

**Universidade Federal de Santa Catarina
Departamento de Informática e Estatística**

César Huegel Richa

**ST-DBP: Uma abordagem baseada na correlação
espaço-temporal para supressão de mensagens em RSSF**

Florianópolis - SC

2019

César Huegel Richa

**ST-DBP: Uma abordagem baseada na correlação
espaço-temporal para supressão de mensagens em RSSF**

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do Grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Antônio Augusto Medeiros Fröhlich

Florianópolis - SC

2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Richa, César Huegel

ST-DBP: Uma abordagem baseada na correlação espaço temporal para supressão de mensagens em RSSF / César Huegel Richa ; orientador, Antônio Augusto Medeiros Fröhlich, 2019.

74 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós Graduação em Ciência da Computação, Florianópolis, 2019.

Inclui referências.

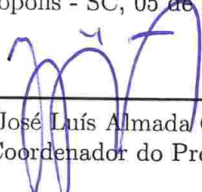
1. Ciência da Computação. 2. Redes de Sensores Sem Fio. 3. Supressão de Mensagens. 4. Correlação Espaço-Temporal. I. Fröhlich, Antônio Augusto Medeiros. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

César Huegel Richa

**ST-DBP: Uma abordagem baseada na correlação
espaço-temporal para supressão de mensagens em RSSF**

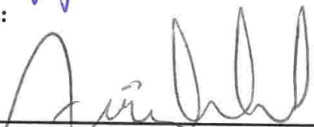
Esta dissertação foi julgada adequada para a obtenção do título de “Mestre em Ciência da Computação”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina.

Florianópolis - SC, 05 de abril de 2019.



Prof. José Luís Almada Güntzel, Dr.
Coordenador do Programa

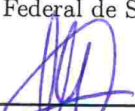
Banca Examinadora:



Prof. Antônio Augusto Medeiros Fröhlich, Dr.
Universidade Federal de Santa Catarina
Orientador

Carla Merkle Westphall

Profa. Carla Merkle Westphall, Dra.
Universidade Federal de Santa Catarina



Prof. Gustavo Medeiros de Araujo, Dr.
Universidade Federal de Santa Catarina



Prof. Ricardo Alexandre Reinaldo de Moraes, Dr.
Universidade Federal de Santa Catarina

Agradecimentos

À minha família que, com muito amor, compreendeu os momentos de ausência e me deu todo o apoio necessário para que eu pudesse concluir mais uma importante etapa em minha trajetória.

Ao Prof. Dr. Antônio Augusto Fröhlich, pela orientação e assistência na execução deste trabalho, pelos conhecimentos técnicos e científicos partilhados, e pela confiança em mim depositada.

Aos membros e colaboradores do LISHA (Laboratório de Integração de Software e Hardware), pela contribuição mútua neste e em outros trabalhos científicos realizados.

À UFSC (Universidade Federal de Santa Catarina) ao PPGCC (Programa de Pós-Graduação em Ciência da Computação) por todo o suporte. Também à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela bolsa de estudos, que viabilizou dedicar-me exclusivamente às atividades acadêmicas, em especial à elaboração desta dissertação.

A todos que, direta ou indiretamente, contribuíram para o êxito deste trabalho, obrigado.

Resumo

Redes de Sensores sem Fio é uma tecnologia emergente que vem ganhando atenção da comunidade acadêmico-científica e tecnológica mundial, principalmente nas últimas décadas. Esta tecnologia apresenta, dentre suas principais características, a disponibilidade limitada de recursos, principalmente energéticos. Assim, um dos principais desafios em se projetar soluções baseadas nesta tecnologia é a maximização da autonomia da rede como um todo. Neste contexto, a aplicação da técnica de supressão de mensagens, baseada em modelos de tendência, como mecanismo para reduzir o número de mensagens no nível de aplicação, promove economia de energia gasta com comunicação e conseqüentemente, maximiza a autonomia da rede de sensores sem fio. Trabalhos recentes propuseram técnicas de supressão de mensagens baseadas em modelos lineares, que apresentaram um bom desempenho de supressão, com baixa complexidade e custo computacional, além de apresentarem um alto grau de simplicidade. Apesar de eficientes, estas técnicas consideram apenas a correlação temporal dos dados, desconsiderando a correlação espacial das grandezas físicas sensoreadas, tipicamente presentes em vários fenômenos monitorados, o que pode reduzir ainda mais o número total de mensagens. Neste sentido, esta dissertação consiste na proposta e avaliação de um mecanismo de supressão de mensagens para redes de sensores sem fio que explora a correlação espaço-temporal das grandezas físicas sensoreadas. O método proposto é avaliado em um ambiente de simulação, envolvendo dados reais de uma rede de sensores sem fio de grande escala, significativamente densa, e com uma frequência de amostragem relativamente alta. Os resultados obtidos confirmam que o método consegue melhorar as taxas de supressão de mensagens e reduzir o consumo de energia, refletido no tempo de vida da rede.

