

Emulação de uma WSN WiFi para Monitoramento Acústico de Mosquitos

Yang da Fontoura Rodrigues¹, Rodrigo Brandão Mansilha¹

¹AI Horizon Labs - Universidade Federal do Pampa (UNIPAMPA)

{yangrodrigues.aluno, rodrigomansilha}@unipampa.edu.br

Abstract. *This work presents an emulation experiment, on Mininet-WiFi, of an IoT wireless sensor network (WSN) for acoustic mosquito monitoring, with edge audio classification and an architecture centralized on a single server that aggregates data from multiple sensor clusters. We evaluated the MQTT, HTTP, and CoAP application protocols under increasing loads (10 to 50 sensors). MQTT presented the lowest latency at all loads, while throughput and packet loss were equivalent across protocols. We conclude that MQTT is the most suitable protocol for this scenario, favored by the centralized architecture adopted.*

Resumo. *Este trabalho apresenta um experimento de emulação, sobre Mininet-WiFi, de uma rede de sensores sem fio (WSN) IoT para monitoramento acústico de mosquitos, com classificação de áudio na borda e uma arquitetura centralizada em um único servidor que agrega os dados de múltiplos clusters de sensores. Avaliamos os protocolos de aplicação MQTT, HTTP e CoAP sob cargas crescentes (10 a 50 sensores). O MQTT apresentou a menor latência em todas as cargas, enquanto vazão e perda de pacotes foram equivalentes entre os protocolos. Concluímos que o MQTT é o protocolo mais adequado a esse cenário, favorecido pela arquitetura centralizada adotada.*

1. Introdução

O monitoramento acústico de mosquitos insere-se no campo mais amplo do monitoramento inteligente de mosquitos, que abrange sistemas IoT, armadilhas inteligentes, classificação por aprendizado de máquina e monitoramento de variáveis ambientais [Siddiqui and Kayte 2022]. Sua relevância decorre da aplicação direta em saúde pública: ao identificar regiões com maior incidência mosquitos vetores de doenças, torna-se possível direcionar estrategicamente soluções avançadas, como por exemplo, dedetização e espalhamento de mosquitos geneticamente modificados. Nesse contexto, sistemas distribuídos IoT de monitoramento acústico realizam o sensoriamento, a classificação e a detecção de mosquitos em escala a partir da frequência do batimento das suas asas.

Diante desse cenário, o objetivo deste trabalho é avaliar, em ambiente emulado, através do software Mininet-WiFi, um sistema distribuído IoT de monitoramento acústico de mosquitos com classificação na borda. Para isso, construímos um cenário de emulação com arquitetura centralizada, na qual um único servidor concentra os dados enviados por múltiplos clusters de sensores conectados via pontos de acesso WiFi, de forma análoga a uma infraestrutura urbana. No experimento, avaliamos o impacto no desempenho que a rede sofre com a escolha do protocolo (MQTT, HTTP e CoAP) e a quantidade de sensores ativos. Analisamos métricas de rede (latência, vazão e perda de pacotes) e do modelo de

classificação (acurácia de classificação), permitindo analisar os trade-offs entre protocolos e subsidiar a escolha arquitetural para implantações reais.

2. Fundamentação Teórica

O monitoramento acústico de mosquitos baseia-se na premissa de que o batimento de asas de cada espécie possui uma frequência sonora característica [Mukundarajan et al. 2017]. A partir da captura do áudio, a classificação pode ser realizada através de métodos baseados em regras – comparando a frequência fundamental com faixas pré-definidas, ou através de métodos de aprendizado de máquina, com destaque para redes neurais convolucionais (CNNs) residuais, como proposto em [Paim et al. 2024].

Sistemas distribuídos de monitoramento de mosquitos empregam nós sensores equipados com microfones, microcontroladores e módulos de rádio. Tipicamente, o processamento ocorre na borda: cada nó executa localmente o pré-processamento, a extração de características e a classificação, transmitindo à infraestrutura de backend apenas o resultado da inferência e dados ambientais. Essa organização permite uma carga de tráfego reduzida na rede, mas mantém restrições ao hardware embarcado – que precisa acomodar modelos de classificação leves – tornando o desempenho sensível à capacidade de processamento local de cada nó.

