



Universidade Federal do Piauí
Centro de Ciências da Natureza
Programa de Pós-Graduação em Ciência da Computação

**Um Mecanismo de Controle de
Congestionamento Disparado por Eventos
Através de Filtragem Local de Dados para RSSFs
Cluster-Tree baseadas no Padrão IEEE 802.15.4**

Natanael Ribeiro da Silva

Teresina-PI, Agosto de 2022

Natanael Ribeiro da Silva

**Um Mecanismo de Controle de Congestionamento
Disparado por Eventos Através de Filtragem Local de
Dados para RSSFs *Cluster-Tree* baseadas no Padrão IEEE
802.15.4**

Trabalho de Defesa de Mestrado apresentado ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação

Orientador: Erico Meneses Leão

Teresina-PI

Agosto de 2022

FICHA CATALOGRÁFICA
Universidade Federal do Piauí
Sistema de Bibliotecas da UFPI – SIBi/UFPI
Biblioteca Setorial do CCN

S586m Silva, Natanael Ribeiro da.
Um mecanismo de controle de congestionamento local disparado por eventos através de compactação de dados para RSSFs *Cluster-Tree* baseadas no padrão IEEE 802.15.4 / Natanael Ribeiro da Silva. – 2022.
62 f.: il.

Dissertação (Mestrado) – Universidade Federal do Piauí, Centro de Ciências da Natureza, Pós-Graduação em Ciência da Computação, Teresina, 2022.
“Orientador: Prof. Dr. Erico Meneses Leão”.

1. Redes de sensores sem fio. 2. *Cluster-tree*. 3. Filtragem de dados. 4. *Internet of things*. I. Leão, Erico Meneses.
II. Título.

CDD 004.6

Bibliotecária: Caryne Maria da Silva Gomes. CRB/3-1461

“Um Mecanismo de Controle de Congestionamento Local Disparado por Eventos Através de Compactação de Dados para RSSFs Cluster-Tree baseadas no Padrão IEEE 802.15.4”

NATANAEL RIBEIRO DA SILVA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Natureza da Universidade Federal do Piauí, como parte integrante dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Aprovada por:

Documento assinado digitalmente
 ERICO MENESES LEAO
Data: 31/08/2022 20:00:59-0300
Verifique em <https://verificador.iti.br>

Prof. Erico Meneses Leão
(Presidente da banca examinadora)

 Documento assinado digitalmente
Carlos Barros Montez
Data: 31/08/2022 17:21:44-0300
CPF: 752.035.027-49
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Carlos Montez
(Examinador externo à instituição)

Documento assinado digitalmente
 JOSE VALDEMIR DOS REIS JUNIOR
Data: 01/09/2022 10:10:33-0300
Verifique em <https://verificador.iti.br>

Prof. José Valdemir dos Reis Júnior
(Examinador interno)

Documento assinado digitalmente
 RICARDO DE ANDRADE LIRA RABELO
Data: 01/09/2022 06:04:24-0300
Verifique em <https://verificador.iti.br>

Prof. Ricardo de Andrade Lira Rabêlo
(Examinador interno)

Teresina, 31 de agosto de 2022

*Aos meus pais Antônio Pereira, Maria Cristina (in memorian) e Eliete Ribeiro, a minha
irmã Luana Cristina, amigos e principalmente a minha família (esposa e filhos),
por sempre estarem comigo em todos os momentos.*

Agradecimentos

Agradeço a Deus, o Senhor de tudo, que torna isso possível.

Agradeço aos meus pais, Antônio Pereira, Maria Cristina (in memoriam) e Eliete Ribeiro por todo o ensinamento e cuidado. A minha irmã Luana Cristina sempre pela torcida. A minha querida esposa Fernanda de Alencar, pela paciência, incentivo e amor. Aos meus lindos filhos Murilo, Estela e Cecília, que são os grandes responsáveis pela minha luta, dedicação e determinação.

Agradeço ao meu orientador, Erico, por todos os conselhos, pela paciência e ajuda nesse período acadêmico. Com certeza, será sempre uma grande referência.

Aos meus amigos de caminhada do mestrado que contribuíram direta e indiretamente para minha evolução e aprendizado.

Aos professores do curso de computação da UFPI, que sempre somaram conteúdo através de muita experiência.

*“Se não podes entender,
crê para que entendas.
A fé precede, o intelecto segue.”
(Santo Agostinho)*

Resumo

A Internet das Coisas (*IoT - Internet of Things*) é uma inovação tecnológica que tem se tornado bastante atrativa no âmbito comercial, industrial e residencial. Com isso, houve uma crescente aplicação principalmente nos negócios, com o desenvolvimento de mecanismos de comunicação eficientes e desenvolvimento de aplicações em Redes de Sensores Sem Fio (RSSF). Essa grande aplicabilidade torna-se necessária uma avaliação de benefícios, estratégias e dificuldades enfrentadas na aplicação dessa tecnologia. Dentro desse contexto, o conjunto de normas IEEE 802.15.4 e ZigBee forma uma pilha adequada de protocolos para implantação de RSSFs, oferecendo interoperabilidade, mobilidade e autonomia entre dispositivos que operam sob baixo consumo energético, baixo custo e baixas taxas de comunicação. Estas normas especificam uma topologia especial *peer-to-peer* chamada *cluster-tree*, na qual permite a sincronização de nós e é apontada como uma das topologias mais apropriadas para implantação em ambientes de larga escala. Entretanto, a ocorrência de congestionamentos evidencia-se como um dos maiores desafios de RSSFs *cluster-tree* de larga escala. Diante disso, torna-se necessário a proposta de mecanismos eficientes para mitigar os elevados atrasos de comunicação fim-a-fim e as possíveis perdas de pacotes, comuns num ambiente congestionado. Este trabalho de mestrado propõe um mecanismo de controle de congestionamento disparado por eventos através da filtragem local de dados para RSSFs *cluster-tree* de larga escala. A ideia subjacente é implementar um mecanismo de filtragem que possa reduzir a quantidade de dados transmitidos pelos nós sensores, evitando ou postergando possíveis congestionamentos e reconfigurações de rede. Como resultados, este trabalho fornece uma avaliação de desempenho por simulação que visa implementar e avaliar a eficiência dos principais mecanismos propostos. 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 assim congestionamentos e processos de reconfiguração de rede.

Palavras-chaves: Rede de Sensores sem Fio. IEEE 802.15.4. ZigBee. Cluster-tree. Controle de congestionamento. Filtragem de Dados. Internet das Coisas.

Abstract

Internet of Things (IoT) is a technological innovation that has become very attractive in the commercial, industrial, and home domains. As a consequence, there was a growing application, mainly in business, with the development of efficient communication approaches for Wireless Sensor Networks (WSN). This wide applicability requires the evaluation of benefits, strategies and difficulties faced in this technology. Within this context, the set of IEEE 802.15.4 and ZigBee standards composes an appropriate protocol stack to deploy WSNs, offering interoperability, mobility, and autonomy. These standards provide a special peer-to-peer topology called cluster-tree, in which allows node synchronization and it is pointed out as one of the most appropriate topologies for deployment in large-scale environments. However, the occurrence of congestion is one of the most challenges of large-scale cluster-tree networks. Therefore, it is necessary to define efficient mechanisms to mitigate the high end-to-end communication delays and packet losses, that are common in a congested environment. This master's work proposes the use of an efficient local filtering mechanism, in order to avoid or postpone congestion for large-scale cluster-tree WSNs. The underlying idea is to implement an filtering mechanism that can reduce the amount of data transmitted by sensor nodes, avoiding possible network congestions and reconfigurations. As expected results, this work intends to contribute to the state of the art of cluster-tree WSNs and provide a performance assessment by simulation of the proposed compression mechanism. The results show that the implementation of local filtering mechanisms at sensor nodes can provide quality of service for the monitoring flows, thus avoiding network congestion and reconfiguration processes.

Keywords: Wireless Sensor Network. IEEE 802.15.4. ZigBee. Cluster-tree. Congestion Control. Data Filtering. Internet of Things.

Lista de ilustrações

| | |
|--|----|
| Figura 1 – A pilha de protocolos IEEE 802.15.4/ZigBee. | 7 |
| Figura 2 – Topologias de Rede do padrão IEEE 802.15.4. | 8 |
| Figura 3 – A topologia <i>cluster-tree</i> com diferentes tipos de nós e relacionamentos. | 10 |
| Figura 4 – Modos de comunicação sem <i>beacons</i> e com <i>beacons</i> | 10 |
| Figura 5 – A estrutura <i>superframe</i> do padrão IEEE 802.15.4. | 11 |
| Figura 6 – Cenário de um ambiente de monitoramento: (a) ambiente normal (b) cenário de evento que poderá ocasionar congestionamento. | 13 |
| Figura 7 – Situação característica do congestionamento de um <i>cluster</i> | 14 |
| Figura 8 – Compressão sem perda (a) e com perda (b). | 18 |
| Figura 9 – Representação da Patente SDT. | 19 |
| Figura 10 – Compressão Delta. | 21 |
| Figura 11 – A arquitetura do mecanismo de filtragem proposto para RSSFs. | 33 |
| Figura 12 – Formação de Rede Orientada às características de geração de dados. | 35 |
| Figura 13 – Rede <i>cluster-tree</i> de monitoramento de variáveis físicas. | 39 |
| Figura 14 – Formato do cabeçalho do quadro de dados. | 39 |
| Figura 15 – Iniciando primeira etapa de compressão. | 40 |
| Figura 16 – Janela de Compressão. | 41 |
| Figura 17 – Representação ilustrativa do SDT. | 41 |
| Figura 18 – Processo de compressão Delta. | 43 |
| Figura 19 – Representação SDT + Delta. | 43 |
| Figura 20 – Funcionamento do mecanismo de compressão local proposto. | 44 |
| Figura 21 – Representação dos dados dos sensores e zonas de evento crítico (zona vermelha) e de funcionamento normal (zona verde). | 46 |
| Figura 22 – Atraso médio de comunicação fim-a-fim total dos nós pertencentes à zona de eventos críticos. | 47 |
| Figura 23 – Atraso médio de comunicação fim-a-fim dos nós pertencentes à zona de eventos críticos, durante a ocorrência do eventos. | 48 |
| Figura 24 – Atraso médio de comunicação fim-a-fim dos nós pertencentes à zona de eventos críticos, fora da ocorrência dos eventos. | 48 |
| Figura 25 – Taxa média de perda de pacotes total dos nós pertencentes à zona de eventos críticos. | 49 |
| Figura 26 – Taxa média de perda de pacotes dos nós pertencentes à zona de eventos críticos, durante a ocorrência do eventos. | 50 |
| Figura 27 – Taxa média de perda de pacotes dos nós pertencentes à zona de eventos críticos, fora da ocorrência dos eventos. | 50 |

| | |
|--|----|
| Figura 28 – Nós localizados fora da região de eventos críticos: (a) Atraso médio total e (b) Taxa média de perda de pacotes. | 51 |
| Figura 29 – Nós localizados na região de eventos críticos, durante a ocorrência do evento: (a) Atraso médio de comunicação e (b) Taxa média de perda de pacotes. | 51 |
| Figura 30 – Nós localizados fora da região de eventos críticos: (a) Atraso médio total e (b) Taxa média de perda de pacotes. | 52 |

Lista de tabelas

| | |
|---|----|
| Tabela 1 – Algoritmos de compressão citados no trabalho. | 30 |
| Tabela 2 – Configuração dos principais parâmetros de simulação. | 47 |

Lista de abreviaturas e siglas

| | |
|---------|--|
| AODV | <i>Ad-hoc On Demand Distance Vector</i> |
| ADCS | <i>Adaptive Data Compression Scheme</i> |
| AQM | <i>Active Queue Management</i> |
| BI | <i>Beacon Interval</i> |
| BO | <i>Beacon Order</i> |
| CA | <i>Critical Aperture</i> |
| CAP | <i>Contention Access Period</i> |
| CBDR | <i>Compression-Based Data Reduction</i> |
| CFP | <i>Contention Free Period</i> |
| CH | <i>Cluster-Head</i> |
| CW | <i>Contention Window</i> |
| CSLB-CS | <i>Cluster Size Load Balancing for CS</i> |
| CSMA/CA | <i>Carrier Sense Multiple Access Collision Avoidance</i> |
| DSR | <i>Dynamic Source Routing</i> |
| ECN | <i>Explicit Congestion Notification</i> |
| EUA | <i>Estados Unidos da América</i> |
| FFD | <i>Full Function Device</i> |
| FR | <i>Fractal Resampling</i> |
| GHz | <i>Giga-Hertz</i> |
| GTS | <i>Guaranteed Time Slots</i> |
| IEEE | <i>Institute of Electrical and Electronics Engineers</i> |
| IoT | <i>Internet of Things</i> |
| JTD | <i>Janela de oportunidade Top-Down</i> |

| | |
|---------|---|
| kbps | <i>Kilobits por segundo</i> |
| LDPC | <i>Low-Density Parity-Check</i> |
| LR | <i>Low Reliability</i> |
| LR-WPAN | <i>Low-Rate Wireless Personal Area Network</i> |
| LZW | <i>Lempel-Ziv-Welch</i> |
| MAC | <i>Medium Access Control</i> |
| MDL | <i>Minimum Description Length</i> |
| MLME | <i>MAC -Layer Management Entity</i> |
| MILP | <i>MAC -Mixed Integer Linear Programming</i> |
| PAN | <i>Personal Area Network</i> |
| PDR | <i>Packet Delivery Ratio</i> |
| PHY | <i>Physical Layer</i> |
| PLME | <i>PHY -Layer Management Entity</i> |
| RED | <i>Random Early Detection</i> |
| REST | <i>Representational State Transfer</i> |
| RLE | <i>Run Length Encoding</i> |
| RFD | <i>Reduced Function Device</i> |
| RDN | <i>Residual Dense Network</i> |
| RFID | <i>Radio Frequency Identification</i> |
| RNA | <i>Rede Neural Artificial</i> |
| RPL | <i>IPv6 Routing Protocol for Low Power and Lossy Networks</i> |
| RSSF | <i>Redes de Sensores Sem Fio</i> |
| s | <i>segundos</i> |
| SD | <i>Superframe Duration</i> |
| SO | <i>Superframe Order</i> |
| SZ | <i>Squeeze</i> |

| | |
|------|--|
| TAC | <i>Tiny Anomaly Compressor</i> |
| TEDA | <i>Análise de Dados de Tipicidade e Excentricidade</i> |
| TTDR | <i>Two-Tier Data Reduction</i> |
| WT | <i>Wavelet Transform</i> |
| ZC | <i>ZigBee Coordinator</i> |
| ZR | <i>ZigBee Router</i> |
| ZED | <i>ZigBee End-Nodes</i> |

Sumário

| | | |
|----------|---|-----------|
| 1 | INTRODUÇÃO | 1 |
| 1.1 | Motivação e Objetivos | 4 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 7 |
| 2.1 | O Conjunto de Normas IEEE 802.15.4 e ZigBee | 7 |
| 2.1.1 | Redes Cluster-tree | 9 |
| 2.2 | Congestionamento em RSSFs | 13 |
| 2.2.1 | Características do Congestionamento | 13 |
| 2.2.2 | Métodos para Detecção de Eventos em RSSFs | 15 |
| 2.3 | Compressão de Dados | 17 |
| 2.3.1 | Swinging Door Trending (SDT) | 18 |
| 2.3.2 | Codificação Delta | 20 |
| 3 | TRABALHOS RELACIONADOS | 23 |
| 3.1 | Algoritmos Tradicionais de Compressão | 23 |
| 3.2 | Algoritmos Baseados em Sistemas Inteligentes | 25 |
| 3.3 | Algoritmos Híbridos | 27 |
| 3.4 | Considerações do Capítulo | 31 |
| 4 | MECANISMO DE CONTROLE DE CONGESTIONAMENTOS ATRAVÉS DE FILTRAGEM LOCAL DE DADOS PARA RSSFS CLUSTER-TREE | 33 |
| 4.1 | Módulo 01: Processo de Formação da Rede Orientada. | 34 |
| 4.2 | Módulo 02: Configuração eficiente das estruturas de comunicação. | 35 |
| 4.3 | Módulo 03: Escalonamento para priorização do tráfego de monitoramento (<i>upstream</i>). | 36 |
| 4.4 | Módulo 04: Funcionamento operacional com a finalidade de detectar possíveis eventos de modificação de carga. | 36 |
| 4.5 | Módulo 05: Mecanismo de reconfiguração dinâmica da rede orientada a eventos. | 37 |
| 4.6 | Módulo 06: Novo módulo de filtragem local de dados proposto. | 38 |
| 4.6.1 | Detecção do Evento de Carga | 38 |
| 4.6.2 | Processo de Filtragem dos Dados | 40 |
| 4.6.2.1 | 1ª Etapa: Processo de Filtragem SDT | 40 |
| 4.6.2.2 | 2ª Etapa: Processo de Compressão Delta | 41 |

| | | |
|------------|--|-----------|
| 5 | RESULTADOS E DISCUSSÃO | 45 |
| 5.1 | Cenários de Simulação | 45 |
| 5.2 | Métricas de Avaliação | 46 |
| 5.3 | Resultados e Discussão do Mecanismo de Filtragem Baseado Apenas na Compressão SDT | 47 |
| 5.4 | Resultados e Discussões do Mecanismo de Compressão Utilizando SDT e Delta | 51 |
| 5.5 | Considerações Finais | 53 |
| 6 | CONCLUSÕES E TRABALHOS FUTUROS | 55 |
| 6.1 | Conclusão | 55 |
| 6.2 | Trabalhos Futuros | 55 |
| | REFERÊNCIAS | 57 |

1 Introdução

As Redes de Sensores Sem Fio (RSSFs) são redes *ad hoc* (que dispensam o uso de um ponto de acesso comum aos dispositivos conectados a ela) 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 (YANG, 2014). 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 (RODRIGUES et al., 2010). Esses dispositivos são caracterizados por serem de baixo custo e se comunicarem com baixas taxas de transmissão e baixo consumo energético (STANKOVIC; WOOD; HE, 2011). Essas particularidades possibilitam o desenvolvimento de diferentes domínios de aplicações de larga escala, tais como: monitoramento da saúde, transporte, casas inteligentes, cidades inteligentes, automação industrial, agricultura de precisão, etc. (MOHAMMADI et al., 2018).

Com o advento da Internet das Coisas (*Internet of Things - IoT*), que é um dos pilares da Quarta Revolução Industrial (*Industry 4.0*), cujo objetivo é conectar sistemas e equipamentos para facilitar a alta personalização e automação de produtos, tornando a produção em massa mais flexível, eficaz e eficiente, as RSSFs têm se tornado uma área extremamente investigada pela sua grande adaptabilidade em aplicações de larga escala (ADEBAYO; CHAUBEY; NUMBU, 2019; KHALIL et al., 2021). Além de ser uma tecnologia adequada para ambientes inóspitos e de difícil acesso, estas redes são compostas por propriedades como dinamicidade, autonomia e flexibilidade no processo de conversação entre os dispositivos (AKYILDIZ et al., 2002). Em um ambiente industrial, por exemplo, onde o ganho, ambiente e a organização são particularidades elementares, a utilização de uma rede cabeada dimensionada para uma grande área interna ou externa torna-se bastante complexa e custosa se comparada com tecnologias de redes sem fio (SHEN; WANG; SUN, 2004). Além disso, estas redes possuem como principal vantagem a capacidade energética dos seus nós sensores, comumente alimentados por baterias, as quais podem durar muito tempo se forem configuradas corretamente. Ademais, elas são totalmente independentes e capazes de agir por conta própria se o contexto exigir, sem que haja qualquer infraestrutura pronta. (ROMAN; ALCARAZ; LOPEZ, 2007).

Com os crescentes avanços tecnológicos na área de RSSF, ela está se tornando parte integrante de nossas vidas (MATIN, 2012). As RSSFs, juntamente com o RFID (*Radio Frequency Identification*), tornaram-se tecnologias chave para o surgimento da IoT (SOBRAL et al., 2018). A principal ideia dessa revolução tecnológica é fazer com que seja possível a conexão de pessoas a coisas de modo a não depender do lugar, e a geração

e disseminação da informação seja feita de forma inteligente, autônoma e de baixo custo (VERMESAN; FRIESS et al., 2014).

A visão da IoT é transformar objetos tradicionais em inteligentes, explorando uma ampla gama de tecnologias avançadas, de dispositivos incorporados e tecnologias de comunicação a protocolos de Internet e análise de dados (MOHAMMADI et al., 2018). Para suportar diferentes tipos de aplicações, a IoT utiliza como base as RSSFs, podendo dessa forma conectar uma variedade enorme de objetos (GLUHAK et al., 2011). Entre as principais áreas de aplicação, temos: saúde, indústria, robótica, educação, transporte, logística, veículos autônomos, agricultura, residências, manutenção, marketing, gestão etc (RGHIOUI; OUMNAD, 2017; ROY et al., 2020). Desde o nascimento da IoT entre 1999 a 2003, o número de dispositivos cresceu para 500 milhões de dispositivos conectados. Em 2023, a previsão é de mais de 50 bilhões de dispositivos conectados, atendendo a uma população de 7.6 bilhões de pessoas (RAMALINGAM; TWETEN, 2019).

