

Unidos vencemos: melhoria de desempenho em redes de sensores sem fio definidas por *software*

Lion Chen e Renan C. A. Alves

Resumo—As redes de sensores sem fio enfrentam restrições energéticas e de memória de forma que é crucial buscar soluções mais eficientes. Nesse contexto, foram propostas, implementadas e testadas técnicas de manipulação do processamento de pacotes, incluindo agregação de pacotes, descarte de redundâncias e aprimoramento do mecanismo de configuração de fluxo através de configuração múltipla. Os resultados indicam melhoria de desempenho: mínimo de 95% em taxa de entrega de pacotes de dados, cerca de 6,6% de redução em consumo de energia e diminuição de 41% em pacotes de configuração de fluxo.

Keywords—Redes de Sensores Sem Fio, Redes Definidas por Software, Análise de Desempenho.

Abstract—Wireless sensor networks face energy and memory constraints, making it crucial to seek efficient solutions. In this context, techniques for manipulating packet processing were proposed, implemented, and tested, including packet aggregation, redundancy removal, and improvement of the flow configuration mechanism through multiple configuration. The results indicate performance improvements: a minimum of 95% in data packet delivery rate, approximately 6.6% reduction in energy consumption, and a 41% decrease in flow configuration packets.

Keywords—Wireless Sensor Networks, Software Defined Networking, Performance Analysis.

I. INTRODUÇÃO

As redes de sensores sem fio (*Wireless Sensor Networks*, WSN) extraem informações do ambiente, como temperatura e umidade, para aplicações de *Internet* das Coisas. São compostas por dispositivos com capacidades limitadas de processamento, armazenamento, comunicação e energia. Os protocolos de comunicação utilizados nestas redes são adaptados para atender a estas restrições, visando reduzir a quantidade de processamento, memória e dados de controle necessários para seu funcionamento. Neste contexto, introduziu-se a ideia de aplicar o paradigma das redes definidas por *software* (*Software Defined Networking*, SDN) às WSNs, concentrando tarefas computacionalmente custosas em um controlador central, reduzindo a demanda sobre os nós sensores, cunhando o termo redes de sensores definidas por *software* (*Software Defined Wireless Sensor Networking*, SDWSN) [6].

O desempenho das SDWSNs é comparável ao de protocolos de roteamento distribuídos, apresentando uma taxa de entrega de pacotes de dados em níveis similares e, simultaneamente, alcançando redução no consumo de energia em alguns cenários [1].

Lion Chen, Renan C. A. Alves, Escola de Artes, Ciência e Humanidades da Universidade de São Paulo, São Paulo-SP, e-mails: lion18@usp.br, renanalves@usp.br. Este trabalho foi financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Brasil, número do processo: #2025/06085-5.

No entanto, ainda há problemas de desempenho que se revelam em situações específicas [7]. Em particular, considerando que as WSNs possuem nós que funcionam como sorvedouro de dados, levanta-se a possibilidade de reduzir perdas de pacotes por estouro de memória e por colisões na camada de enlace através da agregação de pacotes que têm um destino comum. Outra possibilidade de melhoria de desempenho consiste em empregar mecanismos mais eficientes de configuração de fluxo nos nós sensores. Apesar destes mecanismos terem sido previstos anteriormente na teoria [3], [5], [4], não foram implementados na prática no contexto de SDWSN e não há evidências de que realmente são capazes de melhorar o desempenho.

Assim, os objetivos deste trabalho são implementar, testar e aferir os ganhos de desempenho de um mecanismo de agregação de pacotes (Seção II) e de um mecanismo de configuração de fluxo múltiplo (Seção III) aplicados ao IT-SDN [3], uma arquitetura de SDWSN de código aberto. Por fim, conclusões são apresentadas na Seção IV.

II. AGREGAÇÃO DE PACOTES

A limitada capacidade de armazenamento dos nós sensores torna as filas de envio e recepção de pacotes vulneráveis a picos de tráfego, podendo causar perdas de pacotes por *overflow*. Para mitigar esse problema, foi adicionada uma etapa de processamento ao inserir pacotes nestas filas, visando reduzir o tamanho médio da fila, o volume de transmissões e o congestionamento da rede.

Os pacotes são processados de acordo com os seguintes critérios: **1) agregação**, aplicada a pacotes com o mesmo destino e *payloads* distintos; e **2) substituição**, realizada em situações de redundância estrita, quando pacotes de controle apresentam conteúdo, origem e destino idênticos. Assim, novos pacotes são comparados aos já armazenados na fila, avaliando o seu tipo, destino, rota e o conteúdo. Em seguida, o pacote é substituído, agregado ou simplesmente adicionado no final da fila.

