

Um Mecanismo de Controle de Congestionamento Disparado por Eventos Através de Filtragem Local de Dados para RSSFs *Cluster-Tree*

Natanael Ribeiro*, Erico Leão*, Ricardo Rabêlo*, Vitor Vasconcelos* e Miguel Lino[§]

* Departamento de Computação, Universidade Federal do Piauí, Brasil

E-mail: natanrs@ufpi.edu.br, ericoleao@ufpi.edu.br, ricardoalr@ufpi.edu.br, vitor.vmv@ufpi.edu.br

[§] Pós-Graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina, Brazil

E-mail: miguelneto0019@gmail.com

Abstract—A Internet das Coisas (*IoT - Internet of Things*) é uma inovação tecnológica que tem se tornado bastante atrativa no âmbito comercial, industrial e residencial. Por conta disso, houve uma expansão no desenvolvimento de mecanismos de comunicação eficientes e desenvolvimento de aplicações em Redes de Sensores Sem Fio (RSSF). O conjunto de normas IEEE 802.15.4 e ZigBee forma uma pilha adequada de protocolos para implantação de RSSFs. Esse conjunto de normas especifica uma topologia especial *peer-to-peer* chamada *cluster-tree*, apontada como uma das topologias mais apropriadas para implantação de RSSFs de larga escala. Entretanto, a ocorrência de congestionamentos evidencia-se como um dos maiores desafios de redes *cluster-tree*, tornando necessário o desenvolvimento de mecanismos eficientes para mitigar este problema, sem comprometer a integridade dos dados. Este artigo propõe a utilização de um mecanismo local de filtragem de dados eficiente como forma de evitar ou postergar congestionamentos para RSSFs *cluster-tree*, evitando reconfigurações de rede e melhorando métricas importantes, tais como atrasos de comunicação fim-a-fim e perda de pacotes. Os resultados mostram que a implementação de mecanismos de filtragem local nos nós sensores pode fornecer qualidade de serviço para o fluxo de monitoramento, evitando a execução de mecanismos de controle e reconfiguração de redes.

Index Terms—Rede de Sensores sem Fio. IEEE 802.15.4. ZigBee. Cluster-tree. Filtragem de dados. Controle de congestionamento. Internet das Coisas.

I. INTRODUÇÃO

As Redes de Sensores Sem Fio (RSSFs) são redes *ad hoc* especiais compostas de um grupo de sensores e/ou atuadores autônomos interligados através de comunicação sem fio, destinados a monitorar condições físicas ou ambientais e passar seus dados de monitoramento de forma cooperativa para estações base [1]. Os nós sensores são pequenos dispositivos implantados no ambiente monitorado, comumente alimentados por baterias, e são equipados com um rádio transceptor, unidades de sensoriamento e micro-controladores [2]. Esses dispositivos são caracterizados por serem de baixo custo e se comunicarem com baixas taxas de transmissão e baixo consumo energético [3]. Essas particularidades possibilitam o desenvolvimento de diferentes domínios de aplicações de larga escala, tais como: monitoramento da saúde, transporte, casas inteligente, cidades inteligentes, automação industrial, agricultura de precisão, etc. [4].

Com o advento da Internet das Coisas (*Internet of Things - IoT*), um dos pilares da Quarta Revolução Industrial (*Industry 4.0*), as RSSFs têm se tornado uma área extremamente investigada pela sua grande adaptabilidade em aplicações de larga escala [5], [6]. Estas redes têm papel determinante em conectar sistemas e equipamentos para facilitar a alta personalização e automação de produtos, tornando o processo de produção em massa mais flexível e eficiente. Além de ser uma tecnologia adequada para ambientes inóspitos e de difícil acesso, estas redes fornecem dinamicidade, autonomia e flexibilidade no processo de conversação entre os dispositivos [7].

Existem na atualidade diferentes tecnologias que suportam o desenvolvimento e a implementação de soluções de RSSFs, cada uma possuindo suas próprias exigências, ambientes de programação e execução e instrumentos de desenvolvimento [2]. O conjunto de normas IEEE 802.15.4 [8] e Zigbee [9] têm se tornado um padrão *de facto* que permite a comunicação sem fio entre dispositivos de controle e sensores [10]. O padrão IEEE 802.15.4 especifica a camada física e a subcamada de Controle de Acesso ao Meio (MAC - *Medium Access Control*) para aplicações *Low-Rate Wireless Personal Area Network* (LR-WPAN), enquanto o ZigBee define as camadas superiores (Rede e Aplicação) [11].

O conjunto de normas IEEE 802.15.4/ZigBee suporta um tipo especial de rede *peer-to-peer* conhecido como *cluster-tree*, apontado como uma das topologias mais adequadas para implementar RSSFs em larga escala [12], que oferece recursos, tais como: alta escalabilidade, operação de baixo consumo energético e comunicação com baixa latência. Entretanto, devido à sua estrutura hierárquica e ao padrão do tráfego de monitoramento (*convergecast*) gerado pelos nós sensores, essas redes enfrentam típicos problemas de congestionamentos, ocasionando um maior consumo de energia, maior perda de pacotes e maiores atrasos de comunicação fim-a-fim [13], [14].

Diante deste contexto, este artigo propõe um mecanismo eficiente de controle de congestionamento disparado por eventos através de filtragem local de dados para RSSFs *cluster-tree*. Este mecanismo objetiva evitar ou postergar congestionamentos, o que implica a necessidade de reconfiguração de rede, combinando os seguintes esquemas: i) a implementação de

uma arquitetura de reconfiguração de rede já existente baseada nas características de geração de dados dos nós sensores, definida em [13]; ii) o monitoramento contínuo para detecção de eventos de aumento de fluxo de dados, tais como alarmes, que potencializa os congestionamentos; iii) implementação do mecanismo de filtragem de dados para prover uma diminuição do fluxo dos nós sensores sem perder relevância dos dados; e por fim, iv) disparar o processo de reconfiguração de rede, caso o mecanismo de filtragem de dados não consiga suprir o aumento da carga de dados gerada pelo evento.

O restante deste artigo está organizado como segue: Seção II apresenta o conjunto de normas IEEE 802.15.4/ZigBee e redes *cluster-tree*. Seção III sumariza uma visão sobre congestionamentos de rede e o processo de compressão de dados. Seção IV apresenta os trabalhos mais relevantes para o mecanismo proposto neste artigo. Na Seção V, o mecanismo proposto de filtragem local dos dados é apresentado, detalhando seus principais esquemas. A Seção VI apresenta os resultados obtidos através de avaliação por simulação. Finalmente, a Seção VII apresenta as conclusões e trabalhos futuros.

