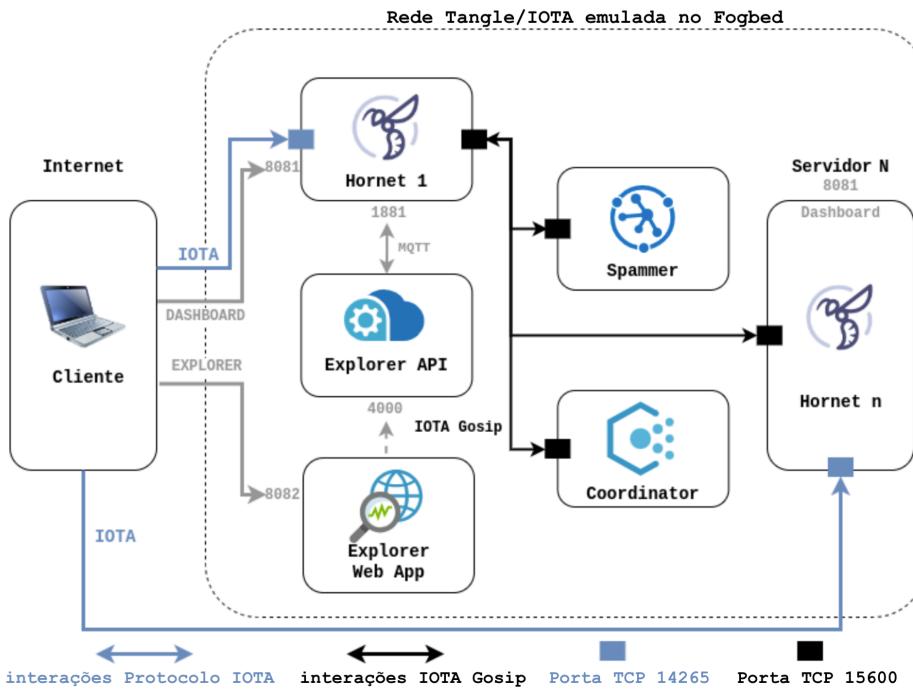


Figura 1: Arquitetura mínima da rede IOTA/Tangle emulada através do plugin FogLedger. Utilizam-se protocolos e portas distintas na troca de mensagens: entre clientes (IOTA) e entre os nós da rede (Gossip). Os componentes Explorer API e Web App permitem a visualização de transações em uma interface web.

Implementação do Experimento - Para demonstrar a solução, foram construídos componentes agregados visando a execução de transações de leitura/escrita na rede Tangle/IOTA através de dois métodos distintos: HTTP/RESTful e MQTT/ZMQ.

- **Bridge MQTT/ZMQ:** Um servidor ZMQ no modelo *publisher/subscriber* foi implementado em Golang como uma ponte (*bridge*) entre os clientes e o nó da rede Tangle/IOTA. Este servidor comunica-se diretamente com o nó da rede pela

porta MQTT (1883), permitindo que clientes assinantes recebam mensagens específicas através do índice passado como parâmetro pela interface ZMQ. Assim, o servidor ZMQ atua como um filtro para as mensagens indexadas da rede Tangle, facilitando sua recuperação de forma similar ao método HTTP/RESTful.

- **Clientes da rede:** Os clientes da rede Tangle/IOTA foram desenvolvidos em Golang usando a biblioteca *iota.go*. Eles executam o envio e o recebimento de mensagens por meio de linhas de execução concorrentes, utilizando *goroutines*. Foram implementados dois tipos de clientes: um HTTP para escrita e leitura e um ZMQ para leitura. O cliente HTTP de escrita realiza o envio de mensagens à Tangle, anexando o tempo de criação das mensagens à área de dados (*payload*). O cliente HTTP de leitura executa um loop infinito em busca de mensagens por índices específicos, enquanto o cliente ZMQ subscreve a um tópico (índice) e aguarda as respostas do servidor MQTT/ZMQ. Ambos armazenam as mensagens em canais (buffers) e calculam os tempos de envio e recepção, permitindo uma comparação justa na busca e recuperação de mensagens na rede Tangle/IOTA.

RESULTADOS E/OU DISCUSSÃO (ou Análise e discussão dos resultados)

O experimento comparou a comunicação com a rede Tangle/IOTA utilizando os métodos HTTP/RESTful e MQTT/ZMQ. Os clientes HTTP de leitura buscaram transações de um índice específico em paralelo com a estratégia utilizada pelos clientes ZMQ. Os testes foram realizados em 11 máquinas Intel Core i3 de 8^a geração, equipadas com 12GB de memória RAM e sistema operacional Linux Ubuntu 20.04 sobre a rede do LARSID no LABOTECIII/DTEC. Dez máquinas foram dedicadas aos clientes, e uma foi designada para a rede Tangle/IOTA. Durante o experimento, o número de clientes HTTP de escrita foi variado de maneira balanceada em cada máquina, enquanto os tempos de resposta foram registrados pelos clientes HTTP e ZMQ de leitura em arquivos CSV. A Tabela 1 mostra os tempos médios de resposta para o envio e recebimento de mensagens na rede Tangle/IOTA para grupos de 10, 100 e 200 clientes HTTP de escrita, incluindo o desvio padrão e o intervalo de confiança, calculado com 95% de confiabilidade.

