

Avaliação Experimental de Desempenho Temporal e Robustez em Arquitetura IoT Distribuída Baseada em LoRaWAN

Bruna Maria Pereira de Sousa; Laurence Rodrigues do Amaral; Matheus de Souza Gomes e Pedro Luiz Lima Bertarini

Resumo— Este trabalho apresenta a avaliação experimental do desempenho temporal e da robustez do SALVE, uma arquitetura IoT distribuída baseada em LoRaWAN que integra sensoriamento, comunicação, processamento em *fog*, notificação ao usuário e persistência em nuvem. Foram analisados dois cenários: operação normal e falha de conectividade com *retry*. Os resultados mostram que as etapas críticas executadas localmente apresentam latência praticamente nula e comportamento determinístico, enquanto a interface supervisória mantém resposta em tempo quase real. Em contraste, a variabilidade fim a fim, em falha, está associada à indisponibilidade da rede. Ainda assim, o sistema demonstrou robustez, garantindo a entrega eventual dos dados.

Palavras-Chave— IoT, LoRaWAN, Latência, Computação em Névoa.

Abstract— This work presents an experimental evaluation of the temporal performance and robustness of SALVE, a distributed IoT architecture based on LoRaWAN that integrates sensing, communication, fog-based processing, user notification, and cloud data persistence. Two scenarios were analyzed: normal operation and connectivity failure with *retry*. The results show that locally executed critical stages exhibit near-zero latency and deterministic behavior, while the supervisory interface maintains near real-time responsiveness. In contrast, end-to-end variability under failure conditions is mainly associated with network unavailability. Nevertheless, the system demonstrated robustness, ensuring eventual data delivery.

Keywords— IoT, LoRaWAN, Latency, Fog Computing.

I. INTRODUÇÃO

A crescente adoção de sistemas baseados em Internet das Coisas (IoT) tem impulsionado aplicações distribuídas que exigem eficiência energética, escalabilidade, baixa latência e confiabilidade operacional. Em aplicações críticas, como sistemas industriais e de controle, estes requisitos são ainda mais rigorosos, demandando tempos de resposta reduzidos [1]. Entretanto, a centralização do processamento em nuvem pode introduzir latência de comunicação e aumentar o tráfego na rede, limitando o atendimento de aplicações IoT sensíveis ao tempo [2], [8], [9]. Neste contexto, *edge* e *fog computing* aproximam recursos de processamento, armazenamento e comunicação dos dispositivos, reduzindo o atraso de serviço e a dependência exclusiva da infraestrutura remota [8], [9].

Esses paradigmas não devem ser compreendidos apenas como alternativas isoladas à nuvem, mas como camadas complementares. *Edge* e *fog* favorecem decisões locais e respostas em tempo reduzido, enquanto a *cloud* mantém vantagens para armazenamento e processamento de maior escala [13]. Como cada paradigma apresenta restrições próprias de armazenamento, cobertura, latência e capacidade, arquiteturas integradas *edge-fog-cloud* permitem distribuir as funções conforme os requisitos da aplicação [10], [11]. Esta integração, contudo, introduz desafios relacionados à gestão de recursos, sincronização, distribuição de dados e garantia de qualidade de serviço [11], [12].

Estudos recentes abordam diferentes dimensões dessas arquiteturas. Das e Inuwa [8] sistematizam questões operacionais, problemas de serviço, gerenciamento de dados e segurança em *fog computing*; Chiang et al. [9] concentram-se na gestão de recursos e orquestração de serviços em *edge*; Kumar e Agrawal [10] analisam *frameworks* integrados *edge-fog-cloud*, comparando vantagens, desvantagens, dados multidimensionais gerados e desafios de implementação; Kar et al. [11] discutem abordagens de otimização para *offloading* em nuvem, névoa e borda; e Walia et al. [12] examinam mecanismos de gerenciamento de recursos em *fog/edge* e métricas de QoS. Entretanto, estes trabalhos concentram-se predominantemente em arquiteturas, orquestração e alocação de recursos, permanecendo espaço para avaliações experimentais de *pipelines* operacionais completos sob operação normal e falhas de conectividade.