Existem na atualidade diferentes plataformas 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 (RODRIGUES et al., 2010). Todas elas são focadas em operações de curto alcance, com baixa taxa de dados, eficiência energética e custos reduzidos de implementação (LINO et al., 2019). Diante deste contexto, faz-se necessário encontrar um conjunto de normas amplamente utilizadas e recomendadas. A robustez necessária, a demanda de baixa taxa de transmissão e o baixo custo de implantação, apontam para as soluções de RSSF utilizando-se a tecnologia ZigBee (ZIGBEE, 2012b). O padrão ZigBee (ZIGBEE, 2012b), foi construído sobre a pilha de protocolos IEEE 802.15.4 IEEE (2016), obedecendo aos requisitos que estejam de acordo com as necessidades únicas das comunicações sem fio, confiáveis, de baixo nível de transmissão entre dispositivos eletrônicos para monitoramento e controle de processos, como por exemplo, uma RSSF. Este padrão possui baixa latência, taxa de transferência de até 250 Kbps, otimização do consumo de energia (a vida útil, para uma pilha AA pode durar de 1 a 2 anos; a vida útil de uma bateria de lítio, pode durar de 10 a 20 anos), baixa corrente na transmissão (usualmente, menos de 30mA), possibilidade de implementação de redes com elevados números de dispositivos (até 2^{16}), pouca complexidade dos nós da rede a um custo relativamente baixo (REIS; FERNANDEZ, 2007). Portanto, ZigBee tem sido considerado como um padrão *de facto* que está de acordo com as necessidades de comunicação sem fio entre dispositivos de controle e sensores (QUEIROZ et al., 2016). É importante salientar que o IEEE 802.15.4 ficou responsável pela definição das duas camadas mais baixas do modelo OSI, a camada física (*PHYSICAL* - PHY) e a subcamada de Controle de Acesso ao Meio (*Medium Access Control* - MAC), para aplicações *Low-Rate Wireless Personal Area Network* (LR-WPAN), enquanto que ZigBee Alliance define as camadas superiores (Redes e Aplicação) (SANTOS, 2013).

O conjunto de normas IEEE 802.15.4/ZigBee suporta várias topologias de rede. Basicamente, pode-se citar: topologias estrela e *peer-to-peer*. A escolha da topologia adequada dependerá dos requisitos da aplicação. A topologia estrela é a organização de rede mais simples, em que todo gerenciamento e comunicação está centrado a um único nó, conhecido como coordenador PAN (*Personal Area Network*), que é responsável por iniciar, encerrar e encaminhar a comunicação pela rede. Ainda que redes estrela sejam fáceis de configurar e gerenciar, elas são limitadas por conta da cobertura da rede pelos nós filhos associados, já que seu raio de detecção irá depender da localização do coordenador. Isso é um fator limitante para a construção de RSSFs em larga escala (LINO et al., 2019).

Em topologias *peer-to-peer*, o coordenador PAN também é o nó central e responsável por iniciar o processo de formação da rede. No entanto, diferentemente de redes estrela, todos os dispositivos podem se comunicar com os demais, desde que estejam dentro do raio de transmissão. Redes *peer-to-peer* são consideradas redes *ad hoc* pela sua capacidade de se auto-organizar (*self-organizing*) e de se auto-estruturar (*self-healing*). O coordenador PAN é responsável por enviar quadros *beacons*, anunciando a existência da rede. Já os dispositivos recém adicionados, irão se estabelecer enviando *beacon frames* para novos candidatos com o objetivo de expandir a rede (SANTOS, 2013). Diante disso, é possível formar redes mais robustas e complexas, além de termos redundância de roteamento e escalabilidade, como, por exemplo, as redes *cluster-tree*, também conhecidas como Árvore de Agrupamentos. *Cluster-tree* é um tipo especial de rede *peer-to-peer* e é apontada na literatura como uma das topologias mais adequadas para implementar RSSFs em larga escala (LEÃO et al., 2017). Neste tipo de topologia, podemos encontrar agrupamentos, compostos por nós, sendo estes gerenciados por um nó chamado de *Cluster-Head (CH)*. Essa é a ideia mais simples a ser implementada, considerando a existência de apenas um *cluster*. Entretanto, numa rede mais robusta, podemos encontrar vários aglomerados interligados pelos seus nós coordenadores (CHs), formando assim, uma estrutura de rede hierárquica *multicluster*, a fim de proporcionar maior cobertura de rede, habilitando, por conseguinte, implementações em larga escala. (LEÃO, 2016).

Por conta dessa complexidade e robustez, em que podemos encontrar diversos *clusters* sobrepostos gerando um fluxo abundante de dados, é comum existir um padrão de tráfego (controle ou monitoramento), na maioria das vezes, intenso. Diante deste contexto, devido às particularidades dos típicos padrões de tráfego empregados nessas redes, é comum a ocorrência de congestionamentos. Isso irá proporcionar uma série de problemas como aumento do consumo energético dos nós sensores, maiores perdas de pacotes, atrasos na comunicação fim-a-fim e impedimento da detecção de incidentes e transmissão confiável de dados. Estes aspectos tornam corriqueiro a questão do congestionamento em RSSFs, sobretudo em ambientes de larga escala. Encontrar meios para realizar o controle de congestionamentos, de forma que seja possível evitá-los, torna-se um desafio importante para uma transmissão eficiente de dados por pacotes (BARATH et al., 2019).

Em vista disso, este trabalho visa mitigar os elevados atrasos de comunicação fim-a-fim e as possíveis perdas de pacote comuns em ambientes congestionados, contribuindo assim com o estado da arte fundamentado em soluções de compressão de dados.

1.1 Motivação e Objetivos

A literatura tem apontado que o uso de topologias *cluster-tree* é o mais indicado em ambientes de RSSFs de larga escala, já que proporcionam características tais como: eficiência energética, sincronização de tempo, boa escalabilidade, possibilidade de integração ou junção de dados (LEÃO, 2016); este modelo de topologia ainda apresenta deficiências geradas pelo modelo hierárquico de comunicação, particularmente sob o tráfego de disseminação de dados. Estas questões acentuam problemas importantes de rede tais como retardos de comunicação fim-a-fim, perda de pacotes e congestionamentos, o que fomenta o desenvolvimento de soluções ou alternativas eficientes para superá-los (LINO et al., 2019).

Dentro desse contexto, o principal objetivo deste trabalho é propor um mecanismo de controle de congestionamento disparado por eventos através da filtragem local de dados para RSSFs *Cluster-Tree* baseados no IEEE 802.15.4/ZigBee. Dessa forma, os objetivos específicos desse trabalho têm sido endereçados como segue:

1. A concepção e elaboração de uma abordagem arquitetural classificada em diferentes fases, que envolvem desde a utilização de uma arquitetura robusta já existente de redes *cluster-tree* (LINO et al., 2019), até o tratamento de eventos de congestionamentos na rede, a fim de proporcionar uma boa qualidade de serviço para típico tráfego de monitoramento gerado pelos dispositivos que compõe a rede.
2. Melhorar uma arquitetura existente, implementando um mecanismo eficiente de filtragem local de dados baseado na detecção de eventos, objetivando evitar congestionamentos em RSSFs *cluster-tree* de larga escala, baseada nas normas IEEE 802.15.4 e ZigBee.
3. Realizar um estudo de simulação, a fim de avaliar os mecanismos propostos neste trabalho sob diferentes perspectivas de rede e apreciar os resultados tendo em vista distintas métricas de avaliação, tais como: atrasos de comunicação fim-a-fim, taxa de perda de mensagens cálculos de reconfiguração de rede.

Questões de Pesquisa

Este trabalho de Dissertação de Mestrado tem o foco na seguinte questão de pesquisa:

Como podemos evitar ou postergar a reconfiguração de uma RSSF *Cluster-Tree* baseada na utilização de um mecanismo de filtragem de dados diante de evento de congestionamento?

Para essa questão, a hipótese definida é que, considerando: i) a utilização de uma arquitetura de reconfiguração baseada nas características de geração de dados dos dispositivos que compõem a rede, ii) um monitoramento contínuo para detecção de eventos de congestionamentos, iii) ativação de mecanismo de filtragem/compressão de dados até a normalização do fluxo e capacidade de processamento da rede, e iv) por fim, caso necessário, disparar um processo de reconfiguração global de rede; é possível implementar uma solução eficiente que possa atuar de forma automatizada para postergar ou evitar congestionamentos de rede, diminuindo assim atrasos de comunicação fim-a-fim para o típico tráfego de monitoramento gerado pelos nós sensores que compõem a rede e mantendo uma boa qualidade de serviço para os principais fluxos de tráfego de dados.

Organização do Documento

O restante deste trabalho está organizado como segue. Capítulo 2 apresenta os conceitos bases para a compreensão desta pesquisa, que inclui os protocolos e os principais componentes de redes *cluster-tree*. Em seguida, o Capítulo 3 sintetiza os principais trabalhos relacionados ao contexto deste trabalho, das ideias mais pertinentes voltados para diversos tipos de soluções de filtragem/compressão de dados.

O Capítulo 4 apresenta o mecanismo de controle de congestionamento disparado por eventos através de filtragem local de dados para RSSFs *cluster-tree* baseadas no conjunto de normas IEEE 802.15.4/ZigBee. Dessa forma, é apresentado o funcionamento do mecanismo proposto, partindo de uma arquitetura de redes *cluster-tree* já existente, até as etapas de detecção e filtragem dos dados de eventos críticos.

O Capítulo 5 apresenta uma avaliação de desempenho através de simulação, a fim de demonstrar a viabilidade do mecanismo de filtragem de dados proposto neste trabalho de mestrado. O mecanismo é avaliado sob diversos cenários, utilizando diferentes métricas de avaliação

Por fim, o Capítulo 6 apresenta as principais conclusões do trabalho, bem como propostas para trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo, serão apresentadas as principais particularidades do conjunto de especificações IEEE 802.15.4 e ZigBee. Posteriormente, delineou-se os aspectos específicos das redes *cluster-tree*, mostrando suas principais vantagens e desvantagens. Em seguida, enfatizamos o problema de congestionamento e formas para controle baseado em eventos.

2.1 O Conjunto de Normas IEEE 802.15.4 e ZigBee

O conjunto de normas IEEE 802.15.4 (IEEE, 2015) e ZigBee (ZIGBEE, 2012a) é apontado na literatura como as tecnologias sem fio mais comumente utilizadas para projetar RSSFs. É importante frisar que o IEEE 802.15.4 ficou responsável pela definição das duas camadas mais baixas da tecnologia ZigBee: camada Física (*PHY*) e a subcamada de Controle de Acesso ao Meio (*MAC*) para aplicações LR-WPANs, enquanto que a norma ZigBee Alliance especifica as camadas superiores (Redes e Aplicação) (SANTOS, 2013). A Figura 1 ilustra a pilha de protocolos IEEE 802.15.4/ZigBee.

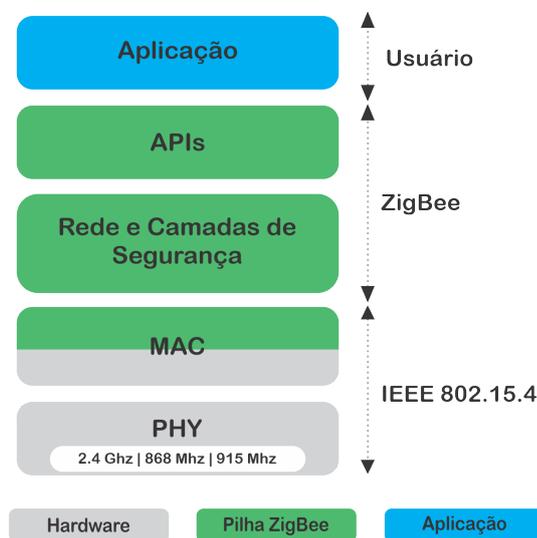


Figura 1 – A pilha de protocolos IEEE 802.15.4/ZigBee.

De acordo com a Figura 1, pode-se observar uma divisão das camadas de acordo com a tecnologia ZigBee ou IEEE 802.15.4. As camadas superiores (Aplicação e APIs) possuem tarefas relacionadas à segurança executadas pelo sensores que são: estabelecimentos de conexões, adição, atualização e remoção de dispositivos (KUMAR; MANE, 2016). A camada de rede (Rede e Camadas de Segurança) suporta alguns protocolos de roteamento, tais como: AODV (*Ad-hoc On Demand Distant Vector*), DSR (*Dynamic Source Routing*)

e RPL (*IPv6 Routing Protocol for Low Power and Lossy Networks*). Nas camadas inferiores temos a MAC e PHY as quais possuem os módulos: MLME (*MAC Layer Management Entity*) e PLME (*PHY Layer Management Entity*). A finalidade da camada MAC é gerenciar o acesso ao meio e as estrutura de comunicação, enquanto que a camada PHY abrange técnicas de modulação, variáveis de energia e bandas de frequência baseadas nas regiões EUA (*902 a 928 MHz*, 10 canais), Europa (*868.0 a 868.6 MHz*, 1 canal) e o resto do mundo (*2,4 a 2,48 GHz*, 16 canais) (HARA et al., 2005).

A utilização dos dispositivos (nós sensores) na rede vai depender das funções que irão desempenhar. O IEEE 802.15.4 estabelece dois tipos de dispositivos: FFD (*Full Function Devices*) e RFD (*Reduced Function Devices*). Já o ZigBee define as seguintes terminologias: ZC (*ZigBee Coordinator*), ZR (*ZigBee Router*) e ZED (*ZigBee End Devices*).

No IEEE 802.15.4, aqueles nós identificados na rede como finais são chamados de RFDs. Já no ZigBee, nós com as mesmas peculiaridades são definidos como ZED. Estes dispositivos não possuem capacidade de coordenar nós vizinhos e são responsáveis unicamente por realizar atividades básicas como, por exemplo, escaneamento de canal, requerimento de associação de rede e formação de tráfego de monitoramento. Já os dispositivos FFDs são considerados mais robustos, usualmente com uma capacidade superior de monitoramento, processamento e capacidade energética. Por fim, são responsáveis por atividades mais complexas, tais como: coordenação da rede, roteamento, envio de pacotes e mecanismos de agregação, filtragem, compactação ou junção de dados.

Como já supracitado, o conjunto de normas IEEE 802.15.4/ZigBee pode trabalhar com duas topologias: estrela e *peer-to-peer*. A escolha da topologia adequada dependerá dos requisitos da aplicação. A Figura 2 ilustra as topologias de rede definidas no padrão IEEE 802.15.4.

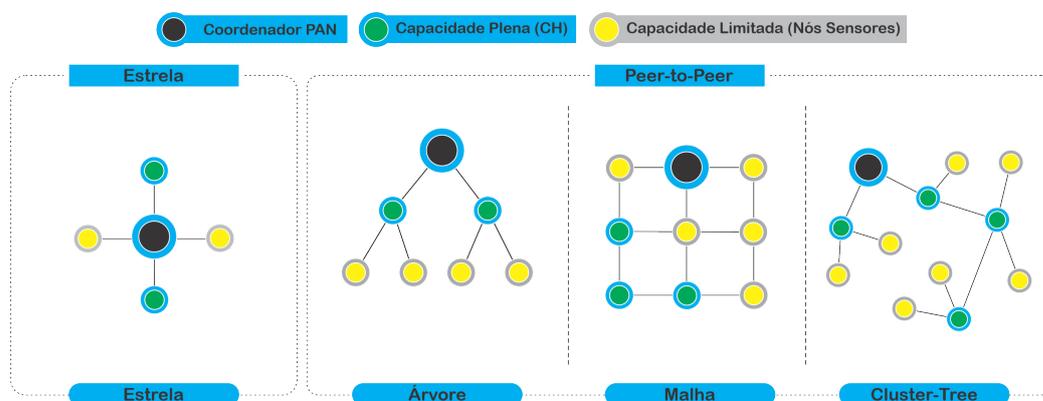


Figura 2 – Topologias de Rede do padrão IEEE 802.15.4.

A topologia estrela é a organização de rede mais simples, onde todos os nós sensores estão diretamente conectados ao coordenador PAN. O coordenador PAN é um nó único, que é responsável por todas as atividades de gestão e comunicação (paradigma de comunicação

centralizada). Apesar de ser fácil de construir e gerenciar, a fragilidade desta topologia é que sua cobertura é limitada pela faixa de detecção de seus nós, o que impede implantações em larga escala (LEÃO et al., 2017).

Na topologia *peer-to-peer* todos os dispositivos podem se comunicar uns com os outros, desde que estejam na mesma área de cobertura. Essa formação de topologia permite variações mais complexas, tais como redes em árvore de agrupamentos (*cluster-tree*) ou em malha (*mesh*) (ANDRADE et al., 2019). Dentre as vantagens da topologia *mesh*, podemos destacar a baixa complexidade, flexibilidade, redundância no roteamento e boa escalabilidade. Entretanto, esta topologia é omissa no mecanismo de sincronização de tempo, o que possibilitaria configurar distintos ciclos de trabalho nos nós a fim de aumentar o tempo de vida da rede. Este fato, em especial, merece atenção, uma vez que a eficiência energética é um dos aspectos primordiais para uma RSSF, que impõem requisitos rigorosos de consumo de energia (PANWAR; KUMAR, 2012).

Em contrapartida, na topologia *cluster-tree*, os nós são agrupados em *clusters* coordenados por um *cluster-head* (CH). O *cluster-head* é o responsável pela sincronização entre nós que fazem parte do seu agrupamento e também pela comunicação entre os demais (ANDRADE et al., 2019). Nesse modelo, todos os nós são organizados em uma relação de pai-filho e a sincronização é alcançada através de um roteamento determinístico, sendo portanto excelente para aplicações de monitoramento de larga escala (SEVERINO; PEREIRA; TOVAR, 2014).

2.1.1 Redes Cluster-tree

Podemos dizer que a topologia *cluster-tree* é um caso especial da topologia *peer-to-peer*. Nessa topologia, os nós sensores são agrupados e organizados em (*clusters*), coordenados por um nó central chamado coordenador (*Cluster-Head – CH*). Os CHs são responsáveis por fornecer sincronização para seus nós filhos associados e centralizar todas as atividades de comunicação contidas em seus *clusters*.

Todo 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. Após formar seu próprio *cluster*, o coordenador PAN pode selecionar uma quantidade específica de nós filhos, através dos pedidos de associação de novos nós, para que possam formar seus próprios *clusters*. 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. A Figura 3 ilustra a topologia *cluster-tree* e seus principais componentes no padrão IEEE 802.15.4.

O processo de formação de redes *cluster-tree* consiste num conjunto de estratégias,

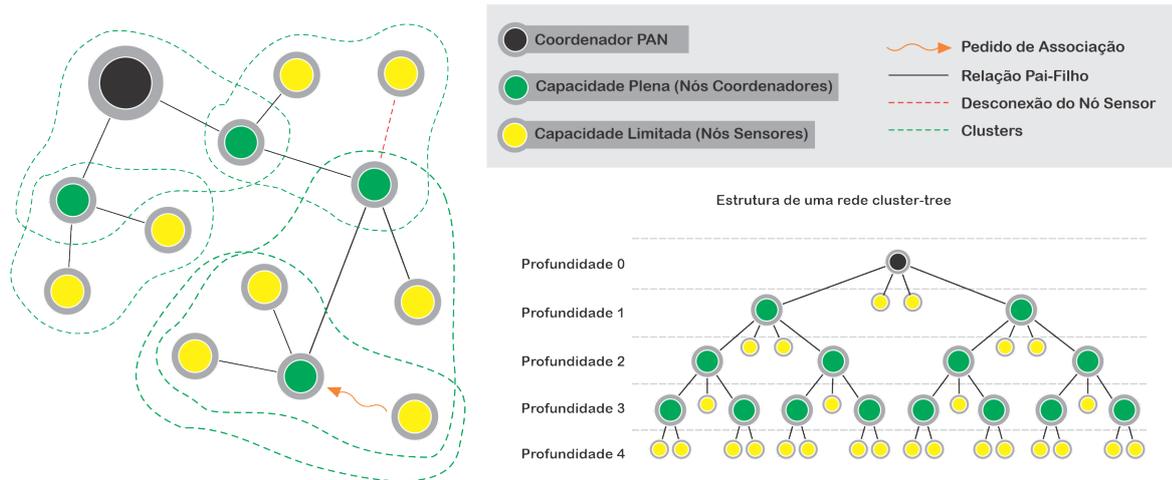


Figura 3 – A topologia *cluster-tree* com diferentes tipos de nós e relacionamentos.

que envolve a escolha de um canal de comunicação, a habilidade de auto organização em *clusters*, procedimentos de associação de nós, política de comunicação inter e entre *clusters*, seleção de nós coordenadores e algoritmos de sincronização e escalonamento. Essas particularidades dependem do quantitativo de nós da rede e de como estes foram ordenados na fase de implantação.

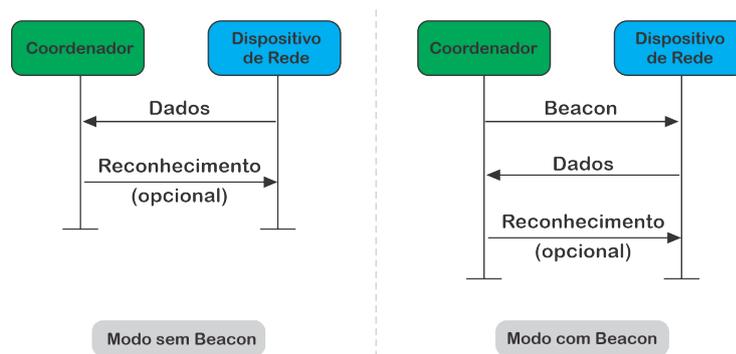


Figura 4 – Modos de comunicação sem *beacons* e com *beacons*

Em topologias *cluster-tree* baseadas no padrão IEEE 802.15.4, temos uma árvore de múltiplos *clusters* que podem implementar ciclos de trabalho reduzidos, provendo uma economia de energia dos seus nós associados. Sucintamente, uma rede *cluster-tree* é formada por CHs, que são responsáveis por coordenar cada *cluster* e realizar a sincronização entre grupos de nós filhos, através de quadros de sinalização (*beacons*) conforme é mostrado na Figura 4. Estas redes 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.