II. REDES DE SENSORES SEM FIO CLUSTER-TREE

As redes *cluster-tree* são um caso especial de topologia *peer-to-peer*, onde os nós sensores são organizados em agrupamentos (*clusters*), coordenados por um nó especial chamado de coordenador do *cluster* (*Cluster-Head – CH*). Os CHs são responsáveis por sincronizar a comunicação com seus nós filhos associados e centralizar as atividades de comunicação dentro dos *clusters*. A Figura 1 ilustra uma rede *cluster-tree* baseada no padrão IEEE 802.15.4.

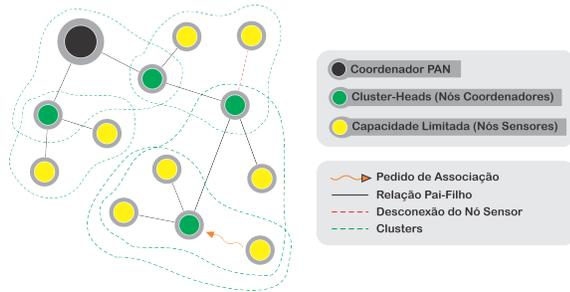


Fig. 1. RSSFS *cluster-tree* baseadas no padrão IEEE 802.15.4.

O processo de formação da rede é iniciado por um nó exclusivo chamado de coordenador PAN, que atua como coordenador da rede (e de seu próprio *cluster*) e é responsável por todas as atividades de gerenciamento. Este processo pode ser executado recursivamente ao longo dos *clusters*, aumentando, assim, a cobertura de rede. Por fim, os *clusters* são interconectados através de seus nós coordenadores, formando assim uma estrutura *multiclusters* hierárquica. Esses múltiplos *clusters* podem implementar ciclos de trabalho reduzidos, permitindo dessa forma uma economia de energia dos seus nós associados.

As redes *cluster-tree* operam no modo de comunicação com *beacons*, em que a comunicação dentro de um *cluster* é organizada com base numa estrutura chamada *superframe*.

Um *superframe* é limitado por quadros *beacons* que são periodicamente transmitidos pelos CHs, a fim de sincronizar seus nós sensores filhos e identificar o PAN [13].

Basicamente, a estrutura *superframe* é definida através de dois parâmetros: *macBeaconOrder* (BO) e *macSuperframeOrder* (SO). Esses parâmetros determinam o intervalo de quadros *beacons* (*Beacon Interval – BI*) e a duração de *superframe* (*Superframe Duration – SD*), respectivamente. O BI diz respeito ao intervalo no qual um coordenador de *cluster* (CH) deve enviar periodicamente seus quadros de *beacons*. Por sua vez, o SD estabelece o período de comunicação ativa dos *clusters*. O BI e SD são definidos como segue:

$$\begin{aligned} BI &= aBaseSuperframeDuration \times 2^{BO}, & 0 \leq BO \leq 14 \\ SD &= aBaseSuperframeDuration \times 2^{SO}, & 0 \leq SO \leq BO \end{aligned} \quad (1)$$

onde o valor 14 é o limite determinado pelo IEEE 802.15.4 (sendo BO = 15 a representação do modo *non-beacon*) e a variável *aBaseSuperframeDuration* representa a duração mínima de um *superframe* quando SO é igual a zero (tendo em vista um rádio padrão, esta variável corresponde a 960 símbolos, que equivale a uma duração de 15,36 ms, com cada símbolo sendo composto por 4 bits e considerando uma banda de frequência de 2.4 GHz e taxas de transmissão de até 250 kbps) [15]. A Figura 2 mostra a estrutura do *superframe*.

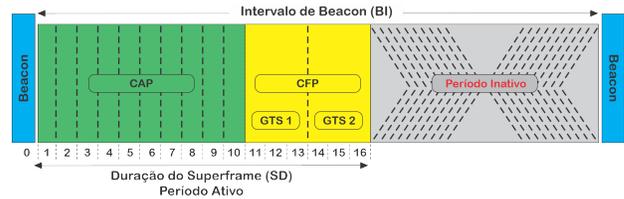


Fig. 2. A estrutura *superframe* do padrão IEEE 802.15.4.

É importante mencionar que o BI possui uma porção, opcionalmente, inativa e outra porção ativa. Dessa forma, quando o SO é menor que o BO, significa que existe uma parte inativa e nós sensores poderão entrar em modo de economia de energia. Quando o BO é igual ao SO, não existe porção inativa e os dispositivos não possuem tempo adicional para economizar energia.

A porção ativa do *superframe* inicia-se imediatamente após o quadro de *beacon* transmitido pelo coordenador, estabelecendo o período em que os nós, tanto coordenadores quanto nós sensores, trocam mensagens. Esta porção é subdividida em dois períodos: Período de Acesso com Contenção (*Contention Access Period – CAP*) e o Período Livre de Contenção (*Contention Free Period – CFP*).

Durante o CAP, os nós sensores disputam pelo canal sem fio utilizando o algoritmo de acesso ao meio CSMA-CA (*Carrier Sense Multiple Access – Collision Avoidance*) com o objetivo de evitar colisões. Já o CFP é opcional e, uma vez solicitado, permite que o coordenador faça reserva de *slots* de tempo garantido (*Guaranteed Time Slot – GTS*) com a intenção de que um nó tenha acesso ao canal de forma dedicada.

Em redes *cluster-tree* existem, particularmente, dois tipos de tráfego de dados: tráfego de controle (*downstream*) e tráfego de monitoramento (*upstream*). O padrão de tráfego mais comum em RSSFs é o tráfego *upstream*, sendo preponderante em aplicações de monitoramento em larga-escala e automação, e condiz ao típico tráfego de dados gerado pelos nós sensores, nas quais mensagens são encaminhadas a um nó especial chamado de *sink* (estação base). O *sink* tem como função a coleta de dados dos nós sensores e é responsável pela ligação da RSSF com as redes externas (por exemplo, a Internet e aplicações IoT). Em aplicações de monitoramento, as mensagens são encaminhadas em direção ao nó *sink*, nas quais atravessam vários *clusters* por meio dos CHs. Habitualmente, o coordenador PAN é utilizado como nó *sink*, por ser o nó raiz e poder ser dotado de maior capacidade de recursos.