Neste contexto, este trabalho avalia experimentalmente o desempenho temporal e a robustez do SALVE, uma arquitetura *edge-fog-cloud* baseada em LoRaWAN para detecção de eventos, processamento local, notificação e persistência remota. A arquitetura é estruturada em múltiplas camadas, integrando dispositivos de sensoriamento, comunicação LoRaWAN e processamento local determinístico. Para a validação experimental, são definidos dois cenários de avaliação de latência: (i) aplicação em operação normal e (ii) aplicação sob falha de conectividade. A contribuição consiste na decomposição temporal do *pipeline* em múltiplos pontos e na avaliação conjunta da operação normal e da recuperação após falhas, utilizando métricas como P95 e IQR.

II. REFERENCIAL TEÓRICO

Sistemas IoT integram dispositivos heterogêneos responsáveis por sensoriamento, comunicação e geração de dados, frequentemente sob restrições de processamento,

Bruna Maria Pereira de Sousa, Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Patos de Minas-MG, e-mail: brunampsousa@ufu.br; Laurence Rodrigues do Amaral, Faculdade de Computação, Universidade Federal de Uberlândia, Patos de Minas-MG, e-mail: laurence@ufu.br; Matheus de Souza Gomes, Instituto de Biotecnologia, Universidade Federal de Uberlândia, Patos de Minas-MG, e-mail: matheusgomes@ufu.br; Pedro Luiz Lima Bertarini, Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Patos de Minas-MG, e-mail: bertarini@ufu.br. Este trabalho foi financiado por FAPEMIG (APQ-05123-23).

armazenamento e energia. O aumento da quantidade e da diversidade destes dados amplia a demanda por recursos computacionais e por mecanismos capazes de atender requisitos de latência, confiabilidade e escalabilidade [4], [5], [12]. Inicialmente, predominavam modelos centralizados baseados em *cloud computing*, nos quais dispositivos enviam dados para processamento remoto. Este modelo, associado ao conceito de *Cloud of Things* (CoT) facilita o armazenamento e a análise de grandes volumes de dados [3].

Embora a *cloud* ofereça elevada capacidade de processamento e armazenamento, sua distância das fontes de dados pode limitar aplicações de tempo real [3], [11], [13]. A *Edge computing* posiciona recursos no dispositivo ou na borda imediata da rede [6], [10], enquanto a *fog computing* introduz nós intermediários para processamento, armazenamento e decisões locais [8]. A integração *edge-fog-cloud* combina resposta local e recursos remotos, mas introduz desafios de gerenciamento, distribuição de dados e sincronização [10], [12]. A aplicação destes paradigmas alcança o setor agropecuário, de modo que sistemas baseados em *fog computing* podem apoiar o controle da produção e o monitoramento em tempo real de cultivos e animais, incluindo aplicações de gestão pecuária [8].

Para a comunicação entre dispositivos distribuídos, redes LPWAN oferecem longo alcance, baixo consumo energético e suporte a dispositivos com baixa taxa de geração de dados [7]. O LoRaWAN adota topologia estrela-de-estrelas (Figura 1), na qual *gateways* encaminham mensagens entre dispositivos finais e um servidor de rede conectado por infraestrutura IP [14]. A comunicação utiliza diferentes canais e taxas de dados, cuja seleção envolve compromisso entre alcance e duração da mensagem, podendo ser ajustada pelo mecanismo *Adaptive Data Rate* (ADR) [14].

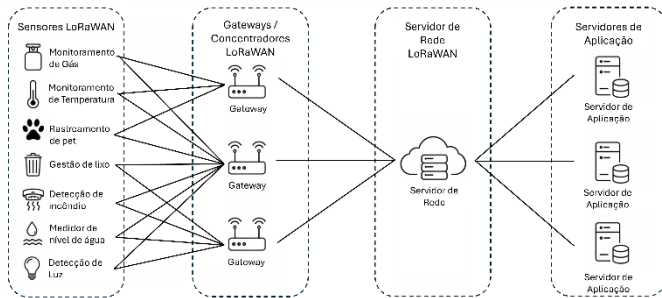


Fig. 1. Topologia estrela-de-estrelas da rede LoRaWAN, com dispositivos finais, *gateways*, servidor de rede e servidor de aplicação.