Basicamente, a estrutura *superframe* é delimitada por dois parâmetros: *macBeaconOrder* (BO) e *macSuperframeOrder* (SO). Esses parâmetros determinam o intervalo de *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 regularmente seus quadros de *beacons*. Por sua vez, o SD estabelece o período de comunicação dos *clusters*. O BI e SD são definidos como segue:

$$\begin{aligned} BI &= aBaseSuperframeDuration \times 2^{BO}, & 0 \leq BO \leq 14 \\ SD &= aBaseSuperframeDuration \times 2^{SO}, & 0 \leq SO \leq BO \end{aligned} \quad (2.1)$$

onde o valor 14 é o limite determinado pela 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) (CUNHA et al., 2007). A Figura 5 mostra a estrutura *superframe*.

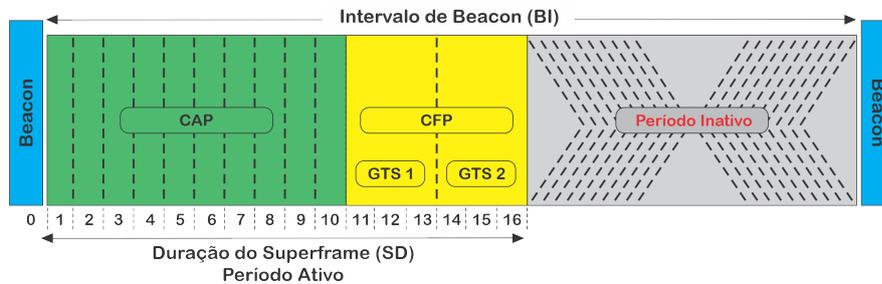


Figura 5 – A estrutura *superframe* do padrão IEEE 802.15.4.

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 semelhante ao SO , não existe porção inativa, assim dizendo, 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 sensores, trocam mensagens. Esta porção é subdividida em dois períodos: Período Livre de Contenção (*Contention Free Period* - CFP) e o Período de Acesso com Contenção (*Contention Access Period* - CAP).

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 – GTS (*Guaranteed Time Slot*) – com a intenção de que um nó tenha acesso ao canal de forma dedicada e possa transmitir mensagens livre de contenção.

Em redes *cluster-tree* existem, particularmente, dois tipos de tráfego de dados: tráfego de controle (*downstream traffic*) e o tráfego de monitoramento (*upstream traffic*).

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 a típicas mensagens geradas periodicamente pelos nós sensores, nas quais são encaminhadas a nós especiais, chamados de nós *sink* (*base station - estação base*). Um nó *sink* encarrega-se a função de coletar dados dos nós sensores e é responsável pela ligação da RSSF e as redes externas (como 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 ser dotado de maior capacidade de recursos.

Já acerca do tráfego de controle (*downstream*), diferentemente do *upstream*, os comandos são gerados pelos nós coordenadores, com o propósito de permitir atividades de controle e/ou reconfigurações da 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 outros nós coordenadores ou nós finais da rede (comportamento *top-down*).

Em redes *cluster-tree* é possível que haja colisões entre os quadros *beacons* de *clusters* vizinhos e/ou quadro de dados se não houver a implementação de mecanismos de sincronização (KOUBAA; CUNHA; ALVES, 2007). Portanto, pelo fato do canal de comunicação ser compartilhado, uma questão fundamental é como gerenciar e coordenar o acesso ao meio, evitando ou detectando essas colisões (FELSKE et al., 2013). À vista disso, é necessário a utilização de técnicas de sincronização, como por exemplo, o método de ordenação de períodos ativos chamado de escalonamento de *beacons*. Em resumo, a ideia é organizar os períodos ativos dos *clusters* ao longo do tempo, com a finalidade de evitar interferências entre eles. Isto posto, durante o período ativo de cada *cluster*, nós sensores associados podem interagir com seus coordenadores e vice-versa, de forma que sejam evitados interferências com os nós sensores de *clusters* vizinhos.

Os níveis da árvore estão diretamente relacionados com os esquemas de escalonamento de *beacons* e influenciam de modo direto no comportamento dos fluxos de mensagens de RSSFs. Por conseguinte, a seleção de um determinado mecanismo de sincronização pode priorizar (de forma opcional) um tipo de tráfego de dados, assim sendo um problema de otimização para redes *cluster-tree*. Exemplificando, o esquema de escalonamento *top-down* tende a priorizar o tráfego de controle, uma vez que os SDs de cada *cluster* são ordenados a partir do coordenador PAN em direção aos *clusters* mais profundos. Por outro lado, o escalonamento *bottom-up* tende a favorecer o tráfego de monitoramento, já que as durações de *superframes* são ordenadas a partir de *clusters* mais profundos, conforme visto na Figura 3, até alcançar o coordenador PAN.

2.2 Congestionamento em RSSFs

Congestionamentos em RSSFs ocorrem quando temos uma intensidade de tráfego maior que a vazão suportada pela rede. Diante de um aumento de fluxo, é possível que aconteça congestionamentos e efeitos adversos na rede. As Figuras 6(a) e 6(b) mostram um ambiente de monitoramento normal e a ocorrência de um fenômeno da natureza capturado por nós sensores, respectivamente. Essa captura poderá ampliar consideravelmente a quantidade de dados enviados pela rede, sobrecarregando por exemplo a capacidade de processamento dos *buffers* dos nós coordenadores.

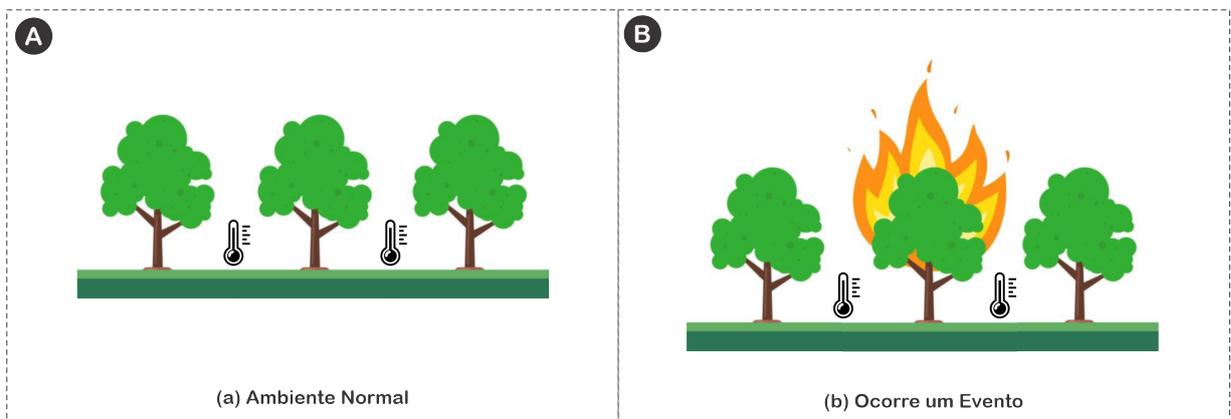


Figura 6 – Cenário de um ambiente de monitoramento: (a) ambiente normal (b) cenário de evento que poderá ocasionar congestionamento.

2.2.1 Características do 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 (KIRAZA; ÇAKIROĞLUB, 2014), 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 (RATHNAYAKA; POTDAR, 2013). Tal acontecimento se deve, sobretudo, devido ao alto fluxo de dados gerados pelos nós sensores, que transitam pela rede em direção ao nó *sink* (YIN et al., 2009), embora este problema também ocorra no fluxo de controle (*downstream*), durante a operação ou reconfiguração da 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 (GHAFFARI, 2015). Assim, para lidar com tais desafios e problemas em RSSFs, é necessário considerar e controlar o fator de congestionamento.

Devido ao escalonamento de *beacons*, os nós coordenadores (CHs) só podem enviar pacotes em determinados momentos do tempo. Por conta disso, o tamanho dos *buffers* podem se tornar pequenos, reduzindo a capacidade de retenção, caracterizando uma situação de congestionamento, tal como ilustrado na Figura 7. Dessa forma, colisão de pacotes, tempo insuficiente de comunicação, sobrecarga de *buffers*, contenção do canal de comunicação, taxa e esquemas de transmissão de dados de *muitos para um* são alguns dos efeitos que favorecem a perda de pacotes e a dificuldade de acesso ao meio, o que pode impossibilitar as atividades de comunicação, gerando, desse modo, atrasos e comprometimento quanto à confiabilidade e desempenho da rede, além de aumentar o consumo de energia dos nós sensores.

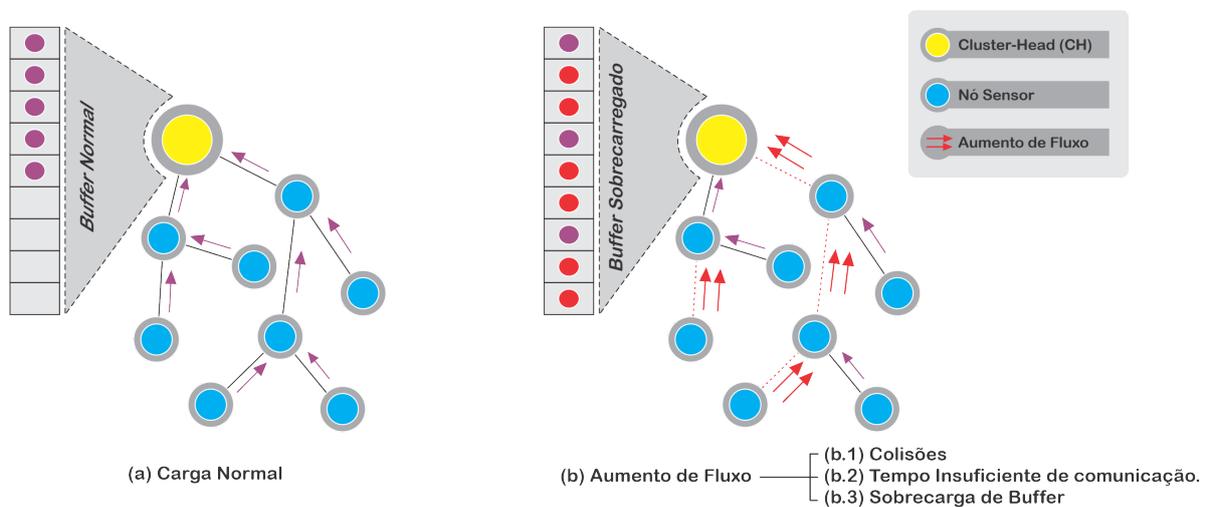


Figura 7 – Situação característica do congestionamento de um *cluster*.

De acordo com a literatura acerca dos mecanismos de controle de congestionamento, existem classificações quanto ao tipo de problema e as abordagens exploradas. Acerca dos problemas, o controle de congestionamento pode ser: baseado na alocação de recursos, no tráfego de dados ou em ambos (híbrido). No método fundamentado no tráfego, os esquemas gerenciam os diferentes fluxos (*upstream* e *downstream*), comumente correlacionados com a sincronização de atividades dos nós sensores na rede. Com a utilização de *beacons* e algoritmos de escalonamento, tal abordagem torna-se interessante para o caso de sobrecarga de dados, oferecendo vantagens na diminuição de atrasos. Em contrapartida, os esquemas baseados em alocação exploram os recursos de rede inoperantes para equilibrar a carga de tráfego sempre que o congestionamento surgir, tornando-se atraente em aplicações de tempo-real, sem comprometer a vida útil da rede (ZHENG; JAMALIPOUR, 2009).

Os mecanismos para controle de congestionamento, são categorizados em: contínuo, baseado em demanda, disparado por eventos e híbrido. No mecanismo contínuo, a atuação é realizada pelo nó *sink*, de forma contínua. Entretanto, além da rede permanecer ocupada durante a detecção, causando congestionamento, tal método pode propiciar ajustes

desnecessários. No método baseado em demanda, os ajustes são calculados e armazenados pelo coordenador a cada salto, enviando somente em caso de pedidos do nó *sink* para atuação na rede. Isto pode afetar bruscamente os recursos energéticos dos dispositivos. Na abordagem disparada por eventos, utilizada nesse trabalho de mestrado, a ideia segue basicamente 3 etapas: detecção do evento, notificação e reajuste. Embora mais complexo, isto pode reduzir congestionamentos de forma proativa, a partir da detecção e envio de respostas com ajustes precisos. Por último, o método híbrido pode combinar um ou mais métodos simultaneamente.

2.2.2 Métodos para Detecção de Eventos em RSSFs

Em diferentes aplicações, principalmente em RSSFs de larga escala com implantação em ambientes de monitoramento e controle, nós sensores geram diferentes fluxo de dados baseados no modelo de fluxo contínuo, com a condição prévia de que todos os nós estejam ativos e envolvidos na coleta e transmissão de dados (WU; HU; WANG, 2010). Dessa forma, eventos são gerados em diferentes pontos da rede conforme alterações são percebidas pelo sensores.

É importante salientar que, antes de lidar com eventos, inicialmente a rede deve implementar estruturas eficientes para detectar e notificar um determinado comportamento atípico. Tal comportamento pode ser motivado por diversos fatores, categorizando os eventos em: internos e externos. Os eventos internos estão relacionados com anomalias dos dispositivos, como por exemplo, esgotamento de energia causada por atividades desordenadas do nó ou perda da conexão por má implantação em áreas de difícil acesso etc; já acerca dos eventos externos, são aqueles gerados por fenômenos físicos da natureza, como: temperatura, umidade, luz e pressão. Isto corrobora a variedade de aplicações e formas de gerenciamento nos quais os eventos podem ser inseridos.

Em Redes de Sensores sem Fio, a detecção de eventos pode ser utilizada em diversos cenários de aplicações e desempenha um papel importante onde busca identificar, através de mecanismos de controle, o momento em que o evento de congestionamento ocorreu. À vista disso, várias técnicas são propostas (BAHREPOUR et al., 2009; PEI et al., 2014). Segundo Yin, Hu e Yang (2009) as abordagens podem ser classificadas como:

1. **Baseada em limites:** a detecção ocorre quando a leitura do sensor ultrapassa um valor limite pré-definido gerando uma notificação. As escolhas dos valores limites ficam a critério do usuário e quando há ocorrência de um evento este é informado sobre a detecção fora dos limites estabelecidos.
2. **Baseadas em padrão:** detecta eventos nas leituras dos nós através de técnicas de padrão de correspondência espaço-temporal. Um evento é detectado quando um

padrão especificado pelo usuário corresponde instantaneamente a um dado recente do nó.

3. **Baseada em aprendizagem de máquina:** voltada para detecção de eventos, tem sido proposta para modelar dependência espaço-temporal entre os dados dos nós, fazendo inferência probabilística sobre os eventos. Nessa categoria, algumas técnicas de *Deep Learning* são utilizadas para formular restrições temporais no processo de estimação.

Segundo [Andrade et al. \(2019\)](#), a compreensão das abordagens para detecção de eventos permite escolher a mais adequada ao cenário de RSSFs. Dentre as abordagens citadas, a baseada em limite e a baseada em padrão necessitam de informações pré-definidas para realizar a detecção. Em contrapartida, a abordagem baseada em aprendizagem de máquina não necessita de informações prévias, sendo bastante promissoras porque as correlações espaço-temporais podem ser explicitamente modeladas para lidar com as incertezas inerentes dos dados do sensor, criando seus modelos por meio da probabilidade, estatísticas e agrupamentos.

Esses mecanismos de controle de congestionamento disparados por eventos podem reportar informações de quais nós necessitam de ajustes, para posteriormente atuarem sobre eles. Dessa forma, esta abordagem compreende quatro etapas: detecção, notificação, compactação e reconfiguração da rede.

A etapa de detecção está relacionada com o tipo de congestionamento que, levando em conta a capacidade de armazenamento e qualidade do sinal sem fio, pode determinar variáveis como: taxa de entrega de pacotes, quantidade de descartes, tamanho de *buffer*, intensidade do sinal de rádio, entre outros. Segundo [Shah, Gorty e Phirke \(2011\)](#), para reportar a informação de congestionamento aos nós coordenadores, a notificação pode ser classificada em dois tipos: implícita e explícita. Na forma implícita, a notificação é enviada no cabeçalho do quadro de dados, trafegando juntamente com os dados. Já na forma explícita, os nós enviam quadros especiais de comando.

Não é viável que, antes de lidar com eventos, a rede não possua esquemas para detectar e notificar congestionamentos. Em razão disso, foi observado que existem algumas técnicas propostas baseadas em limites, padrão e aprendizado de máquina. Além disso, e recomendado na RFC 7567 ([BAKER; FAIRHURST, 2015](#)), ativos de rede em geral (representados por CHs em casos de redes *cluster-tree*) podem realizar o gerenciamento de ativo de filas (*Active Queue Management - AQM*), que descarta, de forma inteligente, pacotes dentro de um *buffer* associado a um controlador de interface. Diante de um *buffer* cheio ou quase completo, o risco de perda aumenta e medidas precisam ser tomadas para politizar o aceite de novas mensagens de forma a reduzir o congestionamento. Com esse objetivo o nó coordenador poderá utilizar vários algoritmos da literatura, dentre eles: o

ECN (*Explicit Congestion Notification*) e o RED (*Random Early Detection*).

Segundo [Mundada, Desai et al. \(2016\)](#) e [Yang et al. \(2018\)](#), a técnica RED (comumente usada no controle de filas de roteadores para gerenciar caches) foi introduzida a RSSFs para evitar o congestionamento, baseado na definição e regulação de *thresholds* que orientam a política de descarte, diminuindo a sobrecarga do *buffer* e melhorando o desempenho. Já a técnica ECN, como definido na RFC 3168 ([RAMAKRISHNAN et al., 2001](#)), é uma técnica para controle de congestionamento assistida pela rede, onde permite a notificação do congestionamento com suporte à camada de transporte, que pode ser adaptada a RSSFs através da marcação de pacotes ao invés de descartá-los, usando os bits ECN no cabeçalho do quadro, por exemplo ‘00’ a ‘11’ ([KUZMANOVIC, 2005](#)).

Em relação às técnicas de reajuste, os mecanismos podem implementar abordagens do tipo: *hop-by-hop*, *end-to-end* ou híbrido. No método *hop-by-hop* está relacionado a uma atuação local, realizada pelo próprio CH dentro do seu *cluster*, através de adaptações de parâmetros de acesso ou realizando tarefas de agregação, filtragem ou fusão de dados. A agregação tem como foco a redução de dados de qualquer tipo de fonte, inclusive dos sensores. Enquanto que a filtragem remete a uma seleção de dados previamente selecionada. E por ultimo a fusão lida com a junção de dados provenientes de qualquer sensor. Já com relação ao método *end-to-end*, diferentemente do método anterior, o ajuste é referenciado pelo coordenador PAN, que possui informações globais e com isso detém uma visão mais abrangente da rede ([JAN et al., 2018](#)).

2.3 Compressão de Dados

A Compressão de Dados (CD) é o processo no qual uma dada 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, a largura de banda e o tempo de ocupação de um sistema de comunicação, a energia usada na transmissão e transporte de dados, entre outros ([SHRAVANA; VEENA, 2017](#)).

A CD conta, principalmente, com dois processos básicos, o de compactação que transforma os dados originais em uma forma comprimida, e o de descompactação, que permite reconstruir os dados originais ou algo próximo deles. De uma forma geral, esses processos de compressão podem ser classificados em: com perdas ou sem perdas.

Na primeira abordagem (compressão com perdas), os dados do arquivo original não podem ser totalmente recuperados após a compressão. Como resultado, o tamanho do arquivo é permanentemente reduzido, eliminando dados redundantes. Essas técnicas são algumas vezes chamadas como compressão irreversível já que é impossível recuperar os

dados anteriores ao processo de compressão. Eles são usados em aplicações que aceitam um grau de erro ou que, pela natureza dos dados, contam com informação e amostras redundantes que, ao serem eliminadas ou representadas de uma maneira comprimida, não afetam o funcionamento. Exemplos de aplicações para esse tipo de compressão são a codificação de áudio e vídeo, a compressão de imagens, a coleta de dados provenientes de sensores, entre outros.

Em contraste, na outra abordagem (compressão sem perdas), todos os dados originais do arquivo compactado podem ser recuperados após descompressão. Esses algoritmos permitem uma reconstrução exata dos dados na etapa de descompressão e são normalmente usados em aplicações e situações que querem uma integridade absoluta da informação, como documentos de texto, planilhas, registros de bancos de dados, entre outros. (HAMDAN; AWAIAN; ALMAJALI, 2019).