Por sua vez, o tráfego *downstream* é caracterizado pelos comandos gerados pelos nós coordenadores, com o propósito de permitir atividades de controle e/ou reconfigurações de rede. É importante mencionar que o tráfego de monitoramento segue um comportamento *bottom-up* (a partir dos nós mais profundos da rede em direção ao nó *sink*), enquanto o tráfego de controle segue um caminho inverso, direcionado a partir dos nós coordenadores para outro nós coordenadores ou nós finais da rede (comportamento *top-down*).

III. CONGESTIONAMENTO E COMPRESSÃO DE DADOS

A. Congestionamento

Além de eventuais problemas resultantes da fase de implantação dos nós sensores, como por exemplo, variações espaciais (usualmente, devido a ações relativas ao ambiente) e falhas temporais ou por limitações impostas pela condição de reserva energética [16], uma RSSF, como um todo, é desafiada por problemas típicos de comunicação. Deste modo, influenciando negativamente o desempenho de RSSFs, o problema de congestionamento tem se tornado uma temática pertinente em redes de larga escala [17]. Tal acontecimento se deve, sobretudo, ao alto fluxo de dados gerados pelos nós sensores, que transitam pela rede em direção ao nó *sink* [18], embora este problema também ocorra no fluxo de controle (*downstream*), durante operações de controle ou reconfiguração de rede. Portanto, o congestionamento tem uma implicação significativa nos parâmetros de qualidade de serviço (*Quality of Service* – QoS), como por exemplo, taxa de entrega de pacotes (*Packet Delivery Ratio* – PDR), atrasos de comunicação fim-a-fim e consumo de energia dos nós sensores [19]. Assim, para lidar com tais desafios e problemas em RSSFs, é necessário considerar e controlar o fator de congestionamento.

Os mecanismos para controle de congestionamento são categorizados em: contínuo, baseado em demanda, disparado por eventos e híbrido [20]. No mecanismo contínuo, a atuação é realizada pelo nó *sink*, de forma contínua, através do monitoramento de parâmetros atuais da rede, tais como ocupação de *buffers* e colisões de pacotes. No método baseado em demanda, os ajustes são calculados e armazenados pelo coordenador a cada salto, enviando quadros de controle somente em caso de pedidos do nó *sink* para atuação na rede. Na abordagem

disparada por eventos (utilizada neste artigo), é seguida basicamente três etapas: detecção do evento, notificação e reajuste. Por último, o método híbrido pode combinar um ou mais métodos simultaneamente.

B. Compressão de Dados

A Compressão de Dados (CD) é o processo no qual uma informação é representada de maneira compactada. A utilização dos mecanismos de CD diminui o tamanho da informação que deve ser transmitida e armazenada, o que é importante em aplicações que requerem um menor consumo de seus recursos, tais como o sistema de armazenamento, largura de banda e o tempo de ocupação de um sistema de comunicação, a energia usada na transmissão e transporte de dados, entre outros [21].

De uma forma geral, o processo de compressão pode ser classificado em: com perdas ou sem perdas. Na primeira abordagem, os dados originais não podem ser totalmente recuperados após a compressão. Como resultado, o tamanho do arquivo é permanentemente reduzido, eliminando dados redundantes. Em contraste, na abordagem sem perdas, todos os dados originais compactados poderão ser recuperados após a descompressão [22].

O trabalho proposto neste artigo utiliza um mecanismo de compressão com perdas baseado no algoritmo *Swinging Door Trending* (SDT). SDT é um método de CD que usa uma linha de tendência para representar uma certa quantidade de amostras. Esse algoritmo de compressão com perda tem como parâmetros mais importantes o Desvio de Compressão (DC), que representa a diferença máxima que um certo ponto pode ter com relação a tendência atual da informação [23].

IV. TRABALHOS RELACIONADOS

A fim de fornecer uma redução da quantidade de dados enviados, os autores em [24] propuseram um esquema com perdas baseado no *Swinging Door Trending*. Neste trabalho, foi criada uma nova métrica de compressão que utiliza uma tendência linear para representar uma série de amostras e permitir que os dispositivos IoT autodeterminem adequadamente o valor do desvio de compressão, parâmetro mais importante, usado localmente para compactar sua quantidade de dados.

Um mecanismo utilizando balanceamento de carga foi proposto pelos autores em [25], no qual um algoritmo de balanceamento de carga de tamanho de *cluster* é desenvolvido para atingir a utilização ideal do método *Compressive Sensing* – técnica de processamento que obtém e reconstrói eficientemente um sinal – em uma rede de sensores baseada em IoT. O algoritmo proposto consiste em três fases: configuração, balanceamento de carga do tamanho do *cluster* e compressão de dados. Sua metodologia reduz o número total de transmissões e melhora a reconstrução do processo de criação otimizando a matriz de sensoriamento de compressão, o consumo geral de energia, total de número de dados transmitidos e erro de reconstrução. Todavia, os autores não mencionam se, após a compactação, os dados terão perdas que podem ser significativas a depender das características dos dados contidos no sinal.

Já o trabalho proposto por [26] baseia-se num algoritmo sem perdas de compressão *Lempel-Ziv-Welch* (LZW). A compressão LZW funciona lendo uma sequência de símbolos, agrupando os símbolos em *strings* e convertendo-os em códigos. A ideia principal é a construção de um dicionário de símbolos ou palavras conforme o texto ou a informação vai sendo processada pelo algoritmo, de forma que o dicionário vai criando novas palavras a partir de combinações de pelo menos 2 caracteres encontrados no sinal. Entretanto, esse método possui algumas limitações ou deficiências no modo de codificação, entre elas a capacidade de armazenamento limitada. Portanto, esse algoritmo não é adequado em situações que envolvam uma grande quantidade de dados.

Em [27], os autores propuseram o desenvolvimento de um algoritmo leve para compressão de dados baseado na codificação delta, cuja compressão é alcançada após a captura da diferença de leitura de um dado do sensor e seu predecessor. Por sua vez, os autores em [28] utilizaram a compressão de dados sem perdas baseado em codificação *Huffman*, para substituir dados redundantes por identificadores, utilizando Matriz de Adjacência. A ideia principal é que, ao encontrar a ocorrência comum de padrões de nós e links, seja realizada a substituição por identificadores, reduzindo assim a quantidade de dados a serem armazenados e transmitidos.