Palavras-chave: Redes de Sensores Sem Fio, Supressão de Mensagens, Correlação Espaço-Temporal.

Abstract

Wireless Sensor Networks is an emerging technology that has been gaining attention from the worldwide academic and scientific community, especially in recent decades. This technology presents, among its main characteristics, the limited resources available, mainly energy. Thus, one of the main challenges in designing solutions based on this technology is the maximization of the autonomy of the network as a whole. In this context, the application of the data suppression technique, based on trend models, as a mechanism to reduce the number of messages at the application level, promotes energy savings spent with communication and, consequently, maximizes the autonomy of the wireless sensor network. Recent works proposed data suppression techniques based on linear models, which presented a good suppression performance, with low complexity and computational cost, besides presenting a high degree of simplicity. Although efficient, these techniques consider only the temporal correlation of the data, disregarding the spatial correlation of the sensed physical quantities, typically present in several monitored phenomena, which may further reduce the total number of messages. This work consists of the proposal of a message suppression mechanism for wireless sensor networks that exploit the spatio-temporal correlation of the sensed physical quantities. The proposed method is evaluated in a simulation environment, involving real data from a large scale network, significantly dense, and with a relatively high sampling frequency. The results confirm that the method can improve message suppression rates and reduce energy consumption, reflected in the network lifetime.

Keywords: Wireless Sensor Networks, Data Suppression, Spatio-Temporal Correlation.

Lista de Figuras

Figura 1	Técnica de supressão de mensagens	26
Figura 2	Interface do <i>SmartData</i> (FRÖHLICH, 2018).	28
Figura 3	DBP (BARTON; MUSILEK, 2016).	30
Figura 4	Delayed DBP (BARTON; MUSILEK, 2016).	31
Figura 5	DBP With Look Ahead (BARTON; MUSILEK, 2016).	31
Figura 6	Tolerâncias de valor e tempo (RAZA et al., 2015).	35
Figura 7	Construção do modelo linear de previsão (RAZA et al., 2015).	36
Figura 8	Integração com SmartData.	45
Figura 9	Estrutura geral do dendograma de dissimilaridade.	49
Figura 10	Taxa de supressão dos parâmetros avaliados no DBP, ST-DBP W/oAgg, e ST-DBP Full.	59
Figura 11	Distribuição do número de regiões particionadas, com ϵ^{rel} de 1,5%.	60
Figura 12	Topologia da rede implantada pelo projeto Sensorscope.	62
Figura 13	Taxa de supressão para a série de temperatura.	63
Figura 14	Raiz do erro quadrático médio.	64
Figura 15	Ganho estimado em tempo de vida da rede para a série de temperatura.	65
Figura 16	Distribuição do número de regiões particionadas para a série de temperatura, com ϵ^{rel} de 5%.	66
Figura 17	Taxa de supressão para a série de irradiação solar.	66
Figura 18	Ganho estimado em tempo de vida da rede para a série de irradiação solar.	67
Figura 19	Distribuição do número de regiões particionadas para a série de irradiação solar, com ϵ^{rel} de 5%.	68

Lista de Tabelas

Tabela 1	Tabela comparativa dos trabalhos relacionados.....	34
Tabela 2	Parâmetros usados durante a avaliação.....	58
Tabela 3	Resumo das simulações com ϵ^{rel} de 1,5%.....	60
Tabela 4	Parâmetros usados durante a avaliação.....	63

Lista de Abreviaturas e Siglas

DBP	<i>Derivative-based Prediction</i>
DSCC	<i>Distributed Spatial Correlation-based Clustering</i>
EPFL	<i>École Polytechnique Fédérale de Lausanne</i>
EPOS	<i>Embedded Parallel Operating System</i>
HLM	<i>Hybrid Linear Model</i>
IoT	<i>Internet of Things</i>
LUCE	<i>Lausanne Urban Canopy Experiment</i>
MAC	<i>Medium Access Control</i>
MD	<i>Manhattan Distance</i>
PA	<i>Progressão Aritmética</i>
RMSE	<i>Root Mean Squared Error</i>
RNN	<i>Recurrent Neural Network</i>
RSSF	<i>Redes de Sensores sem Fio</i>
SEB	<i>Smallest Enclosing Balls</i>
SR	<i>Suppression Ratio</i>
ST-DBP	<i>Space-Time Derivative-based Prediction</i>
TSTP	<i>Trustful Space-Time Protocol</i>
WSN	<i>Wireless Sensor Networks</i>