As tecnologias estão se expandindo rapidamente, o que causa uma aumento na quantidade de dados gerados e das necessidades de armazenamento e transmissão de informação. Atualmente, a taxa de crescimento de dados gerados é maior que a de tecnologias de transmissão e armazenamento. Por conta disso, as técnicas de CD estão se posicionando como uma possível solução para reduzir esta tendência, já que permitem um maior armazenamento de informações na mesma quantidade de espaço e a redução do tempo de uso do canal de comunicação, bem como consumo energético. (JAYASANKAR; THIRUMAL; PONNURANGAM, 2021)

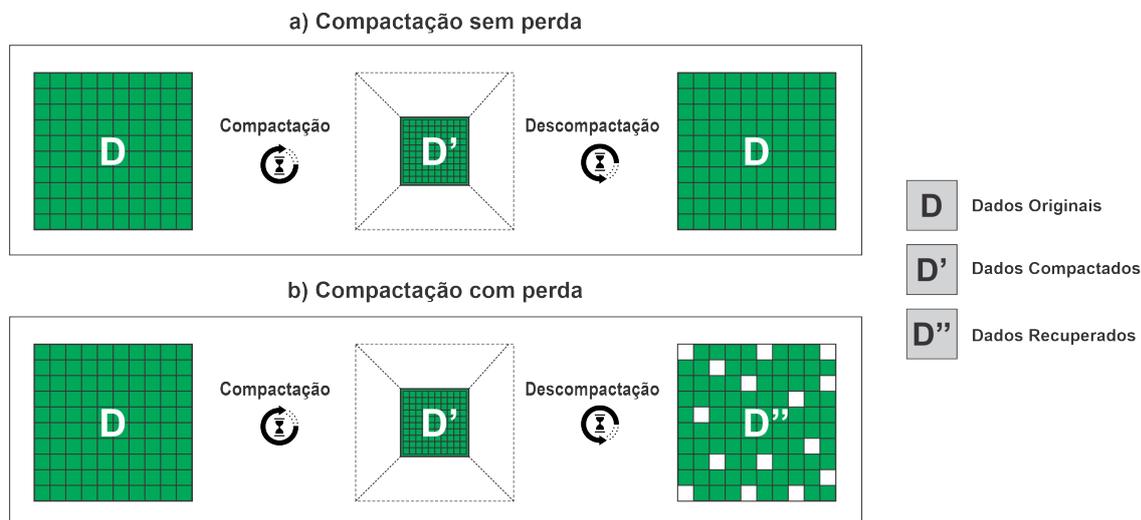


Figura 8 – Compressão sem perda (a) e com perda (b).

2.3.1 Swinging Door Trending (SDT)

O *Swinging Door Trending (SDT)* é um método de CD que usa uma linha de tendência para representar 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 linha atual para ser representado na mesma (BRISTOL, 1987; BRISTOL, 1990).

Na Figura 9 pode-se visualizar um exemplo de SDT utilizando seis amostras (representadas por P). O mecanismo cria uma área de cobertura no formato de um paralelogramo. Essa área cresce com relação ao primeiro ponto da linha de tendência atual ou ponto computado (C), o qual é composto por uma dupla (x,t), sendo x o valor atual (dado capturado pelo sensor, por exemplo) e t o tempo (sequencia dos dados capturados, por exemplo). A área tem uma dimensão longitudinal de duas vezes o DC tendo no centro inicial o (C). O paralelogramo tem um pivô superior (PS) e outro inferior (PI) com uma distância de +DC (que corresponde do ponto C até o PS) e -DC (que corresponde do ponto C até o PI). Podemos também visualizar uma borda superior (BS) e uma inferior (BI). A BS só pode movimentar-se no sentido anti-horário e o BI no sentido horário. Quando um novo dado chega, estes limites podem movimentar-se para inseri-la na área, o que somente é possível quando o coeficiente angular da reta ou inclinação s do BS for menor ou igual ao do BI. Quando esta condição não é cumprida, um novo ponto é criado dentro da área para representar o último ponto de linha de tendência. Esse mesmo ponto é usado como o novo C, e o processo começa novamente (NETO et al., 2014).

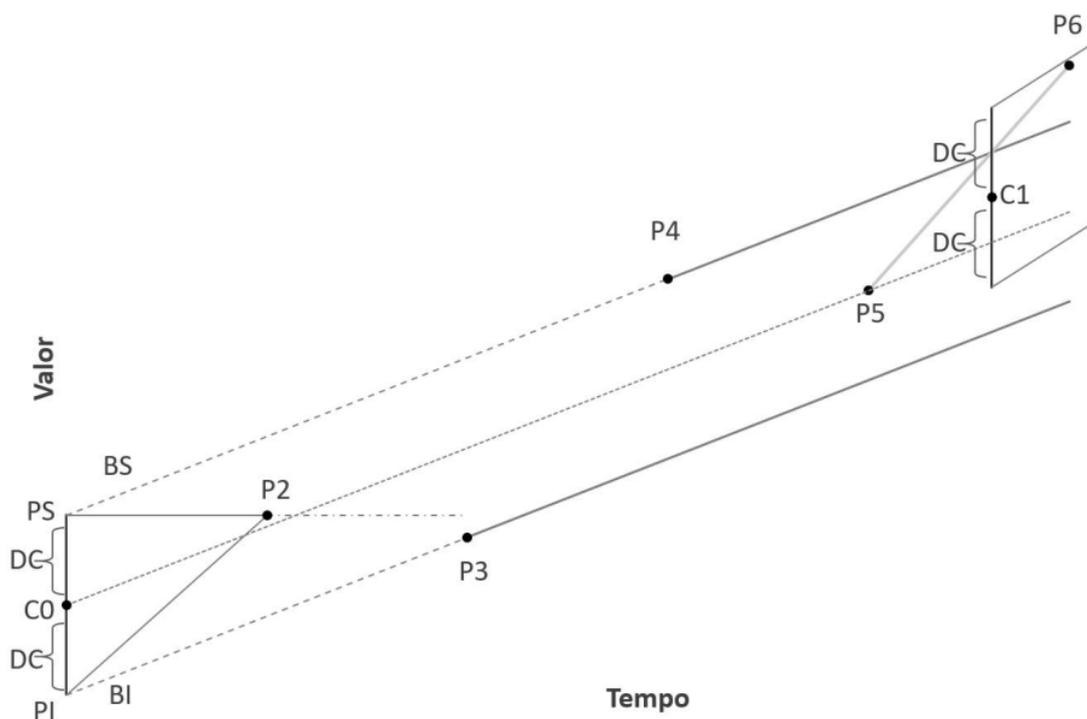


Figura 9 – Representação da Patente SDT.

Fonte: Bristol (1987)

Segundo a patente do SDT (BRISTOL, 1987), este conta com oito passos apresentados abaixo:

1. Receber o primeiro ponto.
2. O pivô superior e inferior são estabelecidos.
3. Receber o ponto seguinte.
4. Calcular as inclinações atuais em relação ao pivô superior e inferior.
5. Comparar os declives atuais com os casos de declives extremos ($S_{max\ s}$, $S_{min\ i}$). Se $S_{max\ s}$ é maior que $S_{min\ i}$, o algoritmo continua no sexto passo; caso contrário, regressa ao terceiro.
6. Localizar o ponto final dentro da área do paralelogramo. A inclinação entre este e o ponto atual é calculada, e a borda cruzada é ajustada para ser paralela à outra. Uma interseção entre a borda cruzada e a inclinação do ponto final e atual é calculada e definida como o novo valor de c .
7. Emitir o ponto final c .
8. O novo ponto inicial c é usado como início do paralelogramo, as inclinações extremas são restauradas, os pivôs são novamente definidos e o ponto atual é avaliado. Finalmente, o algoritmo volta ao terceiro passo.

2.3.2 Codificação Delta

A codificação delta é uma técnica de compactação sem perdas, ela precisa de relativamente menos tempo de compactação devido à sua abordagem simples. Conforme mostrado na Figura 10, as diferenças entre todas as amostras consecutivas em um conjunto de dados são calculadas. A amostra original no conjunto de dados é substituída por sua diferença com a próxima amostra. No conjunto de dados compactado delta resultante, a primeira amostra será a primeira amostra do conjunto de dados original. O resto das amostras são a diferença das amostras consecutivas.

Como método de compactação de dados, eles reduzem os requisitos de memória e a quantidade de dados a serem transmitidos por meio de linhas de dados (requisitos de largura de banda) ao processar dados correlacionados, como dados sequenciais.

Como exemplo, podemos citar o armazenamento de valores de bytes como diferenças (deltas) entre valores sequenciais, em vez dos próprios valores. Portanto, em vez de 17, 19, 18, 21 e 21, seria armazenado 17, 2, -1, 3, 0. Isso reduz a variância (intervalo) dos valores quando as amostras vizinhas são correlacionadas, permitindo um menor uso de bits para os mesmos dados. A Figura 10 ilustra o funcionamento da codificação.

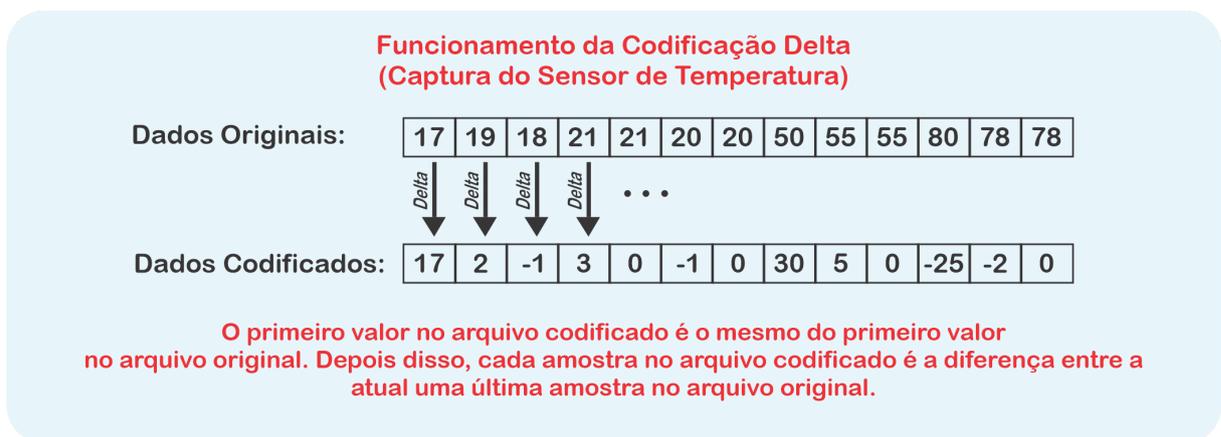


Figura 10 – Compressão Delta.

3 Trabalhos Relacionados

Este capítulo apresenta os protocolos, algoritmos e mecanismos de compactação mais importantes acerca do controle de congestionamento em RSSFs de larga escala. Foi sintetizado algumas literaturas em concordância com os tipos de problemas para os quais são guiados: soluções de compressão com e sem perdas, técnicas utilizadas, onde podem ser aplicados, soluções adaptativas de algoritmos e modelos que buscam aparar arestas, cujo objetivo é minimizar a carga de dados a serem transmitidas. De acordo com [Yu et al. \(2020\)](#), existem muitas técnicas de compressão universais, como compressão Delta, compressão Huffman (é um método de compressão que usa as probabilidades de ocorrência dos símbolos no conjunto de dados a ser comprimido para determinar códigos de tamanho variável para cada símbolo, dessa forma reduzindo a quantidade de dados a serem enviados), codificação aritmética (é uma forma de codificação que utiliza número fixo de bits por caractere e é utilizada na compactação de dados sem perdas) e codificação baseadas em aprendizado de máquina, redes neurais e lógica *fuzzy* (usam parâmetros baseados em dados de treinamento e à medida que os dados de treinamento se expandem para representar o mundo de modo mais realista, o algoritmo calcula resultados mais precisos).

Na Seção 3.1, busca-se apontar esquemas que lidam com soluções que envolvem técnicas tradicionais de compressão, como por exemplo, a utilização das codificações Huffman e Delta, entre outros algoritmos. Já na seção 3.2, mostramos propostas estado-da-arte fundamentadas no problema de compressão e que utilizam sistemas inteligentes como aprendizado de máquina (*Deep Learning*) e redes neurais (*Neural Networks*). Por último, na Seção 3.3, ressaltamos os trabalhos mais significativos sobre compressão utilizando uma abordagem híbrida, salientando os pontos positivos e negativos em comparação ao mecanismo proposto neste trabalho.

3.1 Algoritmos Tradicionais de Compressão

Com relação a trabalhos que focam na implementação de mecanismos tradicionais de compressão, a fim de fornecer uma redução da quantidade de dados enviados e armazenados, [Correa et al. \(2019\)](#) propuseram um esquema com perdas baseado no *Swinging Door Trending*, em que foi criado 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. No entanto, o esquema não é recomendado em situações que requerem dados em tempo real. Seguindo essa mesma linha, [Neto et al. \(2014\)](#) utiliza o *Swinging Door Trending* para comprimir os

dados, no entanto, ele propõe um novo algoritmo que consiste numa adaptação, em que seus principais parâmetros são ajustados dinamicamente através da análise de tendências do sinal em tempo real. Porém, a solução não cita quais parâmetros do algoritmo ASDT são mais adequados para uma maior variedade possível de sinais que possuem comportamentos apropriados. Outro estudo com a utilização de um algoritmo já existente e propondo melhoramento, [Abdulzahra, Al-Qurabat e Idrees \(2021\)](#) utilizou uma técnica de redução de dados, sem perdas, baseada em compressão *Compression-Based Data Reduction* (CBDR) e o algoritmo *Lempel-Ziv-Welch* (LZW), onde propôs uma melhoria reduzindo o número total de dados enviados para o gateway e o processamento necessário. Com isso os autores conseguiram reduzir a quantidade de dados transmitidos e preservar a vida útil dos nós na rede.

Alguns trabalhos têm apresentado um estudo comparativo entre algoritmos de compactação como forma de buscar solução para resolver problemas acerca da transmissão de informações pelos nós em meio ao alto fluxo de dados em ambientes IoT. [Bose et al. \(2016\)](#) faz um comparativo de algoritmos de compressão com perdas. Dentre eles podemos citar: *Critical Aperture* (armazena os valores que envolvem uma mudança significativa de inclinação quando comparado a um valor pré-configurado), *Fractal Resampling* (reivindicações para reter todos os pontos importantes, com base no deslocamento do ponto médio), *Chebyshev Transform* (usa aproximação polinomial de Chebyshev e é usado pela NASA) e *Wavelet Transform* (associa informações de domínio de frequência com sua localização em tempo usando uma abordagem de multi-resolução). Foi observado que as características dos dados têm grande impacto na eficiência dos algoritmos. Portanto, a identificação do algoritmo mais adequado vai depender das características específicas do sinal. Entretanto, esse esquema não considera nenhum algoritmo eficiente que possa ser utilizado como padrão para ambientes em que possuem dados diversificados no seu fluxo de rede.

Outro mecanismo sugerido, utilizando balanceamento de carga, foi proposto por [Aziz et al. \(2019\)](#) no qual propõe um algoritmo de balanceamento de carga de tamanho de cluster para o *Cluster Size Load Balancing for CS (CSLB-CS)* no qual poderia atingir a utilização ideal do método *Compressive Sensing (CS)* — 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 (representando através dos seus dados a leitura dos sensores), melhorando o consumo geral de energia, total de número de dados transmitidos e erro de reconstrução. Todavia, o autor não menciona se, após a compactação, os dados terão perdas que possam ser significativas a depender das características dos dados contidos no sinal.

Já o trabalho de [Le e Vo \(2018\)](#) 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. Como os códigos ocupam menos espaço do que as strings que substituem, obtemos compressão* —, cuja 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, em que o objetivo final é reduzir a quantidade de espaço gasto para seu armazenamento e principalmente transmissão. Existem algumas limitações ou deficiências no modo de codificação, entre elas podemos citar a capacidade de armazenamento limitada. Portanto, esse algoritmo não é recomendado em situações que envolvam uma grande quantidade de dados.

Também visando proporcionar uma compactação dos dados para que possam ser transmitidos e armazenados, podemos encontrar na literatura vários trabalhos que utilizam como base a codificação Delta e/ou codificação *Huffman*. Em [Stojkoska e Nikolovski \(2017\)](#) é sugerido 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. À vista disso, a quantidade de dados a serem transmitidas é bem menor já que não é necessário enviar uma leitura completa. Seguindo uma ideia análoga ao algoritmo LZW, no que se refere a substituição de um padrão por identificadores, [Chatterjee, Shah e Hasan \(2018\)](#) utiliza a compressão de dados sem perdas baseado em codificação *Huffman*. A proposta é substituir dados redundantes por identificadores, utilizando Matriz de Adjacência de Grafos (uma das formas de se representar um grafo) de forma que ao encontrar a ocorrência comum de padrões de nós e links, é realizado a substituição por identificadores, reduzindo assim a quantidade de dados a serem armazenados e transmitidos.

Todavia, os trabalhos não deixam claros a exigência mínima dos requisitos de memória e processamento necessários a serem considerados pelos dispositivos IoT.

3.2 Algoritmos Baseados em Sistemas Inteligentes

Em referência à literatura que destacam a utilização de mecanismos baseados em sistemas inteligentes, como Redes Neurais, Lógica Fuzzy e Aprendizado de Máquina (*Deep Learning*), [Park, Park e Choi \(2018\)](#) propôs fazer a compressão de dados industriais com perdas de dados, utilizando regressão neural em um vetor representativo com compressão *lossy* (com perda). No método de compressão convencional, todos os dados são comprimidos ou mantidos por uma função, dependendo da escolha, pode ser com ou sem perdas de dados. No caso, foi sugerido um método para vetorizar pedaços divididos em um determinado intervalo e combiná-los em um vetor depois de todos eles serem vetorizados. Diante disso,

os dados futuros puderam ser previstos por meio da tendência dos vetores através do método divisão e conquista, no qual o processo de combinar vetores de alcance específico em um único vetor foi realizado por meio de técnicas de Aprendizado de Máquina (*Deep Learning*).

Nessa mesma linha, podemos também destacar o trabalho do [Signoretto et al. \(2021\)](#) que disponibilizam uma nova abordagem chamada de *Tiny Anomaly Compressor (TAC)*, baseada na Análise de Dados de Tipicidade e Excentricidade (TEDA), na qual utilizam aprendizado de máquina em dispositivos minúsculos e computacionalmente restritos. A ideia principal é focar na redução e otimização de modelos existentes, como Redes Neurais Artificiais (RNAs), além de utilizar equações recursivas, permitindo assim um cálculo eficiente com um baixo custo computacional, usando pouca memória e poder de processamento.

Já [Azar et al. \(2019\)](#) apresentam uma abordagem de redução de dados com eficiência energética baseada em um compressor rápido com perdas limitadas por erro antes da transmissão. Posteriormente, os dados transmitidos são reconstruídos em um nó de borda e o processamento é feito utilizando técnicas de aprendizado de máquina supervisionado. [Noura, Salman e Couturier \(2021\)](#) utilizam um modelo de super-resolução *deep learning* que é aplicado para recuperar imagens de alta qualidade (no lado do servidor) dadas como imagens degradadas de entrada com uma alta relação de compressão (no lado do remetente). Dessa forma, o principal desafio era preservar a qualidade da imagem usando uma alta taxa de compressão. O modelo de aprendizagem *Residual Dense Network (RDN)* empregado conseguiu melhorar de forma significativa a qualidade dessas imagens. Entretanto, o modelo foi testado apenas com imagens JPEG e BPG e não foram utilizados outros tipos de dados.

Os autores [Jindal, Kumar e Patidar \(2021\)](#) propuseram uma abordagem que compacta o fluxo de dados em tempo real utilizando a codificação de verificação de paridade de baixa densidade (*Low-Density Parity-Check - LDPC*). Pode ser aplicado a fluxos de dados que são lineares por natureza, ou seja, não mudam muito e mudanças repentinas nos dados são incomuns. A técnica de compressão LDPC também pode ser aplicada em dados criptografados, sendo então benéfico em ambientes cuja privacidade e confidencialidade são essenciais. Para provar a eficiência do código LDPC, alguns parâmetros foram usados e analisados: tempo de transmissão, taxa de redução do tempo de transmissão e tempo de compressão. Como resultado, o tempo de médio de transmissão foi reduzido em 45% utilizando a codificação LDPC. Por conta da pluralidade das aplicações IoT e diversidade de tipos de dados, o estudo foi limitado a conjunto de dados lineares.

No trabalho de [Shanthamallu et al. \(2017\)](#) é feito um levantamento dos conceitos e algoritmos básicos usados para *Machine Learning* e suas aplicações. É possível utilizar métodos supervisionados (regressão linear — *é uma técnica estatística para estimar a*

relação entre as variáveis de entrada e de saída — e logística — é uma forma estatística de modelar um resultado binomial —, máquinas de vetor de suporte — são um dos modelos populares de aprendizado supervisionado, usados principalmente para classificação binária, bem como classificação multiclasse —, classificadores *Naive Bayes* — são classificadores probabilísticos simples — e k-vizinhos mais próximos — é um dos algoritmos de aprendizado de máquina supervisionado mais simples —) e não supervisionados (*clustering* — trata de encontrar uma estrutura ou padrão em uma coleção de conjuntos de dados não rotulados — e quantização de vetor — organiza os dados em vetores e os representa por seus centróides —) e paradigmas de aprendizagem profunda — é um ramo do aprendizado de máquina que ganhou popularidade recentemente, capaz de aprender vários níveis de abstração —. Apesar de citar uma extensa bibliografia de algoritmos de máquina e suas aplicações que podem ser utilizados em ambientes IoT, o trabalho não cita diretamente sobre a compactação dos dados e como o alto fluxo de informações e necessidade de processamento dos nós podem influenciar no desempenho e consumo de energia desses dispositivos. Na mesma linha investigativa, [Azar et al. \(2020\)](#) fizeram uma pesquisa de efeitos de técnicas de compressão *lossy* na tarefa de classificação de séries temporais usando redes neurais profundas no campo de *Machine Learning*. Os autores propuseram uma abordagem eficiente de compressão para séries temporais univariadas (*refere-se às séries temporais que consistem de observações únicas (escalares) registradas sequencialmente durante incrementos iguais de tempo*) e multivariadas (*consiste de várias séries únicas que podemos chamar de componentes*) combinada com qualidade de dados. Por fim, [Yao et al. \(2018\)](#) apontam que um dos principais impedimentos na implantação de redes neurais profundas em dispositivos IoT, reside na alta demanda de recursos de modelos de rede neural profunda treinados. Embora os algoritmos de compressão baseados em redes neurais existentes possam reduzir efetivamente o número de parâmetros de modelo, nem todos esses modelos levam a representações matriciais que podem ser implementadas eficientemente em dispositivos IoT.