Por fim, os autores em [29] propuseram a implementação de *Run Length Encoding* (RLE) e técnicas de compressão sem perdas Delta, que precisam de um menor tempo para compressão de dados. Os resultados apresentados mostram que a combinação da técnica Delta e RLE alcança melhores taxas de compressão do que apenas a técnica de compressão Delta. No entanto, a eficácia da técnica diminui se ocorrer muitos picos de variabilidade dos dados capturados pelos sensores.

Diferente dos trabalhos supracitados, este artigo apresenta um mecanismo dinâmico que implementa filtragem local nos nós sensores de acordo com eventos disparados pelo ambiente monitorado, a fim de reduzir a quantidade de dados a serem transmitidos em redes *cluster-tree*, sem perda de relevância e baixo processamento, diminuindo assim atrasos de comunicação fim-a-fim, perda de pacotes, congestionamentos e reconfigurações de rede.

V. MECANISMO DE CONTROLE DE CONGESTIONAMENTO DISPARADO POR EVENTOS ATRAVÉS DE FILTRAGEM LOCAL DE DADOS

Este artigo propõe um mecanismo eficiente de filtragem local de dados baseado na detecção de eventos, com o objetivo de evitar (ou postergar) congestionamentos e reconfigurações de RSSFs *cluster-tree* de larga escala baseadas nos padrões IEEE 802.15.4 e ZigBee.

Este trabalho utiliza uma infraestrutura de reconfiguração dinâmica de redes *cluster-tree* proposta por [13], que dispara um mecanismo de reconfiguração de rede na detecção de eventos críticos na rede, tais como dados monitorados indicando a detecção de uma possível situação de desastre, o que gera um aumento no ciclo de trabalho dos nós sensores e uma sobrecarga adicional para a infraestrutura de comunicação.

Entretanto, o processo de reconfiguração é custoso para uma RSSF, o que motiva o desenvolvimento de mecanismos locais eficientes que possam evitar ou postergar essas atividades.

A Figura 3 apresenta o modelo arquitetural de reconfiguração global existente (cor cinza), com a inclusão do mecanismo de atuação local proposto (cor azul).

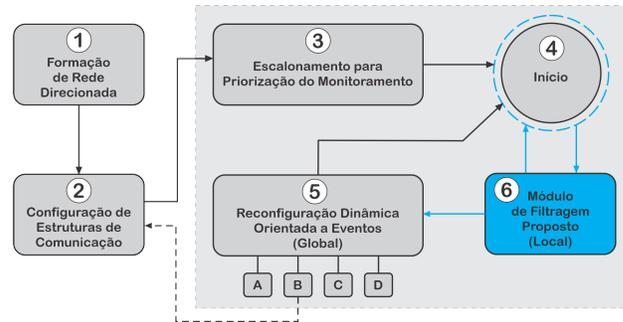


Fig. 3. Arquitetura de reconfiguração global e o mecanismo de filtragem proposto para RSSFs *cluster-tree*.

Basicamente, os módulos estabelecidos segundo [13] são definidos como segue:

- Módulo 1: Processo de formação da rede orientado a geração de dados dos nós sensores;
- Módulo 2: Configuração eficiente das estruturas de comunicação de acordo com [12];
- Módulo 3: Escalonamento para priorização do tráfego de monitoramento (*upstream*);
- Módulo 4: Funcionamento operacional da rede com a finalidade de detectar possíveis eventos de modificação de carga de dados;
- Módulo 5: Mecanismo de reconfiguração dinâmica da rede orientada a eventos em caso de congestionamento, no qual possui os submódulos A (Detecção do Evento de Carga), B (Cálculo da Reconfiguração da Rede), C (Criação da Janela de Oportunidade) e D (Envio das Mensagens de Reconfiguração aos Nós Envolvidos).

Este trabalho propõe a implementação do módulo de filtragem de dados (módulo 6), que em caso de sobrecarga na rede, atua sobre os dados gerados pelos nós sensores, fornecendo filtragem até a normalização do fluxo, evitando ou postergando uma possível reconfiguração na rede.

A. Novo Módulo de Filtragem de Dados Disparado por Eventos

O objetivo deste novo módulo é criar um mecanismo adaptativo que possa monitorar o fluxo de dados gerados pelos nós sensores e atuar localmente entre os módulos 4 (Funcionamento operacional) e 5 (Mecanismo de reconfiguração) da infraestrutura existente, de tal forma que essa atuação seja transparente ao coordenador PAN. O mecanismo é ativado até o aumento do fluxo de dados gerado pelo evento crítico possa ser normalizado, ou seja, a infraestrutura de comunicação consiga dar vazão ao tráfego gerado pelos nós sensores sem sobrecarga dos nós coordenadores.

Este trabalho utiliza um mecanismo de filtragem de dados baseados no algoritmo *Swinging Door Trending* (SDT), pela eficácia comprovada e ser bastante utilizado em RSSFs [24], [30]. O processo de detecção de eventos de carga e o mecanismo de filtragem proposto são descritos a seguir.

1) *Detecção do Evento de Carga*: Conforme apontado por [13], os eventos de carga (eventos críticos) ocorrem esporadicamente durante o sensoriamento constante dos nós sensores, quando uma taxa diferente da usual é transmitida. O modelo de rede proposto considera que os nós sensores detectem alterações originadas de distintos fenômenos físicos e passem a transmitir fluxos de dados com uma periodicidade maior para seus CHs.

2) *Processo de Filtragem SDT*: A proposta é reduzir a quantidade de dados enviados aos CHs fazendo uma filtragem pelo SDT na medida que ocorrem os eventos críticos. Para isso, o mecanismo define uma janela de compressão baseada no primeiro valor capturado pelo nó sensor (Valor de Referência – VR) e configura um limiar (*threshold*) de relevância dos dados. A partir desse ponto, os próximos dados gerados pelos nós sensores são avaliados continuamente pelo SDT. Se os dados estiverem dentro da janela de compressão, respeitando o tempo mínimo (t_{min}) e máximo (t_{max}), ele será previamente descartado. Os dados que ficarem fora da janela de compressão poderão ser selecionados para serem enviados, desde que estejam delimitado pelos tempos t_{min} e t_{max} . O objetivo do t_{min} é filtrar possíveis *outliers*, enquanto que t_{max} é definido para evitar que nós sensores deixem de enviar dados com valores próximos. Após selecionar um novo dado para ser enviado pelo nó sensor, o algoritmo define uma nova janela de compactação, considerando o valor enviado como o próximo VR. O algoritmo é disparado sempre que houver uma detecção de um evento crítico e, naturalmente, o aumento do fluxo de dados por parte dos nós sensores.