A comunicação entre os nós sensores e o backend depende, por sua vez, do protocolo de aplicação escolhido, já que MQTT, HTTP e CoAP representam paradigmas distintos, com implicações diferentes para o desempenho de sistemas IoT. O MQTT opera no modelo publish/subscribe: dispositivos publicam mensagens em tópicos gerenciados por um broker central, e clientes assinam esses tópicos. Essa arquitetura desacopla produtores e consumidores, sendo naturalmente escalável para muitos dispositivos sensores e clientes assinantes [Banks et al. 2019, p. 1-2]. HTTP e CoAP, por outro lado, seguem o modelo requisição-resposta, no qual o cliente envia uma requisição e aguarda a resposta do servidor. O HTTP, apesar da simplicidade e ampla compatibilidade, apresenta maior sobrecarga de cabeçalhos e não mantém conexão persistente por padrão. Já o CoAP opera sobre UDP, com cabeçalho significativamente menor e mecanismos como retransmissão opcional para lidar com a natureza não confiável do protocolo de transporte [Shelby et al. 2014].

3. Trabalhos Relacionados

No projeto Aedes Vigilance, o MosqMon [Rodrigues et al. 2024] propõe um sistema agente/gerente para coleta de áudio em armadilhas, validado funcionalmente com um protótipo de um agente e um gerente. Além disso, diversos trabalhos empregam o Mininet-WiFi para emular redes de sensores sem fio: [Buzura et al. 2022] propuseram um framework para redes 6LoWPAN, demonstrando escalabilidade em um cenário de cidade inteligente, enquanto [Souza et al. 2025] estenderam o Mininet-WPAN com telemetria em rede para emular WSNs de baixa potência de forma programável. Em paralelo, há uma ampla discussão sobre o desempenho de protocolos de aplicação em IoT: [Larmo et al. 2019] mostraram que CoAP/UDP supera o MQTT/TCP em latência, cobertura e capacidade sobre NB-IoT, exceto sob baixa carga; [Seoane et al. 2021] constataram que o MQTT consome mais banda e CPU que o CoAP, ainda que este ofereça apenas mecanismos simples de confiabilidade; e [Gemirter et al. 2021] verificaram, com dados reais

de tráfego urbano, que MQTT e AMQP são cerca de quatro vezes mais rápidos que HTTP em latência, enquanto o HTTP consome cerca de quatro vezes mais CPU. A Tabela 1 sintetiza esses trabalhos e posiciona este artigo em relação a eles.

Tabela 1. Posicionamento dos trabalhos.

Trabalho	Contribuição	Cenário	Plataforma
[Rodrigues et al. 2024]	Arquitetura para coleta de áudio de mosquitos	Monitoramento de mosquitos	MosqMon
[Buzura et al. 2022]	Framework de simulação para redes 6LoWPAN	Cidades inteligentes	Mininet-WiFi
[Souza et al. 2025]	Plataforma de emulação para IoT programáveis	WSNs de baixa potência	Mininet-WPAN
[Larmo et al. 2019]	Ava. de desempenho MQTT e CoAP	Redes celulares NB-IoT	Sim. NB-IoT
[Seoane et al. 2021]	Ava. de desempenho MQTT e CoAP	Segurança de redes IoT	Rasp. Pi e NetEm
[Gemirter et al. 2021]	Ava. de desempenho MQTT, AMQP e HTTP	Cidades inteligentes	Azure IoT Hub
Este trabalho	Ava. de desempenho MQTT, HTTP e CoAP	Monitoramento de mosquitos	Mininet-WiFi

Como sintetizado na Tabela 1, este trabalho se diferencia por integrar o cenário de monitoramento de mosquitos à emulação com Mininet-WiFi para avaliar protocolos de aplicação. Assim, as contribuições deste trabalho são: (i) um experimento de emulação em um cenário de monitoramento acústico de mosquitos, com inferência na borda e uma topologia WiFi com múltiplos clusters e consumo centralizado; e (ii) uma comparação de MQTT, HTTP e CoAP nesse experimento, em latência, vazão e perda de pacotes.