Na substituição, o novo pacote sobrescreve o antigo na mesma posição, mantendo o cabeçalho atualizado e evitando a propagação de dados obsoletos.

Na agregação, o conteúdo do novo pacote é combinado ao conteúdo do pacote armazenado na fila, e este passa a assumir uma estrutura agregada que comporta múltiplos *payloads*, respeitando o limite imposto pelo protocolo IEEE 802.15.4. A Figura 1 ilustra a estrutura de pacotes agregados, que contém um campo para indicar a quantidade de subpacotes.

Por fim, o processo de desagregação ocorre no destino final, onde os pacotes agregados são separados em subpacotes, reencapsulados e processados individualmente.

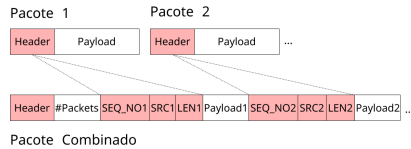


Fig. 1. Estrutura de pacotes agregados

A. Resultados

O impacto das técnicas foi avaliado utilizando o simulador Cooja, comparando uma versão do IT-SDN original adaptada ao Contiki-NG a uma versão modificada com agregação de pacotes. Utilizou-se topologias em grade e aleatórias, variando o número de nós de 16 a 169, mantendo constantes o tempo de simulação (1 hora) e utilizando o *Improved-ND* [2] como o protocolo de descoberta de vizinhos.

A avaliação se ateve às métricas de taxa de entrega de pacotes de dados e consumo médio de energia de envio por nó. Para validar as observações, foi conduzida uma análise utilizando testes ANOVA com nível de significância de 5%.

O mecanismo de agregação de pacotes resultou em uma melhoria estatisticamente significativa na taxa de entrega de pacotes de dados ($F = 61,77$, $p < 0,001$). A versão modificada elevou a média global de entrega de 90,6% para 98,0% ($Cohen's d = 0,787$), mantendo taxas superiores a 95% em ambas as topologias, independentemente da escala, enquanto o IT-SDN original degradou para cerca de 70% nos cenários mais críticos. Para exemplificar o impacto, mais de 5.000 pacotes ($\approx 40\%$ do total) foram agregados na topologia aleatória de 169 nós, sugerindo que a melhoria de desempenho é devida à redução de *overflows*, de congestionamento e de retransmissões (Figura 2), mesmo em cenários com variação estocástica na conectividade das topologias aleatórias.

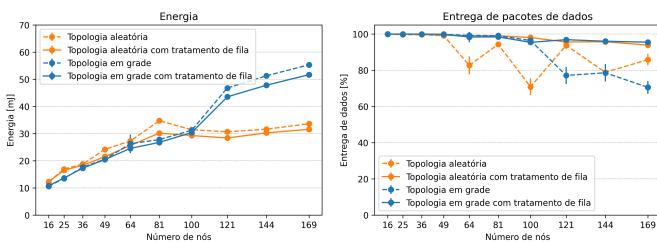


Fig. 2. Energia e taxa de entrega de pacotes de dados

Por fim, a análise do consumo médio de energia por nó mostra que a redução global da média (de 28,1 mJ para 26,5 mJ) não apresentou significância estatística generalizada ($F = 1,90$, $p = 0,168$). Contudo, houve economia energética evidente e pontual em cenários com maior densidade de nós. Na topologia em grade com 169 nós, o consumo médio foi reduzido de 55,3 mJ para 51,7 mJ, representando uma economia aproximada de 6,6%, decorrente principalmente da diminuição do número de transmissões.

III. CONFIGURAÇÃO DE MÚLTIPLOS FLUXOS

Nas redes definidas por *software*, a configuração de fluxo ocorre de forma centralizada: um nó sensor envia um *flow*

request ao controlador quando precisa encaminhar pacotes para um destino que não consta em sua tabela de fluxo. O controlador, por sua vez, calcula a rota e envia um *flow setup* individual para cada nó do caminho. Este envio individual gera um grande volume de pacotes de controle, tornando-se uma fonte de congestionamento e gasto de energia, especialmente em topologias amplas ou com atualizações frequentes.

Utilizar pacotes de *Multiple Data Flow Setup* (MFS) pode mitigar esse sobrecusto, pois permite configurar fluxos em múltiplos nós utilizando um único pacote de controle.