Sumário

1	INTRODUÇÃO	19
1.1	OBJETIVOS	21
1.1.1	Objetivos específicos	21
1.2	METODOLOGIA	21
1.3	ESTRUTURA DO DOCUMENTO	22
2	FUNDAMENTAÇÃO	23
2.1	REDES DE SENSORES SEM FIO	23
2.2	SUPRESSÃO DE MENSAGENS	24
2.3	O TSTP E A API DO SMARTDATA	27
3	TRABALHOS RELACIONADOS	29
3.1	DERIVATIVE-BASED PREDICTION	35
4	SPACE-TIME DERIVATIVE-BASED PREDICTION	39
4.1	CONCEPÇÃO DO MÉTODO	40
4.1.1	Premissas	40
4.1.2	Modelo de Supressão	41
4.1.3	Dissimilaridade Entre Modelos	41
4.1.4	Smallest Enclosing Balls	42
4.1.5	Supressão de Mensagens em Regiões	43
4.2	PROJETO E IMPLEMENTAÇÃO	44
4.2.1	Algoritmo de Trickle	46
4.2.2	Manutenção dos Modelos	47
4.2.3	Bootstrap	48
4.2.4	Aglutinação de Regiões	49
4.2.5	Roteamento Geográfico	53
5	AVALIAÇÃO	57
5.1	MÉTRICAS AVALIADAS	57
5.2	RESULTADOS	58
5.2.1	Estudo de Caso 1	58
5.2.2	Estudo de Caso 2	61
6	CONSIDERAÇÕES FINAIS	69
	REFERÊNCIAS	71

1 INTRODUÇÃO

Redes de sensores sem fio (RSSF ou WSN, do inglês *Wireless Sensor Network*) é uma tecnologia emergente que vem ganhando atenção da comunidade científica e tecnológica mundial, principalmente nas últimas duas décadas. Pesquisas envolveram um diverso conjunto de tópicos, desde as camadas mais baixas dos protocolos de comunicação, até a camada mais alta, a das aplicações (AKYILDIZ et al., 2002).

A razão pela qual as RSSF tem ganhado atenção é a sua aplicabilidade em diversas áreas como: monitoramento ambiental; no monitoramento e controle de processos; automação residencial e industrial, na agricultura e pecuária de precisão, entre outras. Além disso, o rápido avanço tecnológico nas áreas de microeletromecânica, comunicação sem fio e eletrônica digital contribuiu significativamente para o evolução da área, permitindo o desenvolvimento de redes de sensores sem fio de larga escala, de baixo custo e eficientes.

Uma das principais características das RSSF é a disponibilidade limitada de recursos, principalmente energético. Assim, um dos desafios de se projetar soluções de RSSF é a maximização da autonomia de rede como um todo. Neste contexto, a comunicação desempenha um papel crítico, pois é a tarefa responsável pelo maior dreno de energia dos nodos da rede (RAZA et al., 2015).

As RSSF são utilizadas em aplicações de monitoramento e/ou controle, por meio de sensores e atuadores, respectivamente, acoplados aos nodos da rede. Uma característica comum às RSSF de monitoramento é a coleta de dados a uma taxa regular, nas quais os nodos realizam amostragem periódica de sensores e disseminam estas amostras pela rede com destino ao nodo concentrador, que coordena a rede, doravante denominado *sink*. Neste cenário, uma abordagem de comunicação inspirada na técnica de supressão de mensagens, por meio de predição de séries temporais, é uma potencial solução para maximizar a autonomia dos nodos da rede. No entanto, a aplicação deve tolerar uma pequena variação na precisão dos dados, que é uma suposição válida para a maioria das aplicações de RSSF (ISTOMIN et al., 2016).

Várias abordagens para economizar energia em RSSF foram propostas com foco na otimização da pilha de rede, sem a consciência dos dados transmitidos. A supressão de mensagens, por outro lado, é uma técnica, geralmente de nível de aplicação, para reduzir a energia

gasta com a comunicação que leva em consideração os dados transmitidos (ISTOMIN et al., 2016). Trabalhos anteriores utilizaram a supressão de mensagens com base em modelos de tendência para maximizar a autonomia de uma RSSF (i.e., para reduzir o consumo de energia e aumentar o tempo de vida da rede) (TULONE; MADDEN, 2006b; TULONE; MADDEN, 2006a; RAZA et al., 2012; RAZA et al., 2015; BARTON; MUSILEK, 2016; XU; ZHANG, 2017).