O processo é repetido ciclicamente a fim de obter uma menor taxa de amostragem dos dados, sem comprometer sua relevância, a fim de evitar o procedimento de reconfiguração global por parte do coordenador PAN. Se, mesmo aplicando o mecanismo proposto, a carga imposta pelos dados superar a capacidade das estruturas de comunicação [12], o processo de reconfiguração global é disparado. A Figura 4 ilustra o processo de filtragem e os principais parâmetros utilizados.

VI. AVALIAÇÃO DE DESEMPENHO

O mecanismo proposto nesse trabalho foi avaliado através de simulação, utilizando o CT-SIM [31], um conjunto de modelos de simulação que implementa as principais funcionalidades de redes *cluster-tree*. CT-SIM é baseado no *Castalia* [32], um simulador de eventos discretos *open-source* para RSSFs, desenvolvido pela *National ICT Australia* (NICTA) e baseado na plataforma OMNet++.

Para esta avaliação por simulação, é considerado um ambiente variando os cenários em relação ao número de nós: 101, 151 e 201 nós sensores (sendo um nó o coordenador PAN). Para todos os cenários, o coordenador PAN está localizado na posição central do cenário (100m x 100m), enquanto os nós

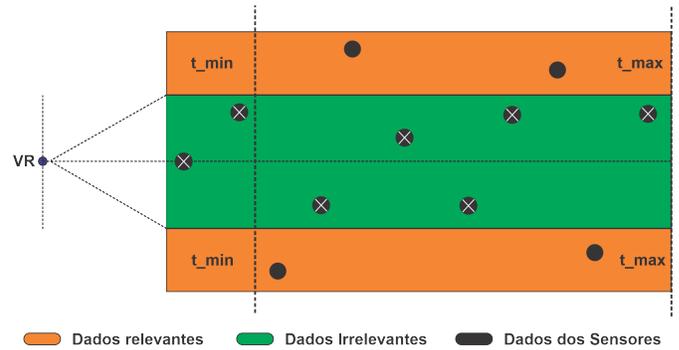


Fig. 4. Representação ilustrativa do algoritmo SDT e seus principais parâmetros.

restantes foram implantados aleatoriamente ao longo de um ambiente de 200m x 200m.

Com relação ao tráfego de monitoramento, foi utilizada uma base de dados real, contendo 10.000 registros oriundos de um sistema de monitoramento em tempo real de distribuição de gás, especificamente uma variável de vazão, que simula a captura de dados pelos nós sensores. Cada nó sensor envia os seus dados em direção ao nó *sink* (coordenador PAN) com uma periodicidade de 1 pacote a cada 20 segundos, seguindo as regras de comunicação hierárquica de redes *cluster-tree* baseadas nos padrões IEEE 802.15.4/ZigBee. É importante salientar que os CHs não implementam nenhuma operação de agregação ou fusão de dados, o que implica que todos os dados são encaminhados para o coordenador PAN, usando um esquema de escalonamento *bottom-up*.

Com relação à ocorrência dos eventos, foi definida uma área do ambiente onde ocorre os eventos críticos. Assim, apenas nós sensores localizados na área delimitada pelo eixo x de 45m a 145m e pelo eixo y de 0m a 45m (uma área em torno de 10% do ambiente total) geram eventos críticos.

Além disso, foi definido empiricamente que a faixa de valores de vazão considerados de operação normal varia de 600 a 650. Dessa forma, qualquer valor fora dessa faixa é considerado a ocorrência de um evento crítico (desde que o nó sensor esteja localizado na zona de eventos), onde a sua periodicidade é aumentada em uma multiplicidade de 10 vezes (ou seja, 1 pacote a cada 2 segundos). Além disso, o limiar utilizado para a simulação foi de 10% do valor de referência. A Figura 5 ilustra o banco de dados reais utilizado e quando existe ou não a ocorrência de um evento crítico (zonas em vermelho e verde, respectivamente).

Para fins de avaliação, as seguintes métricas de desempenho foram utilizadas:

- Atrasos de comunicação fim-a-fim: intervalo de tempo entre a geração de um dado por um nó sensor origem e sua recepção pelo nó *sink* (coordenador PAN).
- Taxa de perda de pacotes: porcentagem de pacotes perdidos durante o período de comunicação, considerando todas as mensagens descartadas devido à falta de espaço nos *buffers*, colisões e/ou falhas de transmissão;

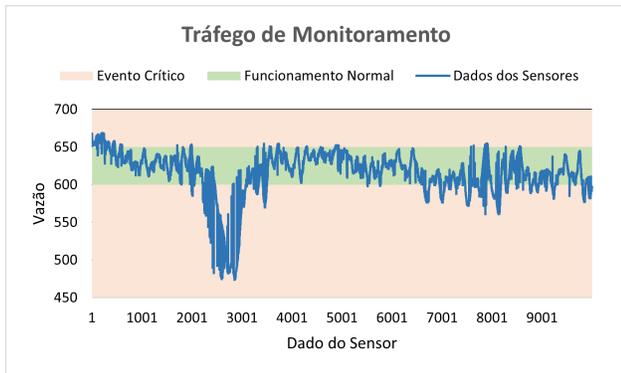


Fig. 5. Representação dos dados dos sensores e zonas de evento crítico (zona vermelha) e de funcionamento normal (zona verde).

- Taxa de compressão: taxa efetiva de filtragem de dados alcançada com o mecanismo implementado.

Cada cenário foi simulado 15 vezes, com um nível de confiança de 95% e os resultados médios foram plotados em gráficos. A tabela I apresenta os principais parâmetros de configuração utilizados neste estudo de simulação.

TABLE I
CONFIGURAÇÃO DOS PRINCIPAIS PARÂMETROS DE SIMULAÇÃO.