Além da arquitetura lógica, o comportamento temporal depende dos parâmetros regionais e físicos da transmissão. Na região AU915, as taxas de *uplink* associam diferentes combinações de *spreading fator* (SF) e largura de banda (BW), incluindo configurações entre SF12/125 kHz e SF8/500 kHz [15]. Consequentemente, SF, BW, tamanho do *payload* e restrições regionais influenciam a duração da transmissão e devem ser considerados na interpretação das latências observadas.

Na classe A do protocolo LoRaWAN, cada transmissão *uplink* é seguida por duas janelas curtas de recepção, e mensagens *downlink* fora destas janelas precisam aguardar um novo *uplink* do dispositivo [14]. Este funcionamento reduz o consumo energético, mas pode introduzir variabilidade no

tempo de entrega de notificações ao dispositivo final. Desta forma, a classe de operação e o ciclo de comunicação devem ser considerados na interpretação das latências de notificação observadas no *wearable*.

Em sistemas distribuídos, a avaliação de desempenho não deve se limitar à operação normal, pois falhas ou interrupções de rede podem comprometer a disponibilidade dos dados. Arquiteturas *fog* orientadas à confiabilidade buscam manter dados e serviços acessíveis durante falhas por meio de armazenamento distribuído, balanceamento de carga e mecanismos de *failover* [8]. De forma complementar, a integração entre *cloud*, *edge* e *fog* permite distribuir as funções de acordo com requisitos de latência, capacidade e disponibilidade [11].

No SALVE, esses princípios são implementados por persistência local, enfileiramento e retransmissão automática, como apresentado na Figura 2b. Quando o serviço *cloud* está indisponível, o evento permanece armazenado localmente e é reenviado após a recuperação da conectividade. Assim, a análise conjunta da latência em operação normal e do processo de *retry* permite avaliar tanto o desempenho temporal quanto a capacidade de entrega eventual da arquitetura.

III. METODOLOGIA

Este trabalho avalia experimentalmente o sistema SALVE, que consiste em uma solução IoT distribuída para detecção de eventos de risco em matrizes suínas, geração de alertas em tempo real e persistência estruturada de dados para análise posterior.

Os experimentos foram conduzidos em ambiente controlado, utilizando uma instância funcional da arquitetura SALVE executada em computador pessoal, integrando sensores LoRaWAN, *gateway*, servidor de rede, *broker* MQTT, aplicação *backend* e *frontend* local, dispositivo vestível, serviço *cloud* e banco de dados. Os eventos utilizados nos testes foram gerados de forma controlada por meio da simulação da condição de risco, permitindo reproduzir o fluxo operacional do sistema sem interferências externas do ambiente produtivo. A escolha deste cenário experimental possibilitou avaliar de forma reprodutível o comportamento temporal da arquitetura e os mecanismos de comunicação, processamento e sincronização de dados. Embora os testes tenham sido realizados em ambiente controlado, a aplicação está diretamente associada à suinocultura de precisão, na qual baixa latência e robustez operacional são fundamentais para suporte à detecção de eventos críticos e geração de alertas em tempo adequado.

A Tabela I resume os principais componentes utilizados na plataforma experimental, enquanto a Tabela II apresenta os parâmetros de configuração da comunicação LoRaWAN empregados durante os experimentos. Estes parâmetros permaneceram constantes em todos os cenários avaliados, garantindo que as diferenças observadas nas métricas de latência fossem atribuídas exclusivamente às condições experimentais investigadas.