4. Metodologia

O experimento consiste em uma emulação de uma WSN construída sobre Mininet-WiFi¹ composta por quatro elementos: nós sensores, que representam dispositivos IoT com capacidade de classificação local; pontos de acesso, que concentram o tráfego dos sensores e o encaminham ao backbone; um servidor central, que recebe os metadados de classificação; e nós cientistas, que representam pesquisadores consumindo esses dados. Um diagrama da arquitetura da emulação proposta é apresentada na figura 1.

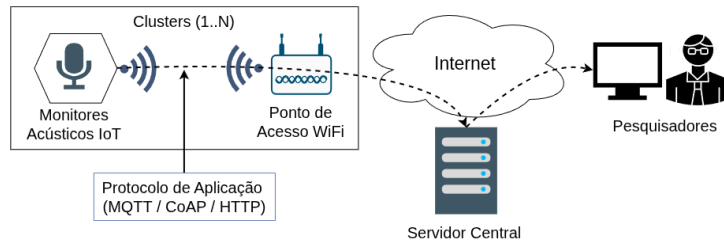


Figura 1. Visão Geral da Arquitetura.

O Mininet-WiFi [Fontes et al. 2015] é um emulador de redes que estende o Mininet para adicionar suporte a emulação de redes WiFi: estações e pontos de acesso virtualizados e modelos de propagação de rádio como o Log-Normal Shadowing. Isso permite reproduzir condições realistas de atenuação de sinal, interferência e posicionamento dos dispositivos, o tornando adequado para avaliar arquiteturas de monitoramento IoT sob diferentes condições.

O classificador é uma CNN residual [Paim et al. 2025] com 4 blocos residuais e 93.116 parâmetros, que recebe espectrogramas mel de janelas de 1,25 s de áudio a 8

¹O código-fonte do experimento, incluindo os cenários de teste e os dados coletados, está disponível em <https://github.com/Y4ngfr/experimento-mosquitos>. O dataset e o modelo de classificação utilizados estão disponíveis em <https://www.inf.ufrgs.br/aedes-vigilance>.

kHz. O dataset possui 2.706 arquivos em quatro classes: macho, fêmea, outros mosquitos e ambiente. Medimos a latência fim-a-fim (do sensor ao servidor, pelo relógio do host), a vazão no servidor, a perda de pacotes e a acurácia do classificador, calculada sobre os metadados recebidos.

Configuração da Emulação. A rede opera em topologia estrela com clusters de sensores associados a pontos de acesso, distribuídos de forma a representar uma implantação urbana com múltiplas armadilhas inteligentes conectadas a um servidor central. A tabela 2 consolida os parâmetros fixos da emulação e a tabela 3 especifica a configuração do ambiente de execução.

Tabela 2. Parâmetros fixos.

Parâmetro	Valor
Duração / repetições	300 s / 3
Carga	10, 20, 30, 40, 50
Endereçamento	IPv4
Tecnologia sem fio	WiFi IEEE 802.11g
Frequência	2,4 GHz
Modelo de propagação	Log-Normal Shadowing
Expoente de perda	2,0
Topologia	Estrela (clusters)
Sensores por ponto de acesso	10
Distância sensor-AP	30–60 m
Distância entre clusters	~150 m
Cenário / relevo	Ar livre / plano

Tabela 3. Ambiente de execução.

Item	Especificação
CPU	AMD Ryzen 5 5500U (6 núcleos, 12 threads)
RAM	5,6 GiB
Sistema operacional	Pop!_OS 22.04 LTS (kernel 6.12.10)
Emulador	Mininet-WiFi 2.7
Broker MQTT	mosquitto 2.0.21
Biblioteca CoAP	aiocoap 0.4.7
Python	3.9.25

A confiabilidade de entrega foi equalizada entre os protocolos: no MQTT adotou-se QoS 1 (*at least once*); no CoAP, o modo confirmável (CON), no qual cada POST exige acknowledge do servidor; e no HTTP a garantia é provida nativamente pela camada de transporte (TCP).

Cada sensor transmite um metadado de aproximadamente 250 bytes (JSON com classe predita, confiança, identificador do sensor, cluster e número de sequência) a cada ciclo de sensoriamento – leitura de ~10 s de áudio real seguida de classificação na borda – resultando em uma mensagem a cada 10 s por sensor, com tráfego agregado crescendo linearmente com a carga da rede.