Descrição	Valor
Área do ambiente	200m x 200m
Número de nós sensores (exceto PAN)	100 / 150 / 200
Faixa normal de valores de dados	600 a 650
Zona de eventos críticos	(45m à 195m; 0m à 45m)
Periodicidade fora do evento	1 pacote a cada 20 segundos
Periodicidade durante um evento	1 pacote a cada 2 segundos
Limiar de relevância de dados	10% o valor de referência
Modelo do rádio	Chipcon CC2420

VII. RESULTADOS E DISCUSSÕES

A Figura 6 ilustra os atrasos médios de comunicação fim-a-fim dos nós pertencentes à zona de eventos, incluindo os pacotes gerados durante a ocorrência dos eventos críticos e fora dos eventos críticos.

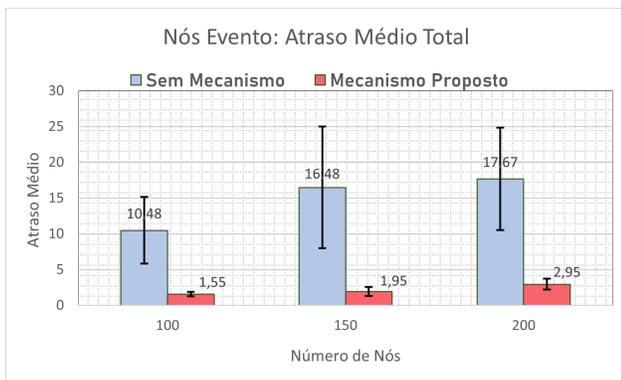


Fig. 6. Atraso médio de comunicação fim-a-fim dos nós pertencentes à zona de eventos críticos.

Como pode ser observado, é possível verificar o impacto direto que o mecanismo de filtragem local nos nós impõe sobre os atrasos de comunicação. Como esperado, os atrasos de comunicação são bem maiores quando não se aplica o mecanismo de filtragem, em todos os cenários avaliados. Isso se dá pela alta taxa de compactação obtida (aproximadamente 79,3%) com o mecanismo implementado, o que filtra valores aproximados (irrelevantes) e diminui consideravelmente a quantidades de dados a serem trafegados pela rede.

Com o objetivo de fazer uma avaliação mais refinada, as Figuras 7 e 8 ilustram os atrasos médios dos nós pertencentes à zona de eventos críticos durante a ocorrência do evento crítico (quando os valores gerados são maiores que 650 ou menores que 600) e fora do evento crítico, respectivamente.

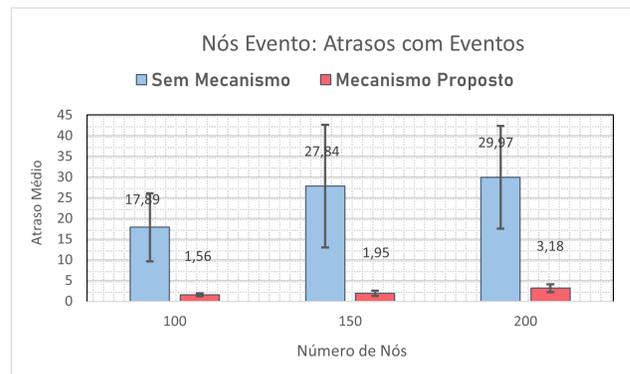


Fig. 7. Atraso médio de comunicação fim-a-fim dos nós pertencentes à zona de eventos críticos, durante a ocorrência dos eventos.

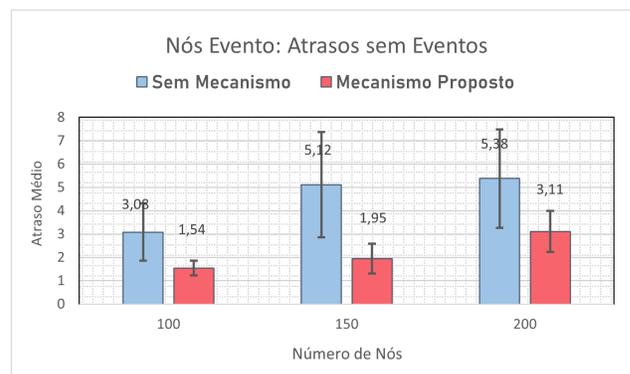


Fig. 8. Atraso médio de comunicação fim-a-fim dos nós pertencentes à zona de eventos críticos, fora da ocorrência dos eventos.

Note que existe uma diferença relevante dos resultados de atraso médio do mecanismo proposto, nos três cenários simulados. Os nós sem filtragem levam muito mais tempo para entregar os seus pacotes, devido a maior quantidade de dados gerados durante o evento. Isso se dá porque os pacotes tendem a se acumularem nos *buffers* dos nós coordenadores, aumentando assim os atrasos de comunicação. Esse efeito de acúmulo nos *buffers* contribuiu também para que os pacotes gerados fora do evento crítico também tenham seus atrasos

aumentados consideravelmente. Esse é o típico cenário de congestionamentos em RSSFs hierárquicas.

Por sua vez, a Figura 9 ilustra as taxas médias de perda de pacotes totais dos nós pertencentes à zona de eventos, incluindo os pacotes gerados durante a ocorrência dos eventos críticos e fora dos eventos críticos.

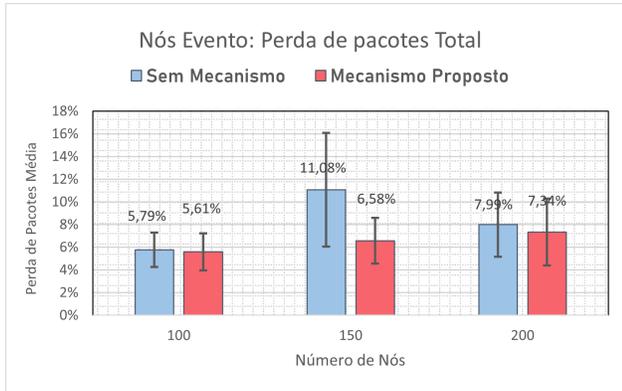


Fig. 9. Taxa média de perda de pacotes total dos nós pertencentes à zona de eventos críticos.