O perfil LoRaWAN utilizou ADR em modo *Auto-Adjust*. Desta forma, o SF e a BW efetivamente empregados em cada *uplink* puderam variar dinamicamente conforme as condições da rede, sendo apresentada na Tabela II a configuração máxima definida para os experimentos.

TABELA I. COMPONENTES DA PLATAFORMA EXPERIMENTAL.

Componente	Especificação
Sensores	Dragino DDS45-LB e CPL03-LB
Gateway	Dragino LG308N
Servidor de rede	Petr Gotthard LoRaWAN Server v0.6.7
Broker MQTT	Eclipse Mosquitto
Infra fog	SALVE Core (back e frontend Python) + PostgreSQL
Dispositivo Vestível	LilyGO T-Watch S3 + SALVE Sentinel (firmware C++ ESP32)
Infra cloud	VPS Linux + PostgreSQL

TABELA II. PARÂMETROS DA COMUNICAÇÃO LoRAWAN UTILIZADOS NOS EXPERIMENTOS.

Parâmetro	Valor
Região e sub-banda	AU915, FSB2
Classe e ativação	Classe A, OTAA
Coding Rate (CR)	4/5
ADR	Auto-Adjust
Taxa máxima	SF8, BW 500 kHz, 12,5 kbit/s
Payloads	DDS45: 8 bytes; CPL03: 11 bytes

A metodologia adotada foi estruturada para analisar o comportamento temporal do sistema sob duas perspectivas: (i) desempenho da aplicação e (ii) robustez frente a falhas de conectividade. Para isto, foram definidos dois cenários experimentais complementares, baseados na observação temporal de eventos em múltiplos pontos da arquitetura.

A. Cenário A: Desempenho da Aplicação

Os testes realizados mediante o cenário A tiveram como objetivo avaliar o desempenho da aplicação durante a operação normal, considerando o fluxo completo de processamento de um evento desde o recebimento do dado pela aplicação *backend* local até a persistência remota. Os *timestamps* foram coletados em diferentes etapas do *pipeline*, apresentado na Figura 2a, permitindo a definição das métricas de latência apresentadas na Tabela III.

As métricas definidas no cenário A permitem decompor o tempo de resposta do sistema em diferentes etapas do *pipeline* de processamento, desde a chegada do dado até sua persistência final. A latência de decisão local T_l representa o tempo necessário para que a aplicação *backend* processe o dado recebido da arquitetura de rede e classifique o evento como alerta, refletindo diretamente a eficiência do algoritmo de detecção. As latências T_{ui_pc} e T_{ui_sw} correspondem, respectivamente, ao tempo de propagação do alerta até sua visualização no *dashboard* do computador e no dispositivo *smartwatch*, sendo indicativas da responsividade do sistema sob a perspectiva do usuário.

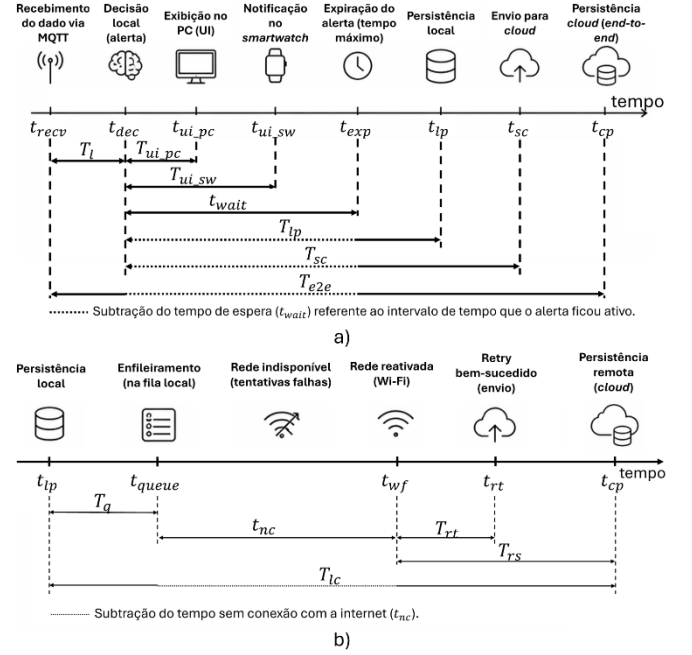


Fig. 2. Pipelines temporais com os pontos de coleta dos timestamps no a) cenário A e no b) cenário B.