3.3 Algoritmos Híbridos

Dentre os principais trabalhos voltados à compactação de dados em ambientes IoT, na qual temos uma abordagem híbrida (combinação de duas ou mais técnicas/soluções de compactação), podemos destacar [Deepu, Heng e Lian \(2016\)](#) que na sua proposta, os dados são compactados com técnicas com e sem perdas, de modo a permitir uma transmissão híbrida sem perdas com suporte à seleção adaptativa de taxa de dados e economia de energia na transmissão sem fio. Os dados foram compactados primeiro usando uma técnica de compressão com perdas com uma alta taxa de compressão. O erro residual entre os dados originais e os dados descomprimidos são preservados usando codificação de entropia (*é usada para comprimir qualquer tipo de dados, e pode ser dividida em duas*

partes: modelamento e codificação. O modelamento atribui probabilidade a símbolos, e a codificação gera uma sequência de bits a partir destas probabilidades), permitindo uma restauração sem perdas dos dados originais quando necessário.

Já Hanumanthaiyah et al. (2019) propuseram a implementação de RLE (*Run Length Encoding*) e técnicas de compressão sem perdas Delta que precisam de um menor tempo de compactação para compressão de dados. Os resultados do teste provam que a combinação da técnica Delta e RLE tem uma melhor taxa de compressão do que apenas a técnica de compressão Delta. No entanto, a eficácia da técnica diminui no decorrer de muitos picos de variabilidade dos dados capturados pelos sensores.

Al-Qurabat, Jaoude e Idrees (2019) propõem uma técnica de redução de dados em duas camadas TTDR (*Two-Tier Data Reduction*) na qual trabalham em duas camadas de rede que são: nós sensores e no gateway. No nível do nó do sensor foi utilizado métodos simples e adequados de compactação de dados para nós sensores IoT restritos. As técnicas exploram a correlação temporal nos dados dos sensores e usam a Codificação Delta seguida de RLE (*Codificação Run-Length*). No nível gateway foi aplicado a clusterização hierárquica para agrupamento de conjuntos de dados recebidos de nós de sensores dependentes do princípio MDL (*Minimum Description Length*) — é um método poderoso de inferência indutiva, a base da modelagem estatística, reconhecimento de padrões e aprendizado de máquina.

Em Saidani, Jianwen e Mansouri (2020) é proposto uma nova abordagem de compressão sem perdas baseada na codificação Delta e em dois algoritmos de resolução de caracteres T-RLE (*Optimization of the RLE (Run Length Encoding)*). O T-RLE é uma proposta que melhora do RLE em que altera o formato dos dados entregues após a compactação. A ideia principal é aumentar a similaridade dos dados aplicando a codificação delta e compactá-los usando o algoritmo T-RLE. Portanto, no primeiro momento, o algoritmo Delta (em que é verificada a diferença entre as amostras sucessivas) é utilizado na fase de processamento para tornar os dados mais convergentes, resultando assim num aumento da taxa de similaridade nos conjuntos de dados. No segundo momento, é aplicado uma proposta do RLE modificado (T-RLE) nos dados na qual ocorre a substituição de caracteres repetidos por número de ocorrências. A técnica demonstrou eficácia na redução do tamanho dos dados a serem transmitidos. No entanto, não foi citado quais dados foram utilizados nos cenários.

Também visando proporcionar compactação de dados em ambientes IoT utilizando técnicas híbridas, Al-Qurabat, Mohammed e Hussein (2021) sugeriram resolver esse problema com um algoritmo de compressão simples e sem perdas baseado na codificação diferencial (DE) e técnicas de *Huffman*, que é particularmente eficaz para nós IoT. Os autores propuseram um gerenciamento de tráfego de dados baseado em técnicas de compressão e MDL (*Minimum Description Length*) que funciona no nível de nós sensores

e no nível de *gateway* de borda. No primeiro nível, um algoritmo de compressão leve e sem perdas baseado em *Differential Encoding (DE)* e técnicas de *Huffman* foram usadas para operar nos nós IoT restritos. Esses métodos exploram a correlação temporal entre os dados do sensor. No segundo nível, o princípio de MDL com agrupamento hierárquico foi utilizado com o propósito de agrupar os conjuntos de dados provenientes do primeiro nível. Qualquer par de conjuntos de dados que podem ser compactados de acordo com o princípio MDL são combinados em um grupo. Como resultado desta estratégia, o número de conjuntos de dados será diminuído gradualmente e o processo de mesclar conjuntos semelhantes em um único *cluster* é interrompido se não houver mais pares de conjuntos que possam ser compactados.

Por fim, um ADCS (*Adaptive Data Compression Scheme*) foi proposto por Al-Kadhim e Al-Raweshidy (2021) no qual o objetivo era controlar eficientemente a taxa de compressão do dispositivo IoT e o consumo de energia na rede IoT baseada em nuvem. O esquema pode ser resumido da seguinte forma: o projeto de uma rede IoT é baseado em nuvem usando um modelo MILP (*Mixed Integer Linear Programming*); a redução do consumo de energia da rede é feita através da ADCS usando a otimização do MILP; ocorre uma distribuição de tráfego pelos *gateways* para evitar congestionamentos na nuvem na rede IoT; as demandas de tráfego são resolvidas através de esquemas de compactação de dados para, dessa forma, maximizar a vida útil da rede, minimizar o número de dispositivos ligados e otimizar a seleção de dispositivos IoT que tenham a energia mínima por bit e potência ociosa (em joules) e os resultados mostraram uma economia de energia de 33% e 40% mas em comparação ao um modelo NCS (*Non-Compression Scheme*).

A Tabela 1 resume alguns dos principais algoritmos de compactação/filtragem citados nos trabalhos relacionados. Estes foram categorizados de acordo com o requisito de compressão, além da técnica utilizada e classificação implementada em cada abordagem.

Tabela 1 – Algoritmos de compressão citados no trabalho.

| Autor | Compressão | Técnica | Classificação |
|--|-------------------|--------------------------------|----------------------|
| Neto et al. (2014) | Com Perdas | Swinging Door Trending | Tradicional |
| Abdulzahra, Al-Qurabat e Idrees (2021) | Híbrido | CBDR + LZW | Tradicional |
| Correa et al. (2019) | Com Perdas | Swinging Door Trending | Tradicional |
| Le e Vo (2018) | Sem Perdas | LZW + BZip2 + GZip2 | Tradicional |
| Stojkoska e Nikolovski (2017) | Sem Perdas | Codificação Delta | Tradicional |
| Jindal, Kumar e Patidar (2021) | Sem Perdas | Codificação LDPC | Tradicional |
| Chatterjee, Shah e Hasan (2018) | Sem Perdas | Codificação Huffman | Tradicional |
| Park, Park e Choi (2018) | Com Perdas | Regressão de Rede Neural | Inteligente |
| Signoretti et al. (2021) | Com Perdas | Ajuste ao Fluxo de Dados | Inteligente |
| Azar et al. (2019) | Com Perdas | Aprendizagem de Máquina | Inteligente |
| Noura, Salman e Couturier (2021) | Com Perdas | Aprendizagem de Máquina | Inteligente |
| Deepu, Heng e Lian (2016) | Híbrido | Entropia | Híbrido |
| Hanumanthaiah et al. (2019) | Sem Perdas | Codificação Delta + RLE | Híbrido |
| Al-Qurabat, Jaoude e Idrees (2019) | Sem Perdas | Codificação Delta + RLE | Híbrido |
| Saidani, Jianwen e Mansouri (2020) | Sem Perdas | Codificação Delta + T-RLE | Híbrido |
| Al-Qurabat, Mohammed e Hussein (2021) | Sem Perdas | Codificação Huffman + DL | Híbrido |
| Al-Kadhim e Al-Raweshidy (2021) | Sem Perdas | S-LZW + S-LEC | Híbrido |
| Trabalho Proposto | Híbrido | Swinging Door Trending + Delta | Híbrido |

3.4 Considerações do Capítulo

Este capítulo apresentou diversos algoritmos e técnicas de compactação de dados em ambientes IoT a fim de economizar energia nos dispositivos e evitar sobrecarga nas redes de sensores sem fio. Dentro deste contexto, este trabalho de Mestrado propõe um mecanismo eficiente de filtragem local de dados baseado no padrão IEEE 802.15.4, que permite evitar ou postergar congestionamentos e atuar em RSSFs *Cluster-Tree* de larga escala. Para isso, propomos um módulo que possa ser utilizado numa arquitetura REST (*Representational State Transfer*) já existente (LINO et al., 2019), de forma que possa ser acionado pelo nó coordenador à medida que informações de sobrecarga forem sendo enviadas pelos nós da rede. A ideia subjacente é implementar um gerenciamento local de dados nos nós sensores e, com isso, disparar ações de reajuste local e global, caso necessário. Este mecanismo visa evitar ou postergar ocorrências de congestionamentos, diminuindo assim perdas de pacotes, atrasos de comunicação fim-a-fim e processos de reconfiguração global de rede.

4 Mecanismo de Controle de Congestionamentos Através de Filtragem Local de dados para RSSFS Cluster-tree

Esse trabalho propõe um mecanismo eficiente de filtragem local de dados baseado na detecção de eventos, a fim de evitar congestionamentos em RSSFs *cluster-tree* de larga escala baseada nas normas IEEE 802.15.4 e ZigBee. Será utilizada uma infraestrutura de rede *cluster-tree* existente que, diante de um aumento de fluxo de dados, dispara um mecanismo de reconfiguração dinâmica das estruturas de comunicação, proposta por Lino et al. (2019). O processo de reconfiguração é custoso para uma RSSF, portanto, o objetivo principal do nosso trabalho é adicionar a essa infraestrutura, um novo módulo de atuação local com o objetivo de evitar ou postergar essa reconfiguração, conforme os eventos de modificação do fluxo de dados sejam identificados. A Figura 11 apresenta o modelo arquitetural da infraestrutura existente (cor cinza), com a inclusão do novo módulo proposto (cor azul).

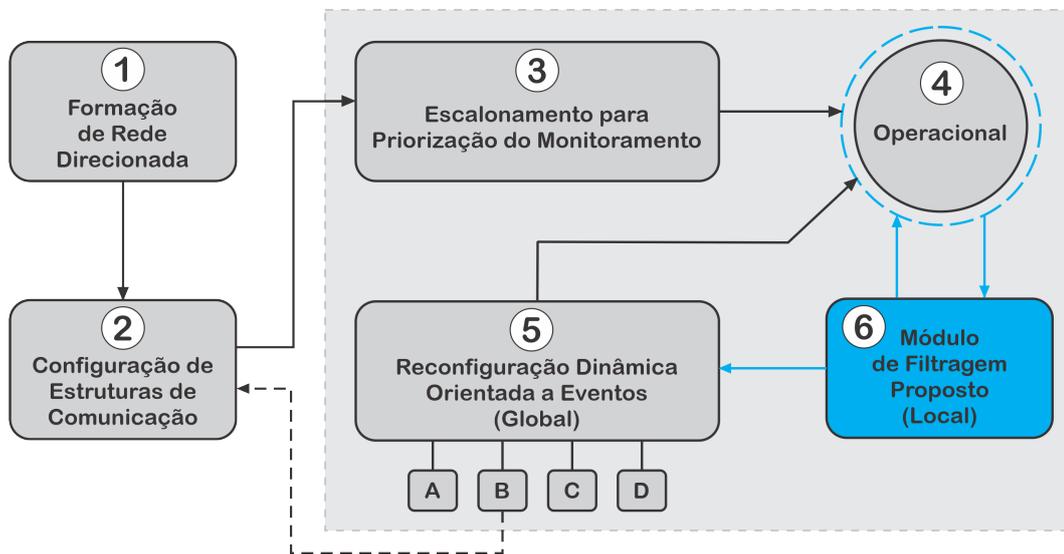


Figura 11 – A arquitetura do mecanismo de filtragem proposto para RSSFs.

Basicamente, os módulos estabelecidos segundo Lino et al. (2019) são:

- Módulo 01: Processo de formação da rede orientado a geração de dados dos nós sensores.
- Módulo 02: Configuração eficiente das estruturas de comunicação.
- Módulo 03: Escalonamento para priorização do tráfego de monitoramento (*upstream*).

- Módulo 04: Funcionamento operacional da rede com a finalidade de detectar possíveis eventos de modificação de carga de dados.
- Módulo 05: Mecanismo de reconfiguração dinâmica da rede orientada a eventos em caso de congestionamento, na qual possui 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).

O módulo proposto neste trabalho foi definido como:

- Módulo 06: Em caso de sobrecarga na rede, atua sobre os dados gerados pelos nós sensores, fornecendo compressão até a normalização do fluxo, evitando ou postergando uma possível reconfiguração na rede.

Nas subseções seguintes, os módulos 01, 02, 03, 04 e 05 da arquitetura existente são sumarizados e, por fim, o mecanismo proposto neste trabalho é apresentado (Módulo 06).

4.1 Módulo 01: Processo de Formação da Rede Orientada.

A topologia de rede é baseada num ambiente em que o processo de formação topológica está fundamentada na associação de nó sensores e definição de *cluster-heads* (CH). É implementado um esquema de formação capaz de reduzir a complexidade de processamento operado nos CHs, considerando apenas as características de geração de dados como critério para o processo de associação dos *clusters*.

O conjunto de nós é implantado de forma aleatória num ambiente de larga escala. A formação topológica é iniciada a partir do coordenador PAN, que recebe solicitações de nós sensores com distintas características de geração de dados. Uma vez que os nós são associados, o coordenador PAN seleciona os sensores filhos como candidatos a *clusters-heads* (para construírem seus próprios *clusters*). Nesse ambiente, metade dos nós selecionados são de uma determinada característica e o restante, outra. A ideia é construir diferentes ramos com base em características distintas de acordo com sensores do ambiente. Dessa forma, é gerado sub-redes mais homogêneas, permitindo assim implementar mecanismos otimizados de alocação de *superframe*.

Posteriormente, após a criação do *cluster* do coordenador PAN, novos *clusters* são gerados pelos CHs baseados nas mesmas características, de forma que apenas nós sensores com as mesmas características de geração de dados sejam aceitos. Neste trabalho, foi considerado duas características diferentes de geração de dados, relacionadas à grandeza física monitorada do ambiente, não havendo impedimento para utilização de uma quantidade maior delas (por exemplo, 3 ou 4, de acordo com o tipo de aplicação proposta).

É importante frisar que cada ramo da árvore contem apenas nós sensores com a mesma característica de geração de pacotes. Esse processo de associação é repetido até que todos os nós tenham sido cobertos.

Esse processo tem o objetivo de formar redes *cluster-tree* mais direcionadas, otimizadas e homogêneas, com uma menor quantidade possível de *clusters* e melhor gerenciamento de ramos, o que possibilita um cenário mais apropriado para a definição das estruturas de comunicação ativa e atuação sobre eventos de carga.

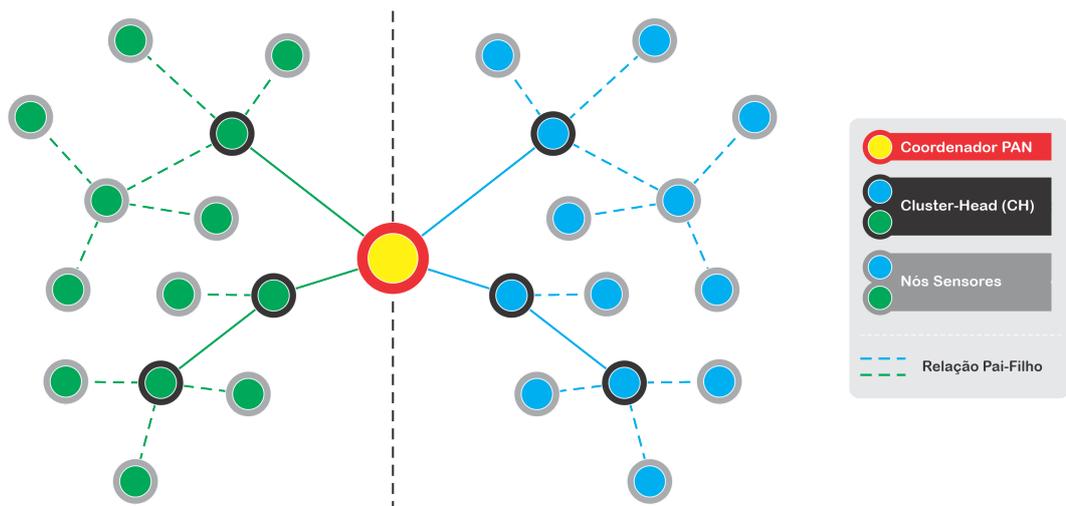


Figura 12 – Formação de Rede Orientada às características de geração de dados.

4.2 Módulo 02: Configuração eficiente das estruturas de comunicação.

Uma vez que ocorre o processo de formação de rede, a abordagem *Load-SDA* (LEÃO et al., 2017), que configura as durações de *superframes* dos *clusters*, é utilizada. A abordagem *Load-SDA* evita atrasos de comunicação, congestionamento de rede e reduz o descarte de pacotes. De forma sucinta, podemos dizer que o esquema aloca, de forma proporcional, durações de *superframes* para cada um dos CHs, baseado em seu número de filhos e sua peculiaridade acerca da geração de dados.

Esse mecanismo de alocação considera que os nós geram e enviam fluxo de mensagens para o coordenador PAN, onde cada fluxo é caracterizado por uma periodicidade de geração e pelo tamanho da mensagem. O fator de utilização imposto por cada fluxo depende da profundidade do nó sensores enquanto as mensagens fluem pela árvore. Dessa forma, o tamanho do intervalo de *beacons* deve ser grande o suficiente para que todas as durações de *superframes* sejam escalonadas dentro do menor BI possível, com o objetivo de reduzir atrasos de comunicação.

No que se refere à definição dos períodos ativos dos *clusters*, a duração do tempo para transmitir uma mensagem é feita correlacionando a camada MAC e as operações de nós sensores no mecanismo de acesso ao meio CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*). As durações de *superframe* para os *clusters* são estimadas a partir da carga imposta pelos nós sensores associados a cada um dos CHs, que depende apenas das taxas de geração de dados e do número de nós associados na ramificação.

4.3 Módulo 03: Escalonamento para priorização do tráfego de monitoramento (*upstream*).

No que diz respeito ao escalonamento utilizado, é proposto aquele que visa priorizar *cluster-heads* mais profundos sobre aqueles com menor profundidade. Antes do funcionamento propriamente dito da rede, é prudente estipular qual será o fluxo que terá prioridade. O modelo de priorização *bottom-up* leva vantagem se comparado com o modelo *top-down* uma vez que o tráfego de monitoramento leva menos intervalos de *beacons* (BI) para chegar até o coordenador PAN.

Por esse motivo, este modelo é utilizado para dar prioridade ao tráfego de monitoramento, no qual é iniciado a partir dos nós sensores, além de promover a detecção dos eventos de carga.

4.4 Módulo 04: Funcionamento operacional com a finalidade de detectar possíveis eventos de modificação de carga.

Após a configuração eficiente das estruturas de comunicação e priorização do tráfego de monitoramento, o módulo 4.4 trata da operação normal da rede. Aqui, as atividades de monitoramento dos nós sensores são iniciadas, com base em suas periodicidades. É determinado a carga de dados padrão que cada nó sensor passa então a coletar os dados processados e transmite para os nós ascendentes, seguindo o roteamento em árvore.

A rede começa a operar com o fluxo de monitoramento seguindo o modelo de escalonamento proposto no Módulo 4.3, até que algum evento de variação das taxas de dados seja identificado, o que pode com isso, gerar problemas de sobrecarga, maiores atrasos de comunicação e, por fim, congestionamento de rede.

4.5 Módulo 05: Mecanismo de reconfiguração dinâmica da rede orientada a eventos.

O módulo 4.5 descreve, as etapas executadas pelo coordenador PAN para reconfigurar uma rede *cluster-tree*, quando um evento de carga é identificado. Este módulo é subdividido em 4 submódulos (A, B, C e D) conforme visto na Figura 20.

- **Submódulo A: Detecção do Evento de Carga.**

Este submódulo descreve como os eventos são detectados pelo coordenador PAN, a partir da percepção e mudança da periodicidade da geração de dados dos nós sensores. O evento de carga é detectado através da utilização de bits do campo reservado, de tal forma que um determinado nó sensor seja capaz de informar o coordenador PAN da alteração da taxa de aquisição de dados. Ao receber os pacotes com os bits de multiplicidade alterados, o coordenador PAN poderá dar início ao processo de reconfiguração de rede, detalhado nos submódulos seguintes.

- **Submódulo B: Cálculo da Reconfiguração da Rede.**

O coordenador PAN é responsável por efetuar os cálculos necessários (baseados nos valores de BO e SO para cada CH da rede, considerando a nova carga imposta pelos nós sensores com alterações nas taxas de aquisição de dados) para reconfigurar a rede *cluster-tree*. O principal objetivo é verificar a necessidade de recalculas as principais estruturas de comunicação dos CHs (SD e BI), a fim de se evitar regiões de possíveis sobrecargas. Caso seja necessário reconfigurar a rede, o coordenador PAN deverá enviar mensagens de controle aos nós envolvidos, com as alterações nas estruturas de comunicação. Os próximos submódulos descrevem o mecanismo utilizado para a rápida disseminação dessas mensagens de controle de reconfiguração de rede.