Como pode ser observado, os resultados mostram que os valores médios são levemente menores para o mecanismo proposto, nos cenários considerados (especialmente de 150 nós sensores), chegando a um cenário de estabilidade considerando a variabilidade dos dados. Isso se dá porque a taxa de perda de pacotes é fortemente impactada pelo descarte de pacotes nos *buffers* dos coordenadores (quando cheios) e pelas colisões na transmissão dos pacotes. Com relação às colisões, a taxa de perda de pacotes tende a aumentar quando se aumenta o número de nós transmissores (não é o caso das simulações). Com relação ao descarte de pacotes, a taxa de perda de pacotes aumenta significativamente quando a disponibilidade de *slots* nos *buffers* dos coordenadores é reduzida.

Nesse estudo de simulação, o tamanho dos *buffers* é definido de acordo com o algoritmo *Load-SDA* [12], que considera a carga total imposta pelos nós filhos e descendentes. Assim, mesmo com o aumento da periodicidade dos dados, os *buffers* dos nós coordenadores conseguem armazenar a quantidade máxima de dados gerados durante o período CAP de cada CH. Por esse motivo, os resultados das taxas de perda de pacotes não foram fortemente impactadas. As Figuras 10 e 11 ilustram, de forma refinada, as taxas médias de perda de pacotes dos nós pertencentes à zona de eventos críticos, durante a ocorrência do evento crítico e fora do evento crítico, respectivamente.

Como pode-se observar, durante o acontecimento do evento, com o aumento da taxa de geração dos dados, a tendência é ter um leve aumento da perda de pacotes, quando comparados com o período fora do evento crítico.

Complementarmente, também foi realizado uma avaliação dos atrasos médios de comunicação e das taxas médias de perda de pacotes dos nós que estão localizados fora da zona de eventos. Os resultados mostraram que, tanto no cenário com o uso e sem o uso do mecanismo proposto, as médias das

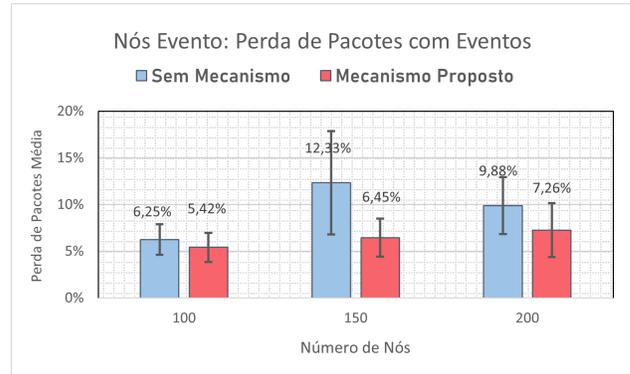


Fig. 10. Taxa média de perda de pacotes total dos nós pertencentes à zona de eventos críticos, durante a ocorrência do eventos.

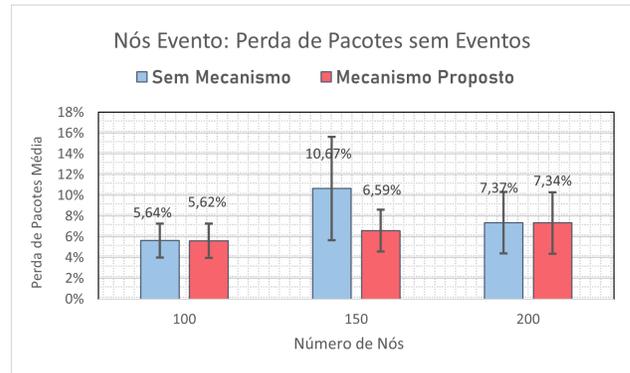


Fig. 11. Taxa média de perda de pacotes total dos nós pertencentes à zona de eventos críticos, fora da ocorrência dos eventos.

métricas de avaliação são levemente menores com a aplicação do mecanismo de filtragem, e os resultados são considerados semelhantes (levando em conta a variabilidade dos dados). Isso mostra que os nós sensores fora da zona de eventos críticos não são impactados pelo mecanismo proposto, pois mantém suas periodicidades constantes durante toda a simulação.

Por fim, foi avaliada a capacidade do mecanismo de filtragem proposto de evitar a ativação do procedimento de reconfiguração global da rede. Para isso, a cada ocorrência do evento crítico, foi calculada a taxa efetiva de geração de pacotes (considerando a taxa de filtragem dos dados) e realizado os cálculos para definição das estruturas de comunicação (*superframes*) de cada *cluster* da rede, com base no *Load-SDA* [12]. Como resultados, considerando as características do evento crítico definido neste estudo de simulação, houveram 149 eventos críticos em cada cenário simulado. Com a aplicação do mecanismo proposto, não foi necessário a realização de nenhum procedimento de reconfiguração global da rede; de forma contrária, a abordagem que não utiliza o mecanismo precisaria ativar o procedimento de reconfiguração de rede em todos os cenários simulados.

VIII. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo propôs um mecanismo de controle de congestionamento disparado por eventos através da implementação de

mecanismo de filtragem local de dados nos nós sensores para RSSFs *cluster-tree* baseadas no padrão IEEE 802.15.4. A ideia principal foi desenvolver um módulo de compressão a fim de diminuir a quantidade de dados a serem transmitidos pela rede, reduzindo assim congestionamentos, atrasos de comunicação fim-a-fim, perdas de pacotes e a necessidade de procedimentos de reconfiguração das estruturas de comunicação.

Os resultados mostraram que é possível reduzir consideravelmente os atrasos de comunicação fim-a-fim e perda de pacotes dos fluxos de alta periodicidade, tipicamente gerados em ambientes de eventos críticos, através da implementação de mecanismos de filtragem de dados, que descarta o envio de dados redundantes, irrelevantes ou *outliers*. O mecanismo proposto se mostrou eficiente para diluir congestionamentos de rede e a necessidade de reconfigurações de parâmetros de comunicação, que tendem a gastar mais energia e interferir sobre os principais fluxos de dados dos nós sensores.

Como trabalhos futuros, pretende-se otimizar o mecanismo de filtragem de dados através da implementação de um mecanismo de compactação de dados redundantes nos nós coordenadores e o desenvolvimento de um protótipo real.

FINANCIAMENTOS

Este trabalho foi parcialmente financiado pelo projeto FAPEPI/MCT/CNPq/CT-INFRA nº 007/2018 e pelo PQ UFPI (nº 05/2021/PROPESQI/PRPG/UFPI).