5. Resultados

A Figura 2 apresenta a latência fim-a-fim média e vazão no servidor para os três protocolos nas cargas de 10 a 50 sensores com desvio padrão entre as repetições (barras de erro).

Os resultados mostram que o MQTT apresentou a menor latência em todas as cargas (1,4 ms – 42,1 ms). Com 10 e 20 sensores, os protocolos diferenciam-se em poucos milissegundos, porém a partir de 30 sensores o aumento da latência para o HTTP e CoAP é mais acentuado do que o MQTT. Com 50 sensores, o HTTP chega a 219,8 ms e o CoAP a 187,8 ms, cerca de 5,2 e 4,5 vezes a latência do MQTT.

O desempenho superior do MQTT reflete a adequação do seu modelo publish/subscribe à arquitetura centralizada avaliada, em que um único servidor consome dados de muitos sensores de forma assíncrona. Entre os protocolos requisição-resposta, o CoAP – mais leve e voltado a dispositivos IoT – obteve menor latência a partir de 30 sensores.

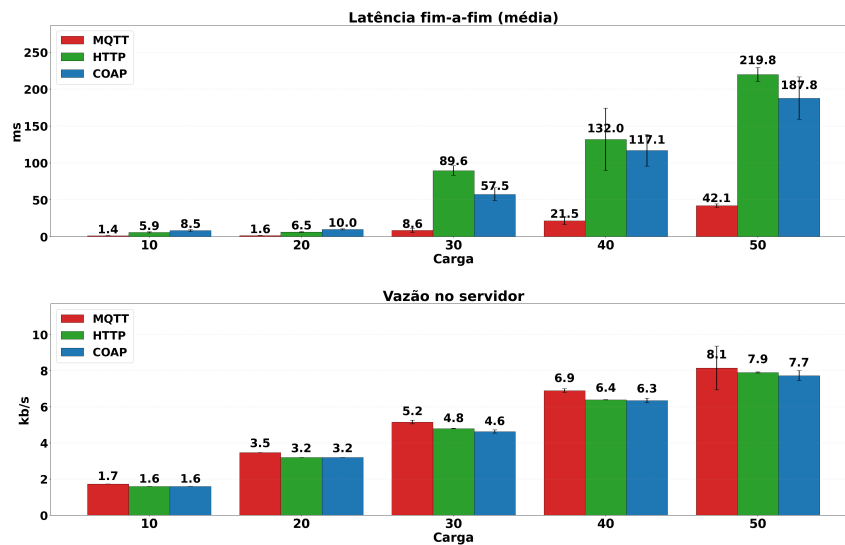


Figura 2. Latência média e vazão acumulada no servido por protocolo e carga (média de 3 repetições; barras: desvio-padrão)

A vazão no servidor cresce de forma aproximadamente linear com a carga nos três protocolos (MQTT: 1,7–8,1 kb/s; HTTP: 1,6–7,9 kb/s; CoAP: 1,6–7,7 kb/s) e não houve perda de pacotes. Assim, a capacidade de transferência não limita o sistema: a diferença entre os protocolos aparece na latência.

A acurácia da classificação na borda manteve-se em aproximadamente 81% em todos os cenários, como esperado: a inferência ocorre no sensor, antes da transmissão, e o protocolo afeta apenas a entrega do metadado. Ela só dependeria do protocolo se a classificação fosse feita no servidor, com envio do áudio pela rede.

6. Considerações Finais

Este trabalho apresentou uma emulação de um sistema distribuído IoT de monitoramento acústico de mosquitos sobre o Mininet-WiFi. Os resultados indicam que o MQTT é o protocolo mais adequado ao contexto avaliado, dada sua aderência ao modelo de servidor centralizado com múltiplos consumidores. Com 50 sensores, HTTP e CoAP apresentaram latências semelhantes entre si e de 4,5 a 5,2 vezes maiores que a do MQTT. Isso sugere que a escolha do protocolo de aplicação deve considerar tanto a arquitetura de rede da implantação quanto sua eficiência em cenários IoT.