TABELA III. MÉTRICAS DE LATÊNCIA DEFINIDAS PARA O CENÁRIO A E RESPECTIVAS FÓRMULAS DE CÁLCULO.

Métricas de latência	Fórmula
Decisão local	$T_l = t_{dec} - t_{recv}$
Visualização supervisória	$T_{ui_pc} = t_{ui_pc} - t_{dec}$
Visualização no vestível	$T_{ui_sw} = t_{ui_sw} - t_{dec}$
Tempo ativo do alerta	$t_{wait} = t_{exp} - t_{dec}$
Persistência local	$T_{lp} = (t_{lp} - t_{dec}) - t_{wait}$
Envio ao cloud	$T_{sc} = (t_{sc} - t_{dec}) - t_{wait}$
Persistência cloud fim a fim	$T_{eze} = (t_{cp} - t_{recv}) - t_{wait}$

Já as métricas T_{lp} , T_{sc} e T_{eze} caracterizam as etapas posteriores ao encerramento do alerta, incluindo persistência local, envio ao serviço *cloud* e persistência remota fim a fim. Nestas métricas, a subtração do intervalo t_{wait} é fundamental para eliminar o tempo em que o sistema permanece inativo aguardando a expiração do alerta, permitindo que os valores obtidos representem exclusivamente o tempo efetivo de processamento e comunicação, sem influência da lógica temporal configurada na aplicação.

B. Cenário B: Robustez Frente a Falhas de Conectividade

No cenário B, avaliou-se o comportamento do sistema durante falha de conectividade e posterior recuperação, com foco no mecanismo de *retry* baseado em persistência local e enfileiramento de eventos. Durante a indisponibilidade da rede, os eventos são armazenados localmente e inseridos em uma fila persistente, sendo retransmitidos automaticamente após o restabelecimento da conectividade. A Figura 2b apresenta o

pipeline deste cenário evidenciando os *timestamps* coletados em cada etapa e a Tabela IV expõe as métricas calculadas.

Estas métricas permitem caracterizar o comportamento temporal do sistema durante períodos de indisponibilidade de rede e sua posterior recuperação. A latência de enfileiramento T_q representa o intervalo entre a persistência local do evento e sua inserção na fila de envio, refletindo o tempo necessário para preparação do mecanismo de *retry*. A latência de *retry* bem-sucedido T_{rt} mede o tempo decorrido entre a reativação da conectividade e o envio efetivo do evento ao *cloud*, sendo um indicador direto da eficiência do algoritmo de reenvio.

TABELA IV. MÉTRICAS DE LATÊNCIA DEFINIDAS PARA O CENÁRIO B E RESPECTIVAS FÓRMULAS DE CÁLCULO.

Métricas de latência	Fórmula
Enfileiramento	$T_q = t_{queue} - t_{lp}$
Retry bem-sucedido	$T_{rt} = t_{rt} - t_{wf}$
Ressincronização	$T_{rs} = t_{cp} - t_{wf}$
Tempo sem conexão	$t_{nc} = t_{wf} - t_{queue}$
Entre persistência local e remota	$T_{lc} = (t_{cp} - t_{lp}) - t_{nc}$

Por outro lado, a latência de ressincronização T_{rs} corresponde ao tempo entre o restabelecimento da rede e a persistência remota do dado, englobando tanto o *retry* quanto o processamento no serviço *cloud*. Por fim, a métrica T_{lc} representa a latência total entre a persistência local e a persistência remota, subtraindo o tempo em que não havia conexão à *internet* (t_{nc}). Esta subtração é feita de modo a capturar o impacto completo da indisponibilidade da rede no ciclo de vida do evento sem a interferência de quanto tempo o sistema ficou sem conectividade à *internet*. Em conjunto, estas métricas permitem avaliar a robustez do sistema e sua capacidade de garantir consistência eventual dos dados mesmo em cenários adversos de conectividade.