Para orquestrar o envio, o controlador acumula requisições de atualização e envia periodicamente um pacote MFS aos nós envolvidos. É importante ressaltar que não há garantia sobre a ordem de recebimento dos *flow_request*. Assim, se o controlador respondesse imediatamente, no pior caso o sistema se comportaria como a abordagem original, enviando um pacote de *flow setup* para cada nó requisitante.

Dessa forma, o controlador extrai todos os caminhos possíveis iniciando nos nós com grau de entrada igual a zero (nós folhas). Esses caminhos representam sequências ordenadas de nós que potencialmente necessitam de atualização de regras de fluxo. Assim, o processo de extração é realizado por meio de uma busca recursiva, que percorre o grafo acumulando os endereços visitados e evitando ciclos, de forma a garantir a validade dos caminhos obtidos.

Após a geração desses caminhos, é realizada uma etapa de resolução de conflitos entre segmentos compartilhados. Como diferentes caminhos podem conter trechos em comum, o controlador identifica segmentos repetidos e seleciona apenas o caminho de menor custo para manter naquele segmento, onde o custo é definido pela posição do segmento ao longo do caminho. Caminhos que perdem um conflito têm seu comprimento truncado, evitando a duplicação de configurações e garantindo que cada trecho da rede seja atualizado por apenas um pacote MFS.

Por fim, o resultado final desse processo é um conjunto otimizado de caminhos disjuntos ou parcialmente disjuntos, que servem como base para o envio dos pacotes de configuração múltipla. Cada caminho corresponde a uma rota pela qual um único pacote pode percorrer a rede, configurando sequencialmente as tabelas de fluxo dos nós envolvidos.

A. Resultados

Os testes compararam três cenários em topologias aleatórias (16 a 169 nós) com agregação habilitada: IT-SDN padrão e IT-SDN com MFS nos intervalos de 5 s e 10 s. Ressalta-se que o objetivo desta etapa foi validar a viabilidade do conceito de *Multiple Flow Setup* e aferir seu impacto direto no *overhead* de controle. Dessa forma, a determinação de um intervalo ótimo de envio requer um estudo exclusivo de modelagem e otimização de tráfego, o qual não constitui o foco principal deste trabalho. Ademais, a escolha específica dos intervalos de 5 e 10 segundos visou balancear a capacidade do controlador em acumular rotas com o atraso na atualização da rede.

Em uma primeira análise (Figura 3), observa-se que a utilização do pacote de MFS manteve valores de taxa de entrega próximos aos obtidos com o roteamento por fonte

padrão, ainda que com leves inconstâncias em topologias com mais de 121 nós. Os dois intervalos de processamento do controle de fluxos testados, por sua vez, não apresentaram impacto significativo sobre esta métrica, indicando que, dentro dos intervalos avaliados, a periodicidade adotada não é determinante para a taxa de entrega de pacotes de dados.

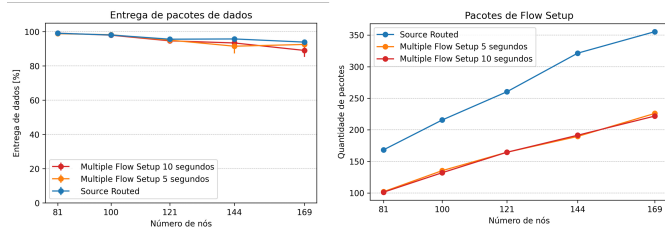


Fig. 3. Taxa de entrega e quantidade de pacotes de *flow setup* enviados.

Por outro lado, conforme apresentado na Figura 3, nota-se uma redução expressiva na quantidade de pacotes de configuração de fluxo enviados pelo controlador. A adoção do *flow setup* múltiplo resultou em uma diminuição de aproximadamente 41% em relação ao cenário original. Paralelamente, como consequência direta da redução no número de transmissões de pacotes de controle, observou-se também uma diminuição no consumo energético associado ao envio de pacotes, variando, em média, entre 5% e 6%.

Entretanto, a redução do número de pacotes no sentido controlador-nó gerou um *trade-off* com o aumento na quantidade de pacotes de *flow request* enviados pelos nós sensores. Esse comportamento está diretamente associado ao atraso na configuração das rotas inerente ao uso de intervalos periódicos do MFS, o qual pode levar à expiração de temporizadores nos nós e gerar retransmissões.