- **Submódulo C: Criação da Janela de Oportunidade.**

Para que as mensagens de controle sejam rapidamente disseminadas, a ideia básica é alterar temporariamente o escalonamento *bottom-up* para um escalonamento *topdown*. O período pelo qual o escalonamento é definido para *top-down* é chamado de Janela de Oportunidade Top-Down (JTD). Dentro desse contexto, o coordenador PAN é responsável por criar mensagens de controle, que são propagadas pela rede, a fim de notificar individualmente cada CH do momento correto de alterar o escalonamento para *top-down* (criação da JTD), evitando assim incoerências temporais. Dessa forma, considerando que cada CH poderá identificar o momento correto do início da janela JTD, o próximo submódulo descreve os procedimentos necessários para a alteração do escalonamento *bottom-up* para *top-down*, permitindo assim a rápida disseminação das mensagens de reconfiguração de rede.

- **Submódulo D: Envio das Mensagens de Reconfiguração aos Nós Envolvidos.**

O modelo de envio de mensagens de reconfiguração aborda um esquema de escalonamento híbrido, baseado na proposta *Fast-HyBeS (Hybrid Beacon Scheduling Scheme)* (LEÃO et al., 2017). O esquema *Fast-HyBeS* define JTDs de forma periódica para a disseminação de mensagens de controle destinadas aos nós descendentes. Diferentemente do *Fast-HyBeS*, a JTD considerada neste trabalho é disparada com base na detecção dos eventos de carga identificados. Após o recebimento das mensagens de reconfiguração pelos CHs, o escalonamento *bottom-up* é novamente reestabelecido e o fluxo de monitoramento passa a ser novamente priorizado, até que um novo evento de carga seja identificado e todo o processo de reconfiguração seja novamente disparado.

4.6 Módulo 06: Novo módulo de filtragem local de dados proposto.

Com o propósito de evitar ou postergar reconfigurações na rede, o objetivo principal desta dissertação é criar um módulo que possa monitorar o fluxo de dados gerados pelos nós sensores e atuar localmente de tal forma que essa atuação seja transparente aos nós coordenadores. Para isso, este trabalho propõe a utilização de um mecanismo de filtragem local de dados baseado nos algoritmos de compressão *Swinging Door Trending (SDT)* e codificação Delta, devido suas eficácias comprovadas na literatura [1][3][5][13-15].

4.6.1 Detecção do Evento de Carga

Conforme citado em Lino et al. (2019), os eventos de carga (eventos críticos) ocorrem esporadicamente durante o monitoramento constante dos nós sensores, quando uma taxa diferente da usual é gerada. Num ambiente real, em que se utiliza aplicações IoT de áreas expostas (como agricultura de precisão, por exemplo) os eventos estão fortemente ligados à mudança de variáveis como temperatura, umidade, luz e pressão executadas pelos nós sensores. Temos na Figura 13 uma representação desse tipo de cenário, onde diversos sensores podem detectar e reportar eventos de carga no ambiente monitorado. Aqui descrevemos como os eventos são detectados pelos CHs, a partir da compreensão e modificação dos dados gerados pelos nós sensores.

O modelo de rede proposto neste trabalho 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. O fluxo inicial é tido como fluxo padrão e, na ocorrência de um evento crítico, é possível que o aumento da periodicidade gere problemas na rede. O mecanismo considerado pondera diferentes graus de multiplicidade das taxas de aquisição de dados, reproduzidos através de *bits* reservados no cabeçalho

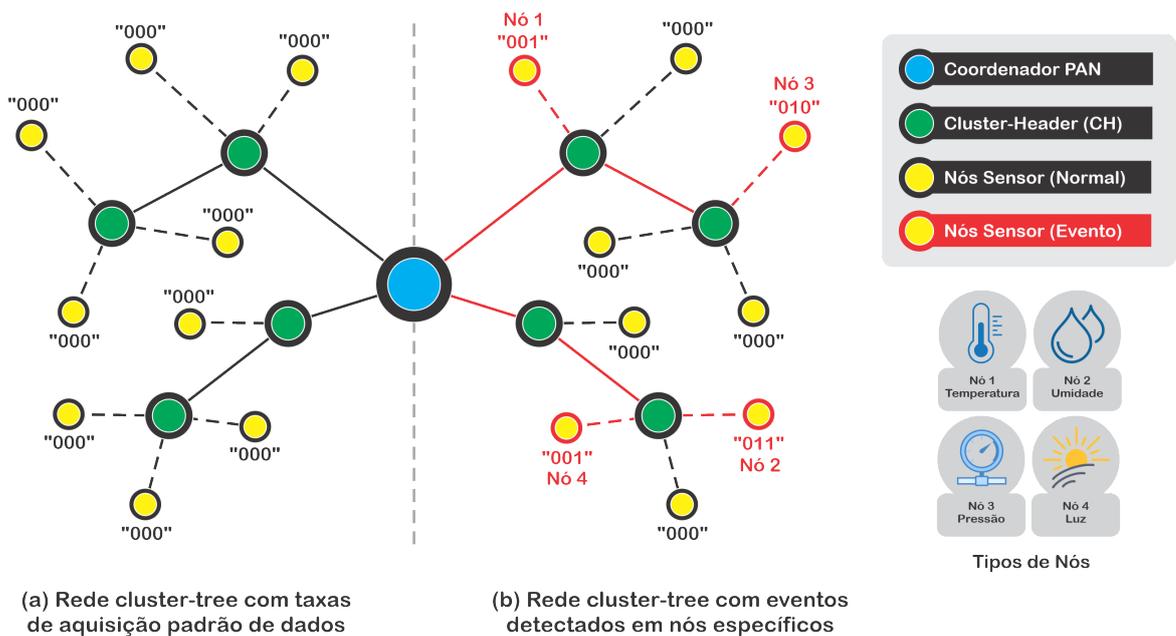


Figura 13 – Rede *cluster-tree* de monitoramento de variáveis físicas.

do quadro do padrão IEEE 802.15.4. A Figura 14 ilustra o formato do cabeçalho MAC 802.15.4 e quadro reservado para variabilidade dos bits.

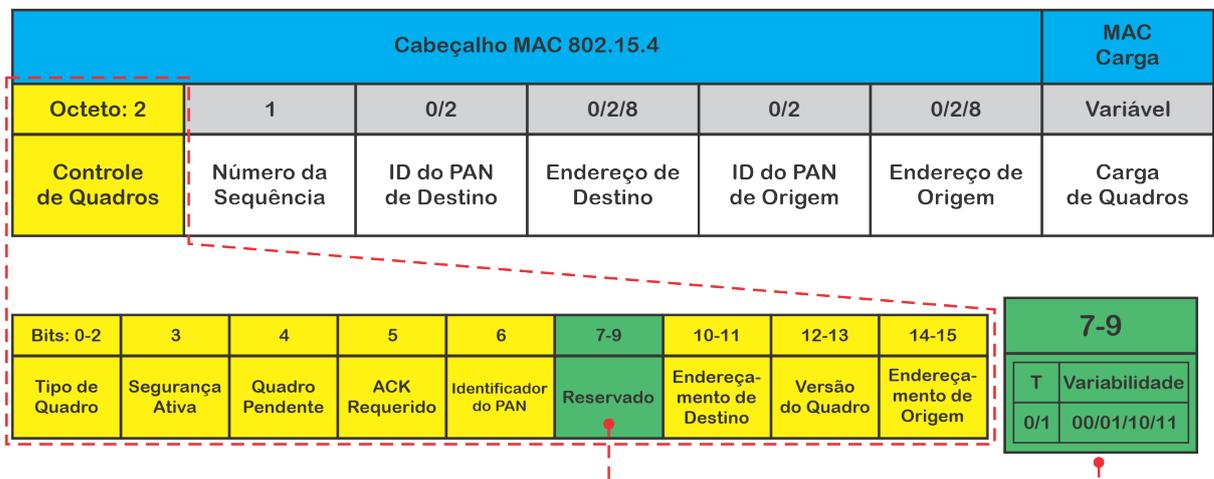


Figura 14 – Formato do cabeçalho do quadro de dados.

À vista disso, o método de detecção de um evento utiliza, portanto, os *bits* do campo reservado para que um determinado nó sensor possa identificar um grau de multiplicidade da periodicidade de geração de dados. Na medida que ocorre aumento de fluxo pelo nós sensores, o mecanismo detecta que houve um evento de carga e o processo de filtragem é iniciado.

É importante destacar que ambos os algoritmos utilizados nesse trabalho são implementados localmente nos nós sensores.

4.6.2 Processo de Filtragem dos Dados

Quanto ao processo de filtragem dos dados, optou-se por uma metodologia de compressão híbrida, utilizando os algoritmos *Swinging Door Trending (SDT)* e Codificação Delta. O objetivo do SDT é reduzir a quantidade de dados enviados ao CHs através da filtragem de dados com valores aproximados e irrelevantes. Já a codificação Delta é utilizada como forma de obter uma maior eficácia do mecanismo SDT, a fim de reduzir a quantidade de pacotes transmitidos para a rede. As próximas subsecções detalham os mecanismos utilizados.

4.6.2.1 1ª Etapa: Processo de Filtragem SDT

A Figura 15 apresenta o fluxograma que corresponde a primeira ação a ser feita pelo módulo de compressão após a detecção do evento crítico. Um processo de filtragem é executado no próprio nó utilizando o *Swinging Door Trending (SDT)*. No primeiro momento, a proposta é reduzir a quantidade de dados enviados aos *cluster-heads* fazendo uma filtragem pelo SDT na medida que transcorre o evento. Durante o evento, o aumento da periodicidade da geração de dados pelos nós pode crescer rapidamente, tendo potencial para iniciar um processo de congestionamento na rede. Se o evento não for detectado então o estado permanece no módulo 4.

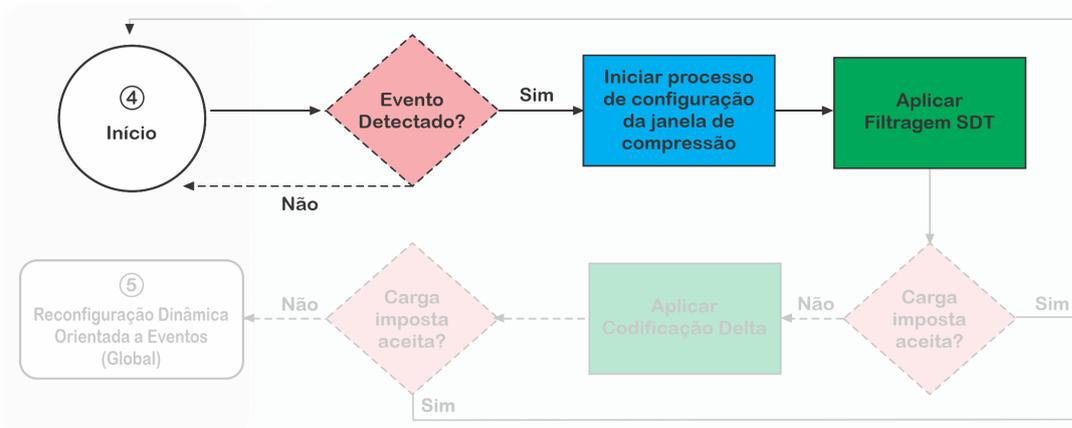


Figura 15 – Iniciando primeira etapa de compressão.

O mecanismo proposto define uma Janela de Compressão (JC) 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 JC, respeitando o tempo mínimo (t_{min}) e máximo (t_{max}), ele será previamente descartado. Os dados que ficarem fora da JC poderão ser selecionados para serem enviados, desde que estejam delimitado pelos tempos t_{min} e t_{max} . O objetivo do t_{min} é a tentativa de filtrar possíveis *outliers*, enquanto que t_{max} é definido para evitar que nós sensores deixem de

enviar dados com valores redundantes. Após selecionar um novo dado para ser enviado pelo nó sensor, o algoritmo define uma nova JC, 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.

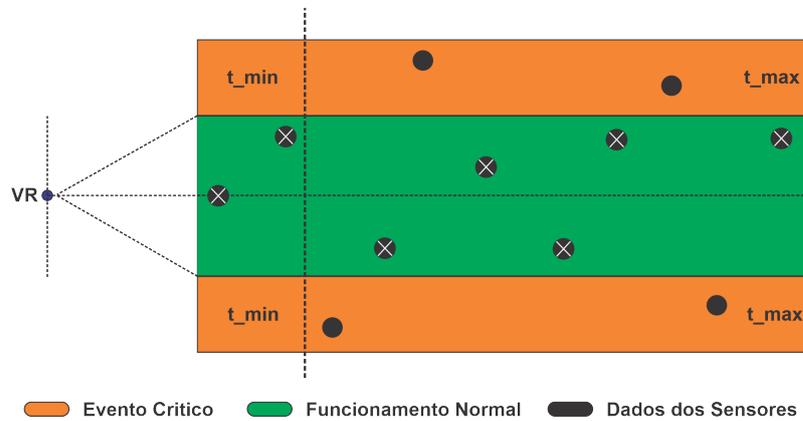


Figura 16 – Janela de Compressão.

A JC é criada sempre que houver um evento crítico, com objetivo de reduzir a quantidade de dados enviados pela rede. O Algoritmo 1 apresenta o pseudo-código do SDT utilizado neste trabalho.

A Figura 17 ilustra o processo de filtragem. Na Figura 17-A, temos o ambiente sem o módulo de compressão, enquanto a 17-B apresenta o módulo de compressão.

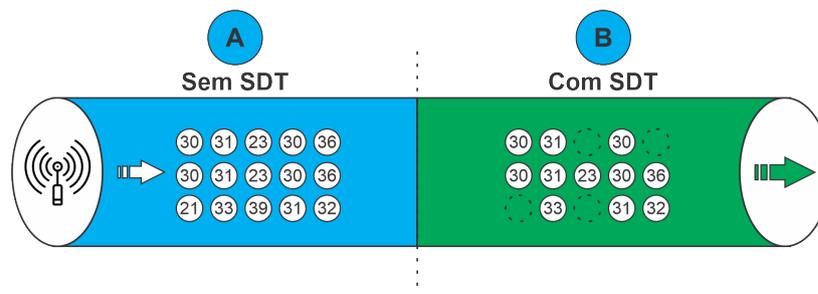


Figura 17 – Representação ilustrativa do SDT.

4.6.2.2 2ª Etapa: Processo de Compressão Delta

O objetivo da segunda etapa de filtragem é a possibilidade de melhorar o mecanismo SDT, através da aplicação de uma segunda camada de compressão. Como pode-se observar na Figura 18, o mecanismo proposto verifica se a filtragem SDT foi suficiente para normalizar o fluxo de dados do nó sensor. Em caso positivo, ele retorna para a etapa 4; caso contrário, a compressão Delta poderá ser aplicado.

Nessa segunda etapa de compressão, o mecanismo implementa a Taxa de Compressão (TC), que define a quantidade de dados de monitoramento que podem ser transmitidos dentro do *payload* de um único pacote da aplicação. Como exemplo, se TC for configurado

Algoritmo 1: Algoritmo SDT

```

1 SDT Initialization
2 referenciaSDT ← dadoSensor;
3 limiteInferior ← referenciaSDT - threshold;
4 limiteSuperior ← referenciaSDT + threshold;
5 tempoMinimo ← t_min;
6 tempoMaximo ← t_max;

7 function SDT(dadoSensor)
8   if dadoSensor > limiteSuperior ou dadoSensor < limiteInferior then
9     if (tempoMinimo == ATIVO) then
10      | dadoSensor deve ser descartado;
11      | return FALSE;
12     else
13      | dadoSensor deve ser enviado;
14      | referenciaSDT ← dadoSensor;
15      | tempoMinimo ← novoValor;
16      | return FALSE;
17      | if (tempoMaximo == ATIVO) then
18      | | cancelaTempoMaximo;
19      | | tempoMaximo ← novoValor;
20      | end
21      | return TRUE;
22     end
23   else
24     if (tempoMinimo == ATIVO) then
25      | dadoSensor deve ser descartado;
26      | return FALSE;
27     else
28      | dadoSensor deve ser enviado;
29      | referenciaSDT ← dadoSensor;
30      | tempoMinimo ← novoValor;
31      | tempoMaximo ← novoValor;
32      | return TRUE;
33     end
34   end
    
```

para 2, ao invés de serem enviados dois pacotes de dados filtrados pelo mecanismo SDT, será enviado apenas um único pacote contendo as duas informações. Podemos visualizar na Figura 19 o funcionamento do mecanismo utilizando apenas o SDT e a proposta híbrida SDT/Delta.

Após findar o evento crítico, havendo assim a normalização do fluxo de dados, os dados dos nós sensores passam a ser enviados para os CHs sem o mecanismo de filtragem e o estado retorna ao módulo 4, no qual temos o funcionamento normal da rede.

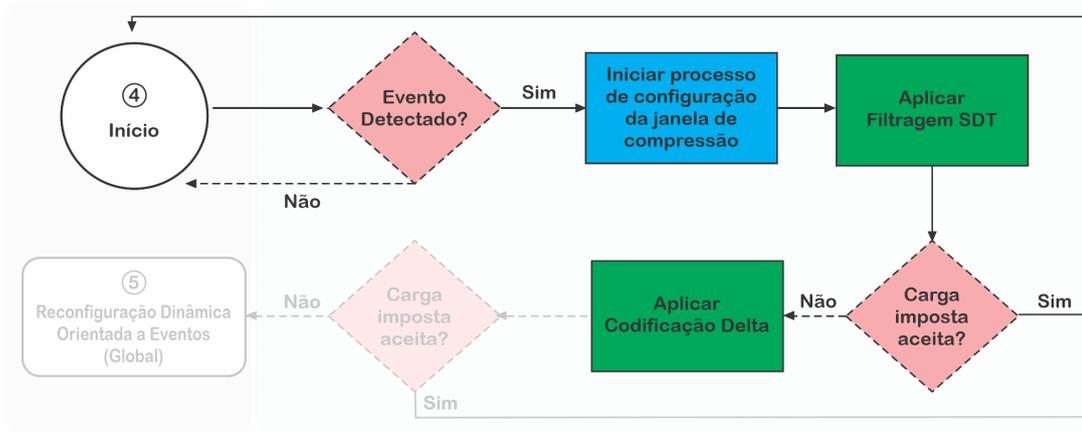


Figura 18 – Processo de compressão Delta.

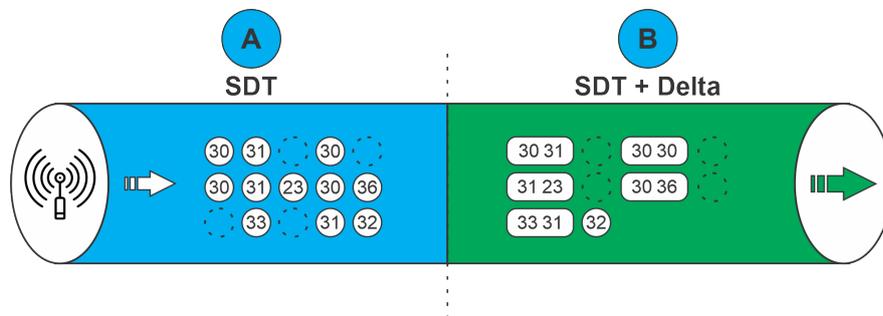


Figura 19 – Representação SDT + Delta.

O processo proposto é repetido ciclicamente, a cada ocorrência de eventos críticos, a fim de obter uma menor taxa de amostragem dos dados, sem comprometer sua relevância (exclusão de dados considerados críticos para aplicação), a fim de evitar um possível processo de reconfiguração global através do coordenador PAN. Entretanto, se mesmo aplicando o mecanismo proposto, a carga imposta pelos nós sensores na ocorrência de um evento crítico superar a capacidade das estruturas de comunicação (LEÃO et al., 2017), o processo de reconfiguração global é disparado (LINO et al., 2019). A Figura 20 ilustra o fluxo do mecanismo de filtragem proposto.