C. Processamento e Análise Estatística dos Dados

Os dados coletados nos dois cenários experimentais foram consolidados e processados em ambiente Python, com o uso do tipo *Decimal* para garantir alta precisão nos cálculos envolvendo *timestamps*. Durante a etapa de limpeza, foram removidos eventos que apresentaram latências negativas, uma vez que estes valores indicam inconsistências temporais decorrentes de diferenças entre fontes de coleta, resolução de registro ou desalinhamento entre relógios dos componentes do sistema.

A análise estatística foi conduzida com métricas descritivas e indicadores de dispersão para caracterizar o comportamento das latências. Para cada métrica, foram calculados contagem, média, mediana, moda, mínimo, máximo, amplitude, variância amostral, desvio padrão amostral, percentis P25, P50, P75 e P95, além do intervalo interquartil (IQR). Estas medidas permitem descrever o comportamento típico, a variabilidade e a distribuição central dos dados, além de fundamentar os *boxplots* apresentados nos resultados. O P95 foi usado para representar o limite abaixo do qual ocorrem 95% dos casos, evidenciando condições próximas ao pior caso típico de

latência. O IQR foi adotado como medida robusta de dispersão, menos sensível a *outliers*, para avaliar a estabilidade das latências entre os componentes do sistema.

IV. RESULTADOS E DISCUSSÃO

No cenário A (Figura 3), que avalia a operação normal, observa-se que as métricas associadas ao processamento local, T_l e T_{lp} , apresentam valores praticamente nulos (mediana igual a 0) e dispersão desprezível (*IQR* aproximadamente 0), com P_{95} inferior a 0.003 e 0.0011 segundos, respectivamente. Estes resultados confirmam o caráter determinístico do processamento em *fog* e mostram que a latência computacional local é negligenciável.

A latência de resposta na interface supervisória (T_{ui_pc}) apresenta comportamento estável, com mediana de 0.49 segundos, P_{95} inferior a 1 segundo e *IQR* reduzido (0,37 segundo), indicando resposta consistente em tempo quase real. Em contraste, a latência de notificação no *smartwatch* (T_{ui_sw}) apresenta elevada variabilidade (*IQR* = 14.85 segundos) e forte assimetria, com P_{95} superior a 24 segundos, evidenciando influência significativa das características do enlace LoRaWAN e do mecanismo de recepção do dispositivo.

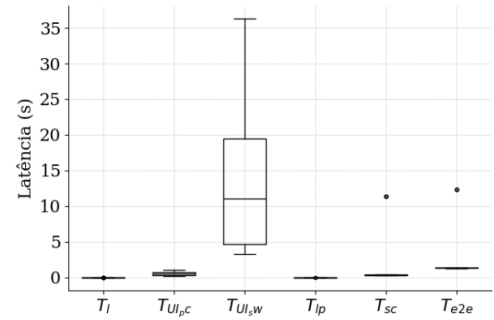


Fig. 3. Distribuição das latências observadas no cenário A, apresentada por meio de *boxplots* para as diferentes métricas avaliadas.

As métricas associadas ao *cloud* (T_{sc} e T_{e2e}) apresentam comportamento assimétrico, com valores centrais reduzidos (medianas de 0.32 e 1.31 segundos), mas com P_{95} elevados (aproximadamente 11.37 e 12.36 segundos), indicando ocorrência de eventos extremos. Ainda assim, o *IQR* reduzido (aproximadamente 0.03 segundo) indica que, na maior parte dos casos, o comportamento é estável, sendo a variabilidade atribuída a *outliers* típicos de sistemas distribuídos. Como estas etapas não pertencem ao caminho crítico de notificação, seu impacto na experiência do usuário é limitado.