Tabela 1. Resultado da análise do tempo de resposta.

Cliente	Número de Clientes	Média (s)	Desvio padrão (s)	Intervalo de confiança (s)
HTTP	10	2.8922	1.9849	2.7748 - 3.0096
	100	4.1142	2.7121	3.9606 - 4.2678
	200	5.5313	2.8094	5.3570 - 5.7056
ZMQ	10	2.1466	2.1481	2.0196 - 2.2737
	100	3.9512	3.4851	3.7538 - 4.1486
	200	5.1270	3.8084	4.3713 - 5.8827

Os dados analisados revelam que o aumento do número de clientes impacta diretamente os tempos de resposta, indicando uma correlação entre a carga da rede e o desempenho. Em comparação com o HTTP, o ZMQ apresenta desvio padrão menor e intervalos de confiança mais estreitos, sugerindo uma maior consistência nos resultados e proporcionando uma visão mais precisa do desempenho médio sobre diferentes cenários de carga. Como ilustrado na Figura 2, o ZMQ oferece tempos médios de resposta inferiores, ressaltando sua eficiência relativa ao método HTTP.

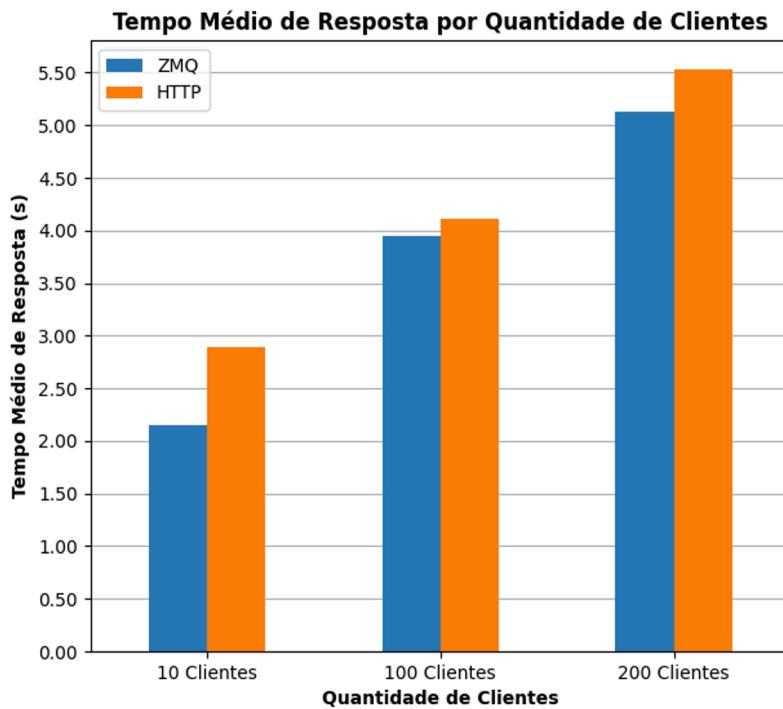


Figura 2: Gráfico demonstrando tempo médio de resposta por quantidade de clientes.

Em resumo, a integração do ZMQ demonstrou vantagens significativas em eficiência quando comparada ao método HTTP, comprovando a viabilidade do experimento na avaliação da rede Tangle/IOTA usando o ambiente Fogbed. Os resultados enfatizam a superioridade do ZMQ em termos de desempenho, sem comprometer a eficácia da rede. Trabalhos futuros podem focar em explorar novas integrações e metodologias de teste para ampliar a capacidade de emulação de redes DLT dentro do ambiente Fogbed.

REFERÊNCIAS

- AAZAM, M.; KHAN, I.; ALSAFFAR, A.; HUH, E. 2014. Cloud of things: Integrating internet of things and cloud computing and the issues involved. In: Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST). IEEE. p. 414-419.
- BANIATA, H.; KERTESZ, A. 2020. A survey on blockchain-fog integration approaches. IEEE Access, vol. 8. p. 102657-102668.
- BONOMI, F.; MILITO, R.; NATARAJAN, P.; ZHU, J. 2014. Fog computing: A platform for internet of things and analytics. In: Big Data and Internet of Things: A Roadmap for Smart Environments. Springer. p. 169-186.
- COUTINHO, A.; GREVE, F.; PRAZERES, C.; CARDOSO, J. 2018. Fogbed: A rapid prototyping emulation environment for fog computing. In: 2018 IEEE International Conference on Communications (ICC), p 1-7.
- HASSIJA, V.; CHAMOLA, V.; SAXENA, V.; JAIN, D.; GOYAL, P.; SIKDAR, B. A. 2019. A Survey on iot security: Application areas, security threats, and solution architectures. IEEE Access, v. 7, p. 82721-82743.
- LUETH, L. 2021. State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time. Disponível em: <https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time>. Acessado em: 9 de março de 2023.