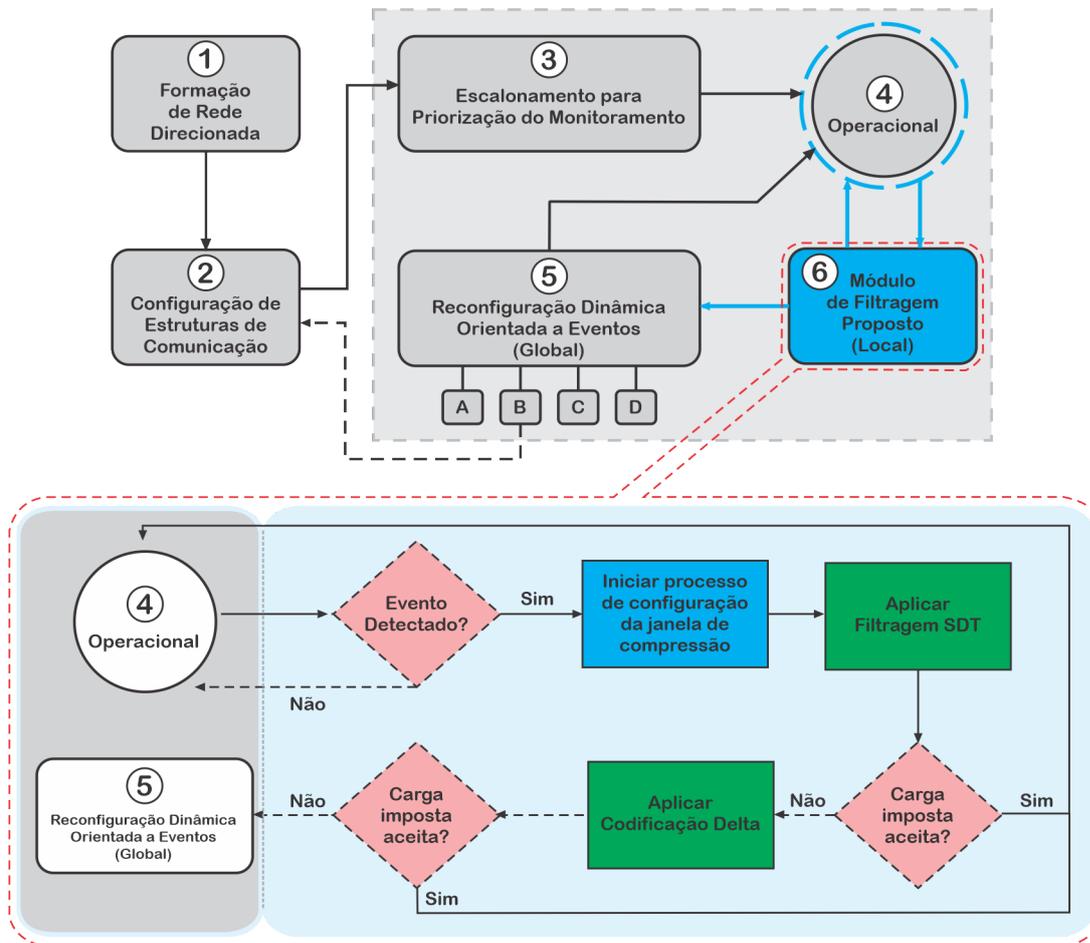


Figura 20 – Funcionamento do mecanismo de compressão local proposto.

5 Resultados e Discussão

Este capítulo apresenta um conjunto de avaliações por simulação, a fim de avaliar e validar o mecanismo de filtragem proposto. Para isso, vários cenários são definidos e métricas de avaliação são utilizadas para avaliar o mecanismo proposto, comparando os resultados com propostas que não utilizam os mecanismos. É importante destacar que parte dos resultados apresentados foram submetidos e aceitos para apresentação no *XII Brazilian Symposium on Computing Systems Engineering* (SBESC 2022).

Este capítulo está organizado da seguinte forma: Seção 5.1 apresenta a definição dos cenários e dos eventos críticos utilizados neste estudo de simulação. A Seção 5.2 apresenta as métricas de simulação utilizadas para avaliar o mecanismo proposto. Por sua vez, a Seção 5.3 apresenta os resultados do mecanismo de filtragem proposto, quando implementado apenas o mecanismo baseado na compressão SDT. Por fim, a Seção 5.4 apresenta os resultados do mecanismo de filtragem proposto baseado nos algoritmos SDT e compressão Delta.

5.1 Cenários de Simulação

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 simulados, o coordenador PAN está localizado na posição central do cenário (100m x 100m), enquanto os nós restantes foram implantados aleatoriamente ao longo de um ambiente de 200m x 200m.

Com relação ao tráfego de monitoramento, foi utilizado uma base de dados real com 10.000 (dez mil) registros da variável física *vazão*. Este banco de dados é utilizado para simular a geração de dados reais pelos nós sensores. Assim, cada nó sensor envia 10.000 dados em direção ao nó *sink* (o coordenador PAN), com uma periodicidade de 1 pacote a cada 20 segundos, seguindo as regras de comunicação hierárquica de redes *cluster-tree*. 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 definido uma área do ambiente onde ocorrem os eventos críticos. Assim, apenas nós sensores localizados nesta área, delimitada pelo eixo x (45m a 145m) e pelo eixo y (0m a 45m), detectam um determinado evento de carga. Esta área corresponde a, aproximadamente, 10% do ambiente total.

Além disso, este trabalho definiu empiricamente que a faixa de valores de vazão

considerados de operação normal varia de 600 a 650. Dessa forma, quaisquer valores 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 o algoritmo de filtragem foi de 1% o valor de referência. A Figura 21 ilustra o banco de dados reais utilizado e quando existe ou não a ocorrência de um evento crítico (zonas em vermelho e verde, respectivamente).

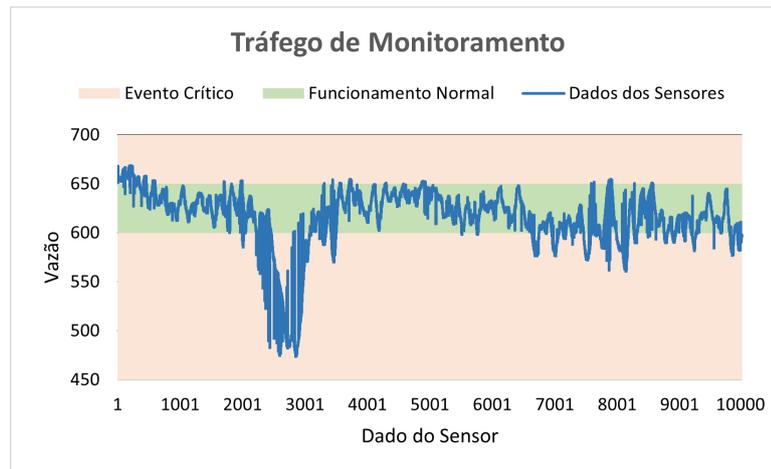


Figura 21 – Representação dos dados dos sensores e zonas de evento crítico (zona vermelha) e de funcionamento normal (zona verde).

5.2 Métricas de Avaliação

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;
- 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 plotadas em gráficos. A tabela 2 apresenta os principais parâmetros de configuração utilizados neste estudo de simulação.

Tabela 2 – Configuração dos principais parâmetros de simulação.

| Descrição | Valor |
|-------------------------------------|-----------------------------|
| Área do ambiente | 200m x 200m |
| Número de nós sensores (exceto PAN) | 100 / 150 / 200 |
| Faixa de valores de dados normal | 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 | 1% o valor de referência |
| Modelo do rádio | Chipcon CC2420 |

5.3 Resultados e Discussão do Mecanismo de Filtragem Baseado Apenas na Compressão SDT

A Figura 22 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.

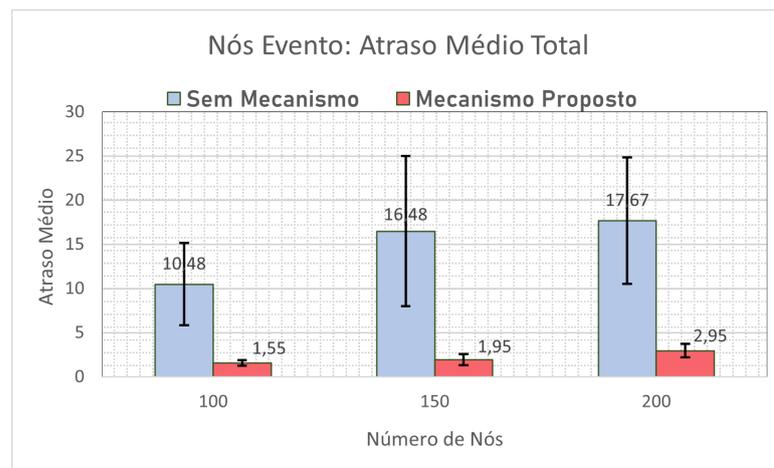


Figura 22 – Atraso médio de comunicação fim-a-fim total dos nós pertencentes à zona de eventos críticos.

Como pode ser observado, é possível verificar o impacto direto que a implementação do mecanismo de filtragem SDT localmente 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 quantidade de dados a serem trafegados pela rede.

Com o objetivo de fazer uma avaliação mais refinada, as Figuras 23 e 24 ilustram os atrasos médios fim-a-fim dos nós pertencentes à zona de eventos críticos durante a ocorrência dos eventos críticos (quando os valores gerados são maiores que 650 ou menores

que 600) e fora dos eventos críticos, respectivamente.

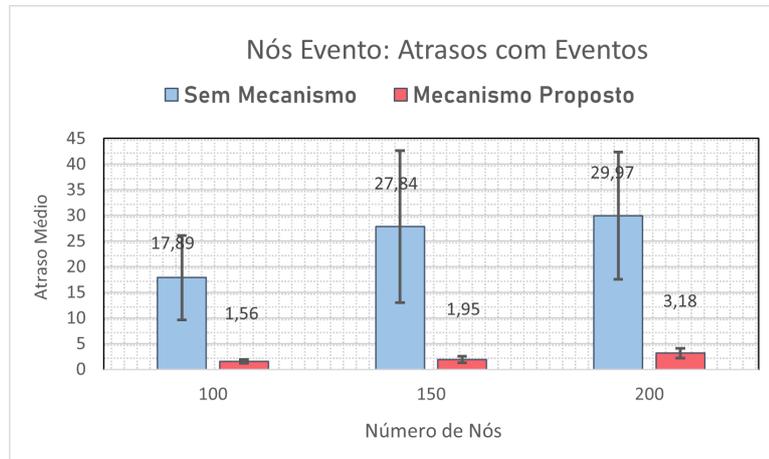


Figura 23 – Atraso médio de comunicação fim-a-fim dos nós pertencentes à zona de eventos críticos, durante a ocorrência do eventos.

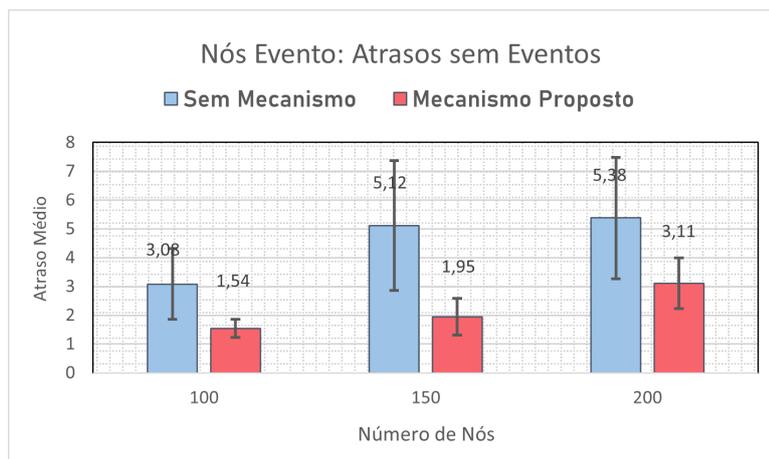


Figura 24 – Atraso médio de comunicação fim-a-fim dos nós pertencentes à zona de eventos críticos, fora da ocorrência dos eventos.

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* contribui também para que os pacotes gerados fora do evento crítico tenham seus atrasos médios aumentados consideravelmente. Esse é o típico cenário de congestionamentos em RSSFs *cluster-tree*.

Por sua vez, a Figura 25 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.

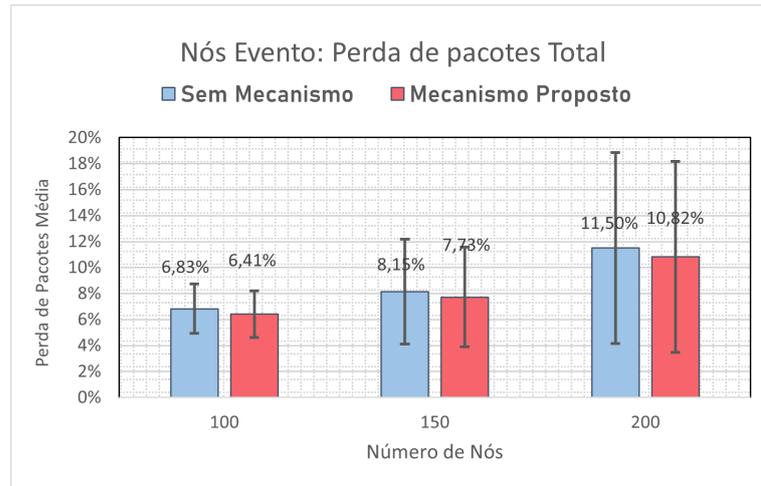


Figura 25 – Taxa média de perda de pacotes total dos nós pertencentes à zona de eventos críticos.

Como pode ser observado, os resultados mostram que os valores médios são menores (em torno de 1,8%) 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, o que não é o caso dos cenários de simulação.

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* são definidos de acordo com o algoritmo *Load-SDA* (LEÃO et al., 2017), que considera a carga total imposta por todos os nós filhos e descendentes. Assim, mesmo com o aumento da geração de dados pelos nós, 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 26 e 27 ilustram, 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 podemos observar, durante a ocorrência dos eventos críticos, com o respectivo aumento da periodicidade dos dados, a tendência é ter um aumento da perda de pacotes (em torno de 5,33%), quando comparados com o período fora do evento crítico.

Ademais, também foi realizado uma avaliação dos atrasos médios de comunicação (Figura 28(a)) e das taxas médias de perda de pacotes (Figura 28(b)) dos nós que estão localizados fora da zona de eventos críticos. Os resultados mostraram que, tanto no cenário com o uso e sem o uso do mecanismo proposto, as médias das métricas de avaliação

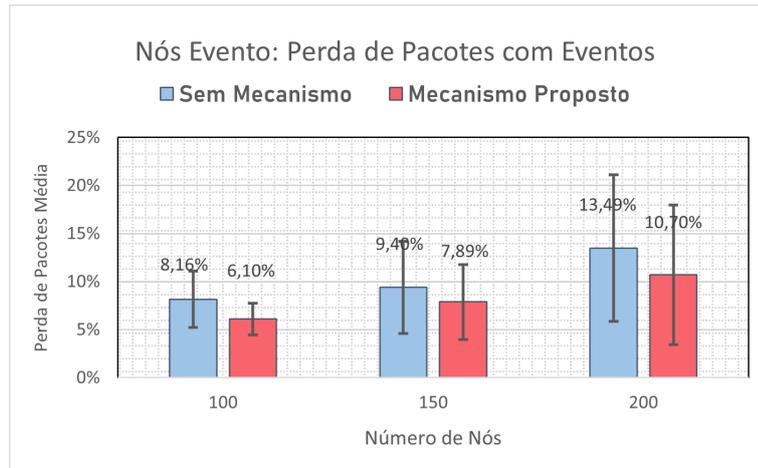


Figura 26 – Taxa média de perda de pacotes dos nós pertencentes à zona de eventos críticos, durante a ocorrência do eventos.

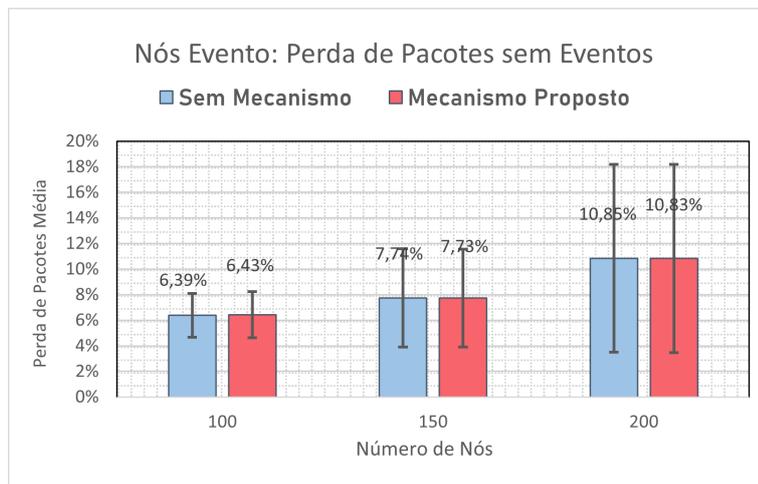


Figura 27 – Taxa média de perda de pacotes dos nós pertencentes à zona de eventos críticos, fora da ocorrência dos eventos.

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 avaliado 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 calculado 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* (LEÃO et al., 2017). 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

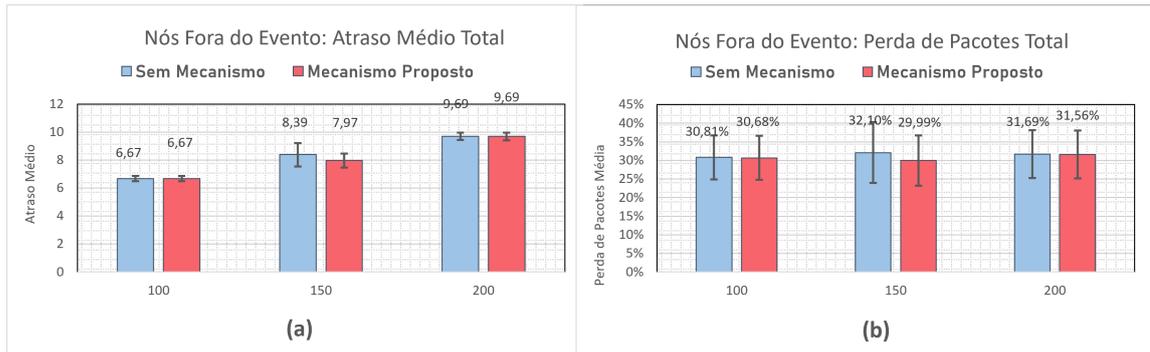


Figura 28 – Nós localizados fora da região de eventos críticos: (a) Atraso médio total e (b) Taxa média de perda de pacotes.

meccanismo precisaria ativar o procedimento de reconfiguração de rede em todos os cenários simulados.

5.4 Resultados e Discussões do Mecanismo de Compressão Utilizando SDT e Delta

Como tentativa de melhorar o desempenho do mecanismo de filtragem baseado no SDT, utilizamos uma segunda etapa de compressão de dados baseados na compressão Delta. A ideia principal é que uma determinada quantidade de dados filtrados pelo SDT fossem enviados no mesmo pacote da aplicação. Para este estudo de simulação, foi utilizado a taxa de compressão (TC) igual a 2, ou seja, dois dados SDT são enviados juntos no mesmo pacote.

A Figura 29 apresenta os atrasos médios e as taxas de perda de pacotes dos nós sensores localizados na zona de eventos críticos, a partir da implementação do mecanismo SDT/Delta.



Figura 29 – Nós localizados na região de eventos críticos, durante a ocorrência do evento: (a) Atraso médio de comunicação e (b) Taxa média de perda de pacotes.

Como pode ser observado, considerando a variabilidade dos resultados e uma TC de 2, a implementação do mecanismo baseado na compressão Delta não trouxe benefícios

para a diminuição dos atrasos e perda de pacotes dos nós localizados na zona de eventos críticos, quando aliados ao SDT.

De fato, além da perda de pacotes ser um problema inerente às RSSFs, o algoritmo MAC CSMA-CA do IEEE 802.15.4 implementa um número máximo de tentativas de retransmissão (*macMaxFrameRetries*) de um pacote não entregue. Após atingir essa quantidade máxima, o algoritmo de acesso ao meio sumariamente descarta o pacote. O cenário se torna ainda mais crítico, considerando que um pacote gerado por nós sensores de maior profundidade da rede precisa atravessar vários nós coordenadores, até atingir o nó *sink* (coordenador PAN). Assim, as redes *cluster-tree* considerada neste trabalho não implementa um mecanismo de entrega confiável de dados fim-a-fim, ou seja, dos nós sensores até o coordenador PAN.

Dentro desse contexto, como o mecanismo Delta implementa uma taxa de compressão de dados em um único pacote de aplicação, um determinado pacote perdido na rede, devido às colisões e atingir o número máximo de retransmissões, contabiliza uma quantidade de perdas multiplicadas pela TC, o que contribui diretamente com os resultados alcançados.

Além disso, como comentado na Seção 5.3, a taxa de perda de pacotes é fortemente impactada pelas colisões e descarte de pacotes nos *buffers* dos nós coordenadores. Assim, o mecanismo Delta acaba não influenciando fortemente nessas duas situações, mantendo assim a taxa de perda constante. A mais, os atrasos são impactados fortemente pelo enfileiramento de pacotes nos *buffers*. Como a taxa de perda de pacotes não foi impactada, presume-se aqui que a quantidade de pacotes enfileirados nos *buffers* são semelhantes nos dois cenários avaliados, o que gera atrasos também semelhantes.

Por sua vez, a Figura 30 apresenta os atrasos médios e as taxas de perda de pacotes dos nós sensores localizados fora da zona de eventos críticos.

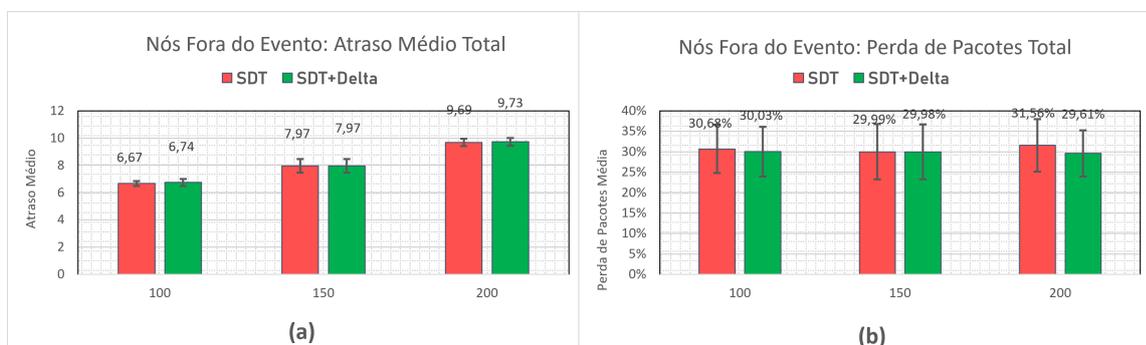


Figura 30 – Nós localizados fora da região de eventos críticos: (a) Atraso médio total e (b) Taxa média de perda de pacotes.