No cenário B (Figura 4), os resultados mostram o comportamento do sistema sob indisponibilidade de conectividade. A latência de enfileiramento (T_q) apresenta mediana próxima de zero (0.003 segundo) e *IQR* extremamente reduzido (0.0017 segundo), indicando inserção praticamente imediata dos eventos na fila local. Entretanto, o P_{95} elevado (aproximadamente 12 segundos) revela a ocorrência de poucos eventos com atraso significativo, possivelmente associados a condições específicas de processamento ou concorrência.

Por outro lado, as latências de *retry* T_{rt} e ressincronização T_{rs} apresentam comportamento estatístico muito semelhante, com medianas próximas de 200 segundos, *IQR* elevado (aproximadamente 66 segundos) e P_{95} superior a 250 segundos.

Estes valores refletem diretamente o intervalo periódico para reenvio ao serviço remoto, que ocorre a cada cinco minutos de execução do *software* de processamento.

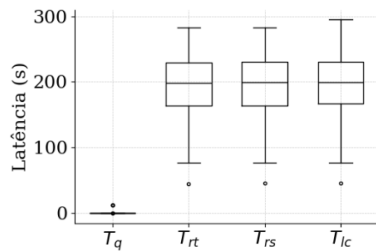


Fig. 4. Distribuição das latências observadas no cenário B, apresentada por meio de *boxplots* para as métricas associadas ao mecanismo de *retry* e à resincronização.

A latência entre persistência local e remota (T_{lc}) apresenta distribuição consistente com estas métricas, com mediana de aproximadamente 199 segundos, *IQR* de 63.9 segundos e P_{95} próximo de 266 segundos, evidenciando forte correlação com o processo de recuperação da conectividade. A proximidade entre T_{rs} , T_{rs} e T_{lc} indica que, uma vez restabelecida a rede, o envio e a persistência remota ocorrem de forma eficiente, sem introdução de atrasos adicionais relevantes.

Em conjunto, os percentis e o *IQR* mostram que o processamento em *fog* é rápido e estável, que a variabilidade das etapas *cloud* concentra-se em eventos extremos e que, sob falha, o desempenho é dominado pela indisponibilidade da rede. A arquitetura, portanto, mantém baixa latência no caminho crítico e entrega eventual dos dados.

V. CONCLUSÕES

Este trabalho avaliou experimentalmente o desempenho temporal e a robustez do SALVE, uma arquitetura IoT distribuída baseada em computação *edge*, *fog* e *cloud*. Os resultados demonstraram que o processamento na camada *fog* apresenta baixa latência e comportamento altamente previsível nas etapas críticas do sistema, validando o processamento local para aplicações sensíveis ao tempo.

A análise da interação com o usuário indicou que a interface supervisória mantém resposta consistente em tempo quase real, enquanto a notificação em dispositivos *smartwatch* apresenta maior variabilidade, associada à abertura das janelas de recepção em classe A. As etapas dependentes de *cloud*, por sua vez, apresentaram comportamento mais variável, sem impactar o caminho crítico de resposta ao usuário. Sob condições de falha de conectividade, a arquitetura demonstrou robustez, garantindo entrega eventual dos dados por meio de mecanismos de fila persistente e reenvio. Os resultados indicam que o impacto temporal nestes cenários é predominantemente determinado pela indisponibilidade da rede, enquanto o sistema mantém comportamento consistente e eficiente internamente.

De forma geral, a análise estatística, incluindo medidas baseadas em distribuição, permitiu caracterizar o desempenho médio e a estabilidade do sistema. Os resultados confirmam que a arquitetura proposta é adequada para aplicações IoT distribuídas que demandam baixa latência e alta disponibilidade. Como trabalhos futuros, propõe-se a

otimização das etapas associadas à comunicação com o *cloud* e a avaliação do sistema em cenários com maior densidade de dispositivos e diferentes configurações LoRaWAN, visando investigar aspectos de escalabilidade e desempenho em ambientes mais complexos.