REFERENCES

- [1] K. Yang, *Wireless sensor networks*. Springer, 2014.
- [2] T. C. Rodrigues, P. V. Dantas, F. C. Delicato, P. F. Pires, C. Miceli, and L. Pirmez, "Using mda for building wireless sensor network applications," in *2010 Fourth Brazilian Symposium on Software Components, Architectures and Reuse*. IEEE, 2010, pp. 110–119.
- [3] J. A. Stankovic, A. D. Wood, and T. He, "Realistic applications for wireless sensor networks," in *Theoretical aspects of distributed computing in sensor networks*. Springer, 2011, pp. 835–863.
- [4] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.
- [5] A. O. Adebayo, M. S. Chaubey, and L. P. Numbu, "Industry 4.0: The fourth industrial revolution and how it relates to the application of internet of things (IoT)," *Journal of Multidisciplinary Engineering Science Studies (JMESS)*, vol. 5, no. 2, 2019.
- [6] R. A. Khalil, N. Saeed, M. Masood, Y. M. Fard, M.-S. Alouini, and T. Y. Al-Naffouri, "Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications," *IEEE Internet of Things Journal*, 2021.
- [7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [8] IEEE, "Ieee 802.15. 4-2015-IEEE standard for low-rate wireless networks," *IEEE standards association*, 2015.
- [9] ZigBee, "Zigbee specification overview," Tech. Rep. 10, 2012.
- [10] D. V. d. Queiroz *et al.*, "Simulação realista de redes de sensores sem fio industriais," Master's thesis, Universidade Federal da Paraíba, 2016.
- [11] E. L. F. dos Santos, "A IEEE 802.15.4 como plataforma de comunicação de dados," *Revista Ilha Digital*, vol. 4, pp. 97–105, 2013.
- [12] E. Leão, C. Montez, R. Moraes, P. Portugal, and F. Vasques, "Superframe duration allocation schemes to improve the throughput of cluster-tree wireless sensor networks," *Sensors*, vol. 17, no. 2, p. 249, 2017.
- [13] M. Lino, V. Vasconcelos, E. Leão, A. Soares, C. Montez, R. Moraes, and F. Vasques, "Mecanismo eficiente para disseminação do tráfego de controle em redes de sensores sem fio cluster-tree baseadas no IEEE 802.15. 4," in *Anais Estendidos do IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais*. SBC, 2019, pp. 1–8.
- [14] S. R. Barath, B. Usha, G. Arpitha, V. Kumar, and S. Yaseen, "Strategies for congestion control in wireless sensor networks: A survey," in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, 2019, pp. 629–633.
- [15] A. Cunha, A. Koubaa, R. Severino, and M. Alves, "Open-zb: an open-source implementation of the IEEE 802.15. 4/zigbee protocol stack on tinys," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*. IEEE, 2007, pp. 1–12.
- [16] A. Kiraza and M. Çakıroğlu, "A survey of congestion control protocols providing energy conservation in wireless sensor networks," *Turkish Journal of Engineering*, vol. 1, pp. 12–22, 2014.
- [17] A. D. Rathnayaka and V. M. Potdar, "Wireless sensor network transport protocol: A critical review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 134–146, 2013.
- [18] X. Yin, X. Zhou, R. Huang, Y. Fang, and S. Li, "A fairness-aware congestion control scheme in wireless sensor networks," *IEEE transactions on vehicular technology*, vol. 58, no. 9, pp. 5225–5234, 2009.
- [19] A. Ghaffari, "Congestion control mechanisms in wireless sensor networks: A survey," *Journal of network and computer applications*, vol. 52, pp. 101–115, 2015.
- [20] M. A. Jan, S. R. U. Jan, M. Alam, A. Akhuzada, and I. U. Rahman, "A comprehensive analysis of congestion control protocols in wireless sensor networks," *Mobile networks and applications*, vol. 23, no. 3, pp. 456–468, 2018.
- [21] K. P. Shrivana and D. S. V. Veena, "Review on lossless data compression using x-matchpro algorithm," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, 2017, pp. 1095–1100.
- [22] S. Hamdan, A. Awaian, and S. Almajali, "Compression techniques used in iot: A comparative study," in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*. IEEE, 2019, pp. 1–5.
- [23] E. Bristol, "Swinging door trending: Adaptive trend recording?" in *ISA National Conf. Proc., 1990*, 1990, pp. 749–754.
- [24] J. D. A. Correa, A. S. R. Pinto, C. Montez, and E. Leão, "Swinging door trending compression algorithm for iot environments," in *Anais Estendidos do IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais*. SBC, 2019, pp. 143–148.
- [25] A. Aziz, K. Singh, W. Osamy, and A. M. Khedr, "Effective algorithm for optimizing compressive sensing in iot and periodic monitoring applications," *Journal of Network and Computer Applications*, vol. 126, pp. 12–28, 2019.
- [26] T. L. Le and M.-H. Vo, "Lossless data compression algorithm to save energy in wireless sensor network," in *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*. IEEE, 2018, pp. 597–600.
- [27] B. R. Stojkoska and Z. Nikolovski, "Data compression for energy efficient iot solutions," in *2017 25th telecommunication forum (TELFOR)*. IEEE, 2017, pp. 1–4.
- [28] A. Chatterjee, R. J. Shah, and K. S. Hasan, "Efficient data compression for iot devices using huffman coding based techniques," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 5137–5141.
- [29] A. Hanumanthaiah, A. Gopinath, C. Arun, B. Hariharan, and R. Murugan, "Comparison of lossless data compression techniques in low-cost low-power (lclp) iot systems," in *2019 9th International Symposium on Embedded Computing and System Design (ISED)*. IEEE, 2019, pp. 1–5.
- [30] E. J. Neto, L. Augusto, D. C. Souza, and L. A. Guedes, "Adaptive swinging door trending: um algoritmo adaptativo para compressão de dados em tempo real," in *Proceedings of the Anais do XX Congresso Brasileiro de Automática, Belo Horizonte, Brazil, 2014*, pp. 20–24.
- [31] E. Leão, R. Moraes, C. Montez, P. Portugal, and F. Vasques, "CT-SIM: A simulation model for wide-scale cluster-tree networks based on the IEEE 802.15.4 and zigbee standards," *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, pp. 1–17, 2017.
- [32] Y. Tselishchev, A. Boulis, and L. Libman, "Experiences and lessons from implementing a wireless sensor network mac protocol in the castalia simulator," in *2010 IEEE Wireless Communication and Networking Conference*. IEEE, 2010, pp. 1–6.