Como trabalhos futuros, pretende-se estender a avaliação para diferentes cenários de rede, como maior densidade de sensores, outras tecnologias de comunicação sem fio (e.g., LoRa e IEEE 802.15.4) e diferentes topologias de rede.

Agradecimentos. Apoio do CNPq (PIBIT), FAPERGS (PqG processo 25/2551-0002572-9) e FAPESP (processos 2020/05183-0 and 2023/00816-2).

Referências

Banks, A., Briggs, E., Borgendale, K., and Gupta, R. (2019). *MQTT Version 5.0*. OASIS Standard. Especificação do protocolo MQTT versão 5.0.

- Buzura, S., Dadarlat, V., Peculea, A., Bertrand, H., and Chevalier, R. (2022). Simulation framework for 6lowpan networks using mininet-wifi. In *2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, pages 1–5. DOI:10.1109/AQTR55203.2022.9802017
- Fontes, R. R., Afzal, S., Brito, S. H. B., Santos, M. A. S., and Rothenberg, C. E. (2015). Mininet-wifi: Emulating software-defined wireless networks. In *2015 11th International Conference on Network and Service Management (CNSM)*, pages 384–389. DOI:10.1109/CNSM.2015.7367387
- Gemirter, C. B., Şenturca, , and Baydere, (2021). A comparative evaluation of amqp, mqtt and http protocols using real-time public smart city data. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, pages 542–547. DOI:10.1109/UBMK52708.2021.9559032
- Larmo, A., Ratilainen, A., and Saarinen, J. (2019). Impact of coap and mqtt on nb-iot system performance. *Sensors*, 19(1). DOI:10.3390/s19010007
- Mukundarajan, H., Hol, F. J. H., Castillo, E. A., Newby, C., and Prakash, M. (2017). Using mobile phones as acoustic sensors for high-throughput mosquito surveillance. *eLife*, 6:e27854. DOI:10.7554/eLife.27854
- Paim, K., Silva, T., Dilly, J., Tavares, A., Cunha, M., Silva, O., Recamonde-Mendoza, M., Mansilha, R., and Cordeiro, W. (2025). Acoustic identification of aedes aegypti: Experimental assessment of different machine learning techniques. In *Anais do LII Seminário Integrado de Software e Hardware*, pages 1–12, Porto Alegre, RS, Brasil. SBC. DOI:10.5753/semish.2025.6555
- Paim, K. O., Rohweder, R., Recamonde-Mendoza, M., Mansilha, R. B., and Cordeiro, W. (2024). Acoustic identification of ae. aegypti mosquitoes using smartphone apps and residual convolutional neural networks. *Biomedical Signal Processing and Control*, 95:106342. DOI:10.1016/j.bspc.2024.106342
- Rodrigues, Y. d. F., Rodrigues, R. N., Paim, K. O., Kreutz, D., Cordeiro, W., and Mansilha, R. B. (2024). MosqMon: um sistema distribuído para captura de faixas de áudio para treinamento de RNAs detectoras de mosquito Ae. aegypti. In *XXI Escola Regional de Redes de Computadores (ERRC)*. DOI:10.5753/errc.2024.4713
- Seoane, V., Garcia-Rubio, C., Almenares, F., and Campo, C. (2021). Performance evaluation of coap and mqtt with security support for iot environments. *Computer Networks*, 197:108338. DOI:10.1016/j.comnet.2021.108338
- Shelby, Z., Hartke, K., and Bormann, C. (2014). The Constrained Application Protocol (CoAP). Technical report, RFC Editor. DOI:10.17487/RFC7252
- Siddiqui, A. A. I. and Kayte, C. (2022). Mosquito tracking, classification, and identification: A glance at the technologies available. *International Journal of Mosquito Research*, 9(2):104–110. DOI:10.22271/23487941.2022.v9.i2b.606
- Souza, T., Neto, A. V., Rosário, D. L., Fontes, R. R., and Cerqueira, E. (2025). P4lowpan: Transforming iot networks with a programmable data plane and in-band telemetry. In *2025 International Wireless Communications and Mobile Computing (IWCMC)*, pages 14–19. DOI:10.1109/IWCMC65282.2025.11059469