De forma semelhante ao mecanismo SDT, os resultados mostram que a implementação Delta também não impacta nos atrasos e perda de pacotes dos nós sensores localizados fora da zona de evento crítico. Isso se deve ao fato de que o fluxo de comunicação desses

nós se mantém constante do início ao fim da simulação e de que não há alteração na quantidade de nós sensores em contenção ao meio de comunicação nos *clusters*.

5.5 Considerações Finais

Os resultados de simulação apresentados mostram a relevância de se implementar mecanismos eficientes de filtragem local de dados em redes *cluster-tree*, com o objetivo de diluir congestionamentos de rede e evitar/postergar possíveis reconfigurações das estruturas de comunicação de rede.

O mecanismo baseado no SDT se mostrou eficiente na melhoria de métricas de redes e em evitar possíveis reconfigurações de rede na existência de eventos de carga. Por sua vez, o mecanismo baseado em SDT/Delta não mostrou melhorias significativas, quando comparado ao mecanismo SDT. Assim, é preciso avaliar a necessidade da implementação de um mecanismo de comunicação confiável de dados fim-a-fim, aliado ao mecanismo de compressão de dados Delta. Além disso, é preciso avaliar se o mecanismo de compressão Delta pode trazer vantagens significativas quando implementados nos coordenadores de *clusters*, ao invés dos nós sensores finais.

6 Conclusões e Trabalhos Futuros

6.1 Conclusão

Neste trabalho foi desenvolvido um Mecanismo de Controle de Congestionamento Local Disparado por Eventos Através de Filtragem de Dados para RSSFs *cluster-tree* baseadas no padrão IEEE 802.15.4.

Considerando uma infraestrutura de rede existente, a ideia principal deste trabalho é desenvolver um módulo de filtragem local de dados, através da implementação de um mecanismo de compressão nos nós sensores, baseado nos algoritmos *Swinging Door Trending* e codificação Delta, que possa evitar a ocorrência de congestionamentos de rede e, conseqüentemente, o aumento dos atrasos de comunicação fim-a-fim, perdas de pacotes e a necessidade de procedimentos de reconfiguração das estruturas de comunicação. Dessa forma, o mecanismo proposto foi comparado com diferentes abordagens e, considerando as principais métricas de redes, foi avaliado em diferentes cenários de simulação.

Por meio de avaliações de simulação com o uso do conjunto de modelos de simulação CT-SIM, o mecanismo proposto mostrou relevância de se implementar mecanismo eficientes de filtragem local de dados em redes *cluster-tree*, com o objetivo de diluir congestionamentos de rede e evitar/postergar possíveis reconfigurações das estruturas de comunicação de rede.

Os resultados de simulação mostraram que a implementação de um mecanismo de filtragem de dados, por desconsiderar dados considerados redundantes ou *outliers*, é relevante para a diminuição de dados a serem transmitidos pela rede, reduzindo assim os atrasos de comunicação fim-a-fim e perda de pacotes de fluxos de dados de alta periodicidade, tipicamente gerados em ambientes de eventos críticos. Dessa forma, o mecanismo proposto se mostrou eficiente para reduzir congestionamentos de rede e disparar processos de reconfiguração de parâmetros de comunicação, que tendem a gastar mais energia e interferir sobre os fluxos de dados dos nós sensores.

6.2 Trabalhos Futuros

Como trabalhos futuros, pretende-se otimizar o mecanismo de filtragem de dados, através da implementação de um mecanismo de comunicação confiável fim-a-fim, dos nós sensores até o coordenador PAN, aliados ao mecanismo de compressão Delta. Além disso, pretende-se também avaliar a eficiência do mecanismo Delta na compressão de dados localizados nos *buffers* dos nós coordenadores.

Ademais, pretende-se utilizar algoritmos de Aprendizado de Máquina (*Deep Learning*) ou Lógica *Fuzzy*, a fim de implementar um mecanismo dinâmico de filtragem de dados de acordo com as características de operação da rede, tais como quantidade de pacotes nos buffers dos nós coordenadores e taxas de colisão inter e intra *clusters*.

Por fim, os mecanismos propostos serão avaliados através da implementação real, utilizando dispositivos embarcados.

Referências

- ABDULZAHRA, S. A.; AL-QURABAT, A. K. M.; IDREES, A. K. Compression-based data reduction technique for iot sensor networks. *Baghdad Science Journal*, v. 18, n. 1, p. 0184–0184, 2021. Citado 2 vezes nas páginas 24 e 30.
- ADEBAYO, A. O.; CHAUBEY, M. S.; NUMBU, L. P. 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)*, v. 5, n. 2, 2019. Citado na página 1.
- AKYILDIZ, I. F. et al. Wireless sensor networks: a survey. *Computer networks*, Elsevier, v. 38, n. 4, p. 393–422, 2002. Citado na página 1.
- AL-KADHIM, H. M.; AL-RAWESHIDY, H. S. Energy efficient data compression in cloud based iot. *IEEE Sensors Journal*, IEEE, v. 21, n. 10, p. 12212–12219, 2021. Citado 2 vezes nas páginas 29 e 30.
- AL-QURABAT, A. K. M.; JAOUDE, C. A.; IDREES, A. K. Two tier data reduction technique for reducing data transmission in iot sensors. In: IEEE. *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. [S.l.], 2019. p. 168–173. Citado 2 vezes nas páginas 28 e 30.
- AL-QURABAT, A. K. M.; MOHAMMED, Z. A.; HUSSEIN, Z. J. Data traffic management based on compression and mdl techniques for smart agriculture in iot. *Wireless Personal Communications*, Springer, p. 1–32, 2021. Citado 2 vezes nas páginas 28 e 30.
- ANDRADE, A. T. C. et al. Uma abordagem orientada a dados para reconfiguração de topologia de rede cluster-tree. 2019. Citado 2 vezes nas páginas 9 e 16.
- AZAR, J. et al. An energy efficient iot data compression approach for edge machine learning. *Future Generation Computer Systems*, Elsevier, v. 96, p. 168–175, 2019. Citado 2 vezes nas páginas 26 e 30.
- AZAR, J. et al. Robust iot time series classification with data compression and deep learning. *Neurocomputing*, Elsevier, v. 398, p. 222–234, 2020. Citado na página 27.
- AZIZ, A. et al. Effective algorithm for optimizing compressive sensing in iot and periodic monitoring applications. *Journal of Network and Computer Applications*, Elsevier, v. 126, p. 12–28, 2019. Citado na página 24.
- BAHREPOUR, M. et al. Use of event detection approaches for outlier detection in wireless sensor networks. In: IEEE. *2009 International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. [S.l.], 2009. p. 439–444. Citado na página 15.
- BAKER, F.; FAIRHURST, G. Ietf recommendations regarding active queue management: Internet best current practice (bcp 197). 2015. Citado na página 16.

- BARATH, S. R. et al. Strategies for congestion control in wireless sensor networks: A survey. In: IEEE. *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. [S.l.], 2019. p. 629–633. Citado na página 3.
- BOSE, T. et al. Signal characteristics on sensor data compression in iot-an investigation. In: IEEE. *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. [S.l.], 2016. p. 1–6. Citado na página 24.
- BRISTOL, E. Swinging door trending: Adaptive trend recording? In: *ISA National Conf. Proc., 1990*. [S.l.: s.n.], 1990. p. 749–754. Citado na página 19.
- BRISTOL, E. H. *Data compression for display and storage*. [S.l.]: Google Patents, 1987. US Patent 4,669,097. Citado na página 19.
- CHATTERJEE, A.; SHAH, R. J.; HASAN, K. S. Efficient data compression for iot devices using huffman coding based techniques. In: IEEE. *2018 IEEE International Conference on Big Data (Big Data)*. [S.l.], 2018. p. 5137–5141. Citado 2 vezes nas páginas 25 e 30.
- CORREA, J. D. A. et al. Swinging door trending compression algorithm for iot environments. In: SBC. *Anais Estendidos do IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais*. [S.l.], 2019. p. 143–148. Citado 2 vezes nas páginas 23 e 30.
- CUNHA, A. et al. Open-zb: an open-source implementation of the ieee 802.15. 4/zigbee protocol stack on tinys. In: IEEE. *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*. [S.l.], 2007. p. 1–12. Citado na página 11.
- DEEPU, C. J.; HENG, C.-H.; LIAN, Y. A hybrid data compression scheme for power reduction in wireless sensors for iot. *IEEE transactions on biomedical circuits and systems*, IEEE, v. 11, n. 2, p. 245–254, 2016. Citado 2 vezes nas páginas 27 e 30.
- FELSKE, M. S. et al. *GLHOVE: uma abordagem para o monitoramento uniforme em RSSF IEEE 802.15. 4 cluster tree*. Dissertação (Engenharia de Automação e Sistemas) — Universidade Federal de Santa Catarina, 2013. Citado na página 12.
- GHAFFARI, A. Congestion control mechanisms in wireless sensor networks: A survey. *Journal of network and computer applications*, Elsevier, v. 52, p. 101–115, 2015. Citado na página 13.
- GLUHAK, A. et al. A survey on facilities for experimental internet of things research. *IEEE Communications Magazine*, IEEE, v. 49, n. 11, p. 58–67, 2011. Citado na página 2.
- HAMDAN, S.; AWAIAN, A.; ALMAJALI, S. Compression techniques used in iot: A comparative study. In: IEEE. *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*. [S.l.], 2019. p. 1–5. Citado na página 18.
- HANUMANTHAI AH, A. et al. Comparison of lossless data compression techniques in low-cost low-power (lclp) iot systems. In: IEEE. *2019 9th International Symposium on Embedded Computing and System Design (ISED)*. [S.l.], 2019. p. 1–5. Citado 2 vezes nas páginas 28 e 30.
- HARA, S. et al. Propagation characteristics of ieee 802.15. 4 radio signal and their application for location estimation. In: IEEE. *2005 IEEE 61st Vehicular Technology Conference*. [S.l.], 2005. v. 1, p. 97–101. Citado na página 8.

IEEE. Ieee 802.15. 4-2015-ieee standard for low-rate wireless networks. *IEEE standards association*, 2015. Citado na página 7.

IEEE. *IEEE Standard for Low-Rate Wireless Networks. IEEE Std 802.15. 4-2015 (Revision of IEEE Std 802.15. 4-2011)*. [S.l.]: IEEE Piscataway, NJ, USA, 2016. Citado na página 2.

JAN, M. A. et al. A comprehensive analysis of congestion control protocols in wireless sensor networks. *Mobile networks and applications*, Springer, v. 23, n. 3, p. 456–468, 2018. Citado na página 17.

JAYASANKAR, U.; THIRUMAL, V.; PONNURANGAM, D. A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications. *Journal of King Saud University-Computer and Information Sciences*, Elsevier, v. 33, n. 2, p. 119–140, 2021. Citado na página 18.

JINDAL, R.; KUMAR, N.; PATIDAR, S. Iot stream data compression using ldpc coding. In: *Evolution in Computational Intelligence*. [S.l.]: Springer, 2021. p. 455–466. Citado 2 vezes nas páginas 26 e 30.

KHALIL, R. A. et al. Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. *IEEE Internet of Things Journal*, IEEE, 2021. Citado na página 1.

KIRAZA, A.; ÇAKIROĞLUB, M. A survey of congestion control protocols providing energy conservation in wireless sensor networks. *Turkish Journal of Engineering*, v. 1, p. 12–22, 2014. Citado na página 13.

KOUBAA, A.; CUNHA, A.; ALVES, M. A time division beacon scheduling mechanism for ieee 802.15. 4/zigbee cluster-tree wireless sensor networks. In: IEEE. *19th Euromicro Conference on Real-Time Systems (ECRTS'07)*. [S.l.], 2007. p. 125–135. Citado na página 12.

KUMAR, T.; MANE, P. Zigbee topology: A survey. In: IEEE. *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. [S.l.], 2016. p. 164–166. Citado na página 7.

KUZMANOVIC, A. The power of explicit congestion notification. In: *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*. [S.l.: s.n.], 2005. p. 61–72. Citado na página 17.

LE, T. L.; VO, M.-H. Lossless data compression algorithm to save energy in wireless sensor network. In: IEEE. *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*. [S.l.], 2018. p. 597–600. Citado 2 vezes nas páginas 25 e 30.

LEÃO, E. et al. Superframe duration allocation schemes to improve the throughput of cluster-tree wireless sensor networks. *Sensors*, MDPI, v. 17, n. 2, p. 249, 2017. Citado 7 vezes nas páginas 3, 9, 35, 38, 43, 49 e 50.

LEÃO, E. M. *Improving Source-to-Destination Communication Schemes in Wide-Scale Cluster-Tree Wireless Sensor Networks*. Tese (Doutorado) — Universidade do Porto (Portugal), 2016. Citado 2 vezes nas páginas 3 e 4.

- LINO, M. et al. Mecanismo eficiente para disseminação do tráfego de controle em redes de sensores sem fio cluster-tree baseadas no iee 802.15. 4. In: SBC. *Anais Estendidos do IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais*. [S.l.], 2019. p. 1–8. Citado 7 vezes nas páginas 2, 3, 4, 31, 33, 38 e 43.
- MATIN, M. A. *Wireless sensor networks: Technology and protocols*. [S.l.]: BoD–Books on Demand, 2012. Citado na página 1.
- MOHAMMADI, M. et al. Deep learning for iot big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 4, p. 2923–2960, 2018. Citado 2 vezes nas páginas 1 e 2.
- MUNDADA, M. R.; DESAI, P. B. et al. A survey of congestion in wireless sensor networks. In: IEEE. *2016 International Conference on Advances in Human Machine Interaction (HMI)*. [S.l.], 2016. p. 1–5. Citado na página 17.
- NETO, E. J. et al. 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*. [S.l.: s.n.], 2014. p. 20–24. Citado 3 vezes nas páginas 19, 23 e 30.
- NOURA, H. N.; SALMAN, O.; COUTURIER, R. A deep learning scheme for efficient multimedia iot data compression. *arXiv preprint arXiv:2105.09280*, 2021. Citado 2 vezes nas páginas 26 e 30.
- PANWAR, A.; KUMAR, S. A. Localization schemes in wireless sensor networks. In: IEEE. *2012 Second International Conference on Advanced Computing & Communication Technologies*. [S.l.], 2012. p. 443–449. Citado na página 9.
- PARK, J.; PARK, H.; CHOI, Y.-J. Data compression and prediction using machine learning for industrial iot. In: IEEE. *2018 International Conference on Information Networking (ICOIN)*. [S.l.], 2018. p. 818–820. Citado 2 vezes nas páginas 25 e 30.
- PEI, X. et al. Spatio-temporal event detection: a hierarchy based approach for wireless sensor network. In: IEEE. *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*. [S.l.], 2014. p. 372–379. Citado na página 15.
- QUEIROZ, D. V. d. et al. *Simulação realista de redes de sensores sem fio industriais*. Dissertação (Mestrado) — Universidade Federal da Paraíba, 2016. Citado na página 2.
- RAMAKRISHNAN, K. et al. The addition of explicit congestion notification (ecn) to ip. rfc 3168, September, 2001. Citado na página 17.
- RAMALINGAM, T.; TWETEN, D. Assessing the potential of iot in systems engineering discipline. In: WILEY ONLINE LIBRARY. *INCOSE International Symposium*. [S.l.], 2019. v. 29, p. 291–301. Citado na página 2.
- RATHNAYAKA, A. D.; POTDAR, V. M. Wireless sensor network transport protocol: A critical review. *Journal of Network and Computer Applications*, Elsevier, v. 36, n. 1, p. 134–146, 2013. Citado na página 13.
- REIS, H. P.; FERNANDEZ, F. J. R. Análise das perdas de pacotes alterando as variáveis de distancia obstáculos e interferência electromagnetic utilizando iee 802.15. 4. *VII Encontro Latino Americano de Pós-Graduação – UVP*, 2007. Citado na página 2.

- RGHIOUI, A.; OUMNAD, A. Internet of things: Visions, technologies, and areas of application. *technology*, v. 6, n. 7, 2017. Citado na página 2.
- RODRIGUES, T. C. et al. Using mda for building wireless sensor network applications. In: IEEE. *2010 Fourth Brazilian Symposium on Software Components, Architectures and Reuse*. [S.l.], 2010. p. 110–119. Citado 2 vezes nas páginas 1 e 2.
- ROMAN, R.; ALCARAZ, C.; LOPEZ, J. The role of wireless sensor networks in the area of critical information infrastructure protection. *Information Security Technical Report*, Elsevier, v. 12, n. 1, p. 24–31, 2007. Citado na página 1.
- ROY, S. K. et al. Agrisens: Iot-based dynamic irrigation scheduling system for water management of irrigated crops. *IEEE Internet of Things Journal*, IEEE, v. 8, n. 6, p. 5023–5030, 2020. Citado na página 2.
- SAIDANI, A.; JIANWEN, X.; MANSOURI, D. A lossless compression approach based on delta encoding and t-rlc in wsns. *Wireless Communications and Mobile Computing*, Hindawi, v. 2020, 2020. Citado 2 vezes nas páginas 28 e 30.
- SANTOS, E. L. F. dos. A ieee 802.15. 4 como plataforma de comunicação de dados. *Revista Ilha Digital*, v. 4, p. 97–105, 2013. Citado 3 vezes nas páginas 2, 3 e 7.
- SEVERINO, R.; PEREIRA, N.; TOVAR, E. Dynamic cluster scheduling for cluster-tree wsns. *SpringerPlus*, Springer, v. 3, n. 1, p. 1–17, 2014. Citado na página 9.
- SHAH, K.; GORTY, V. L.; PHIRKE, A. *Technology Systems and Management: First International Conference, ICTSM 2011, Mumbai, India, February 25-27, 2011. Selected Papers*. [S.l.]: Springer, 2011. v. 145. Citado na página 16.
- SHANTHAMALLU, U. S. et al. A brief survey of machine learning methods and their sensor and iot applications. In: IEEE. *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*. [S.l.], 2017. p. 1–8. Citado na página 26.
- SHEN, X.; WANG, Z.; SUN, Y. Wireless sensor networks for industrial applications. In: IEEE. *Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788)*. [S.l.], 2004. v. 4, p. 3636–3640. Citado na página 1.
- SHRAVANA, K. P.; VEENA, D. S. V. Review on lossless data compression using x-matchpro algorithm. In: IEEE. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. [S.l.], 2017. p. 1095–1100. Citado na página 17.
- SIGNORETTI, G. et al. An evolving tinyml compression algorithm for iot environments based on data eccentricity. *Sensors*, MDPI, v. 21, n. 12, p. 4153, 2021. Citado 2 vezes nas páginas 26 e 30.
- SOBRAL, J. V. et al. A framework for enhancing the performance of internet of things applications based on rfid and wsns. *Journal of Network and Computer Applications*, Elsevier, v. 107, p. 56–68, 2018. Citado na página 1.
- STANKOVIC, J. A.; WOOD, A. D.; HE, T. Realistic applications for wireless sensor networks. In: *Theoretical aspects of distributed computing in sensor networks*. [S.l.]: Springer, 2011. p. 835–863. Citado na página 1.

- STOJKOSKA, B. R.; NIKOLOVSKI, Z. Data compression for energy efficient iot solutions. In: IEEE. *2017 25th telecommunication forum (TELFOR)*. [S.l.], 2017. p. 1–4. Citado 2 vezes nas páginas 25 e 30.
- VERMESAN, O.; FRIESS, P. et al. *Internet of things-from research and innovation to market deployment*. [S.l.]: River publishers Aalborg, 2014. v. 29. Citado na página 2.
- WU, J.; HU, Y.; WANG, X. Research on event-driven wireless sensor network multi-path routing protocol. In: IEEE. *2010 Sixth International Conference on Natural Computation*. [S.l.], 2010. v. 2, p. 534–538. Citado na página 15.
- YANG, K. *Wireless sensor networks*. [S.l.]: Springer, 2014. Citado na página 1.
- YANG, X. et al. Wireless sensor network congestion control based on standard particle swarm optimization and single neuron pid. *Sensors*, MDPI, v. 18, n. 4, p. 1265, 2018. Citado na página 17.
- YAO, S. et al. Deep learning for the internet of things. *Computer*, IEEE, v. 51, n. 5, p. 32–41, 2018. Citado na página 27.
- YIN, J.; HU, D. H.; YANG, Q. Spatio-temporal event detection using dynamic conditional random fields. In: *Twenty-First International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2009. Citado na página 15.
- YIN, X. et al. A fairness-aware congestion control scheme in wireless sensor networks. *IEEE transactions on vehicular technology*, IEEE, v. 58, n. 9, p. 5225–5234, 2009. Citado na página 13.
- YU, X. et al. Two-level data compression using machine learning in time series database. In: IEEE. *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. [S.l.], 2020. p. 1333–1344. Citado na página 23.
- ZHENG, J.; JAMALIPOUR, A. *Wireless sensor networks: a networking perspective*. [S.l.]: John Wiley & Sons, 2009. Citado na página 14.
- ZIGBEE. *Zigbee specification overview*. [S.l.], 2012. v. 14, n. 10. Citado na página 7.
- ZIGBEE, A. Document 053474r20: Zigbee specification. *Davis, CA: ZigBee Alliance*, 2012. Citado na página 2.