AGRADECIMENTOS

À FAPEMIG, pelo apoio financeiro (APQ-05123-23), e às empresas Auma Tecnologia e Auma Suínos, pelo suporte técnico e infraestrutura de testes. Ao LATUR e Laboratório de Internet das Coisas da UFU campus Patos de Minas, pelo apoio institucional e infraestrutura de pesquisa.

REFERÊNCIAS

- [1] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities", *IEEE Internet of Things Journal*, v. 3(6), pp. 854–864, 2016.
- [2] J. Hong, Y. G. Hong, X. de Foy, M. Kovatsch, E. Schooler and D. Kutscher, "RFC 9556: Internet of Things (IoT) Edge Challenges and Functions", *IRTF*, 2024.
- [3] M. Burhan, H. Alam, A. Arsalan, R. A. Rehman, M. Anwar, M. Faheem and M. W. Ashraf, "A Comprehensive Survey on the Cooperation of Fog Computing Paradigm-Based IoT Applications: Layered Architecture, Real-Time Security Issues, and Solutions", *IEEE Access*, v. 11, pp. 73303–73329, 2023.
- [4] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications", *Journal of Electrical and Computer Engineering*, v. 2017(1), pp. 9324035, 2017.
- [5] M. M. Islam, S. Nooruddin, F. Karray and G. Muhammad, "Internet of Things: Device Capabilities, Architectures, Protocols, and Smart Applications in Healthcare Domain", *IEEE Internet Of Things Journal*, v. 10(4), pp. 3611–3641, 2023.
- [6] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges", *IEEE Internet Of Things Journal*, v. 3(5), pp. 637–646, 2016.
- [7] N. S. Chilamkurthy, O. J. Pandey, A. Ghosh, L. R. Cenkeramaddi and H. Dai, "Low-Power Wide-Area Networks: A Broad Overview of Its Different Aspects", *IEEE Access*, v. 10, pp. 81926–81959, 2022.
- [8] R. Das and M. M. Inuwa, "A Review on Fog Computing: Issues, Characteristics, Challenges, and Potential Applications", *Telematics and Informatics Reports*, v. 10, pp. 100049, 2023.
- [9] Y. Chiang, Y. Zhang, H. Luo, T.-Y. Chen, G.-H. Chen, H.-T. Chen, Y.-J. Wang, H.-Y. Wei and C.-T. Chou, "Management and Orchestration of Edge Computing for IoT: A Comprehensive Survey", *IEEE Internet of Things Journal*, v. 10(16), pp. 14307–14331, 2023.
- [10] R. Kumar and N. Agrawal, "Analysis of Multi-Dimensional Industrial IoT (IIoT) Data in Edge-Fog-Cloud Based Architectural Frameworks: A Survey on Current State and Research Challenges", *Journal of Industrial Information Integration*, v. 35, pp. 100504, 2023.
- [11] B. Kar, W. Yahya, Y.-D. Lin and A. Ali, "Offloading Using Traditional Optimization and Machine Learning in Federated Cloud-Edge-Fog Systems: A Survey", *IEEE Communications Surveys & Tutorials*, v. 25(2), pp. 1199–1226, 2023.
- [12] G. K. Walia, M. Kumar and S. S. Gill, "AI-Empowered Fog/Edge Resource Management for IoT Applications: A Comprehensive Review, Research Challenges, and Future Perspectives", *IEEE Communications Surveys & Tutorials*, v. 26(1), pp. 619–669, 2024.
- [13] A. Chakraborty, M. Kumar, N. Chaurasia and S. S. Gill, "Journey from Cloud of Things to Fog of Things: Survey, New Trends, and Research Directions", *Software: Practice and Experience*, v. 53(2), pp. 496–551, 2023.
- [14] LoRa Alliance Technical Committee, "LoRaWAN Specification v1.0.3", *LoRa Alliance*, 2018.
- [15] LoRa Alliance Technical Committee Regional Parameters Workgroup, "LoRaWAN Regional Parameters RP002-1.0.5", *LoRa Alliance*, 2025.