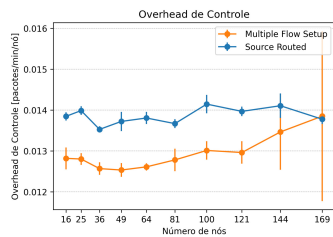


Fig. 4. Overhead de pacotes de controles.

Contudo, ao analisar o impacto líquido dessa troca, a Figura 4 demonstra que, mesmo com o aumento das requisições, o cenário com MFS reduziu o *overhead* de controle médio. Embora essa redução na média global seja sutil, ela é justificada pela própria composição do tráfego de pacotes de controle do protocolo: a diminuição de pacotes de *flow setup* acaba sendo diluída na estatística final, porque outros pacotes de controle, como os de descoberta de vizinhos (*neighbor discovery*), são gerados em maior quantidade e dominam a métrica. Por sua vez, a maior variância em redes maiores decorre do aumento da influência das perdas de pacotes de controle, que ocorrem de forma estocástica de acordo com a semente da simulação, consequentemente elevando a dispersão dos dados.

Dessa forma, conclui-se que a abordagem cumpre o objetivo de reduzir o tráfego de controle e mitigar o congestionamento, absorvendo o *trade-off* de forma vantajosa. Fica evidente, no entanto, que otimizações futuras, especialmente nos intervalos de configuração do controlador, podem extrair um potencial ainda maior do MFS.

IV. CONCLUSÕES

Este trabalho propôs e avaliou melhorias para o protocolo IT-SDN, com o objetivo de mitigar o consumo excessivo de recursos e o congestionamento da rede. Para isso, as melhorias foram concentradas em duas frentes principais: agregação de pacotes e a configuração de fluxos.

Os resultados experimentais demonstraram que o mecanismo de agregação e substituição de pacotes nas filas foi eficaz. A técnica garantiu taxas de entrega de pacotes superiores a 95% mesmo em cenários críticos, promovendo também uma redução no consumo de energia em topologias mais densas.

Paralelamente, a introdução do pacote de MFS para a configuração de fluxos mitigou o *overhead* do plano de controle, diminuindo em aproximadamente 41% a quantidade de pacotes de *flow setup* enviados pelo controlador e gerando economia energética adicional. Contudo, o estudo revelou um importante *trade-off*: o acúmulo de requisições no controlador gerou um descasamento temporal com os *timeouts* dos nós sensores, resultando em um aumento das retransmissões de *flow requests*. Essa constatação demonstra que, embora o mecanismo reduza drasticamente os pacotes de configuração de rota, seu potencial máximo depende da sincronia com o comportamento das requisições.

Dessa forma, de modo geral, conclui-se que as modificações propostas elevam a eficiência operacional do IT-SDN, tornando-o mais viável para implementações em larga escala. Como trabalhos futuros, sugere-se a investigação de políticas dinâmicas para a definição dos intervalos de envio do MFS, buscando o ponto de equilíbrio ideal entre a agilidade de resposta do controlador e a economia de recursos da rede.

REFERÊNCIAS

- [1] R. C. A. Alves, D. A. G. Oliveira, G. A. Nunez Segura, and C. B. Margi. The cost of software-defining things: A scalability study of software-defined sensor networks. *IEEE Access*, 7:115093–115108, 2019.
- [2] Renan Cerqueira Afonso Alves, Cintia Borges Margi, and Fernando A. Kuipers. Know when to listen: SDN-based protocols for directed IoT networks. *Computer Communications*, 150:672–686, 2020.
- [3] Renan Cerqueira Afonso Alves, Doriedson Alves Galdino de Oliveira, Gustavo Alonso Núñez Segura, and Cintia Borges Margi. IT-SDN: improved architecture for SDWSN. In *SBRC*. SBC, 2017.
- [4] Bruno Trevizan de Oliveira, Cintia Borges Margi, and Lucas Batista Gabriel. Tinsdn: Enabling multiple controllers for software-defined wireless sensor networks. In *LATINCOM*, pages 1–6, 2014.
- [5] Laura Galluccio, Sebastiano Milardo, Giacomo Morabito, and Sergio Palazzo. Sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks. In *INFOCOM*, 2015.
- [6] Hlabishi I. Kobo, Adnan M. Abu-Mahfouz, and Gerhard P. Hancke. A survey on software-defined wireless sensor networks: Challenges and design requirements. *IEEE Access*, 5:1872–1899, 2017.
- [7] Tarek Sayjari, Regina Melo Silveira, and Cintia Borges Margi. IEEE 802.15.4e TSCH traffic isolation approach impact on SDWSN's control plane performance. In *LATINCOM*, 2023.