As soluções existentes, baseadas em modelos de tendência, se mostraram eficientes, reduzindo o número de mensagens enviadas na rede, maximizando a autonomia da rede. No entanto, consideram apenas a correlação temporal, desconsiderando a correlação espacial das grandezas físicas sensoreadas, o que poderia reduzir ainda mais o número total de mensagens.

Quando a correlação espacial da quantidade física sensoreada é muito baixa, os modelos de predição gerados por nodos sensores distintos, mas espacialmente próximos, tendem a se aproximarem um do outro (i.e, suas curvas apresentam-se próximas, com uma diferença pequena e aproximadamente constante, em um determinado intervalo de tempo). Neste caso, a principal questão de pesquisa abordada nesta dissertação é se a expansão da técnica de supressão baseada em modelos de tendência para explorar também a correlação espacial destas grandezas pode contribuir ainda mais para diminuição do número total de mensagens enviadas, reduzindo o consumo de energia dos nodos sensores, aumentando a autonomia da rede.

Neste trabalho, é proposta a expansão da técnica de supressão de mensagens, baseada em modelos de tendência, para explorar a correlação espacial das grandezas físicas sensoreadas, integrada a um protocolo de RSSF e à uma interface centrada em dados, com o intuito de otimizar ainda mais o uso de recursos em RSSF, de forma transparente para a aplicação. Além disso, é objetivo deste trabalho, avaliar a utilização da técnica na rede em termos das métricas de taxa de supressão de mensagens e tempo de vida da rede, relacionando-as com a precisão dos dados requisitada pela aplicação.

Partes deste trabalho foram previamente publicadas em:

- Space-Time Derivative-Based Prediction: A Novel Trickling Mechanism for WSN (RICHA et al., 2017)

1.1 OBJETIVOS

Este trabalho tem como objetivo principal demonstrar que a expansão da técnica de supressão de mensagens, baseada em modelos de tendência, para considerar a correlação espacial, juntamente com a sua integração a um protocolo de RSSF e uma interface centrada em dados, conduz à uma solução que, além de maximizar a autonomia de uma RSSF, provê uma interface que abstrai este mecanismo e garante os requisitos de qualidade dos dados, requisitada pelas aplicações de RSSF.

1.1.1 Objetivos específicos

De modo a alcançar o objetivo geral deste trabalho, são definidos os objetivos específicos a seguir.

1. Formalizar o método de supressão de mensagens proposto;
2. Implementar e integrar o método à um protocolo de RSSF e à uma interface centrada em dados, tanto em uma plataforma real de RSSF, quanto em um ambiente de simulação;
3. Executar experimentos para validar e avaliar o método em relação ao seu impacto em termos da supressão de mensagens e do tempo de vida da rede.

1.2 METODOLOGIA

Este trabalho de pesquisa caracteriza-se, metodologicamente, como uma pesquisa: aplicada, quanto à natureza, objetivando gerar conhecimentos, aplicáveis na solução de problemas reais; e exploratória, quanto aos objetivos, devido ao seu viés de inovação (GERHARDT; SILVEIRA, 2009).

A formalização do método proposto foi realizada pela descrição formal, matemática e algorítmica, dos elementos e mecanismos abarcados pelo método. Além disso, objetiva-se que todos os achados técnico-científicos sejam documentados e divulgados por meio de publicações técnico-científicas.

A implementação e integração, ao TSTP (RESNER, 2018) e à interface do SmartData, foi realizada em uma plataforma real de RSSF baseada no Sistema Operacional Paralelo e Embarcado (EPOS, do inglês *Embedded Parallel Operating System*) (LISHA, 2017a), para a plataforma EPOSMote III (LISHA, 2017b). A mesma implementação e integração foi realizada utilizando o simulador de eventos discretos OMNeT++ (OPENSIM, 2017) com um *framework* de simulação de RSSF, denominado Castalia (BOULIS, 2017), com o objetivo de permitir a sua avaliação. Ambas as plataformas, real e simulada, já contavam com uma implementação do TSTP. As implementações foram feitas em linguagem C++, já utilizada nas implementações de ambas as plataformas. Em seu desenvolvimento foram empregadas técnicas de orientação a objetos e meta-programação, visando código otimizado, conforme os requisitos inerentes ao cenário de RSSF.

A avaliação e validação da solução proposta ocorreu sobretudo, por meio de uma análise comparativa com uma solução base, em um ambiente de simulação. Neste sentido, foram analisadas métricas como a taxa de supressão e o ganho estimado em tempo de vida, e o impacto da qualidade de dados requisitada pela aplicação sobre estas métricas.

1.3 ESTRUTURA DO DOCUMENTO

O restante do documento está organizado conforme a seguir. O Capítulo 2 apresenta os principais conceitos, técnicas e artefatos relacionados à esta pesquisa. Na sequência, o Capítulo 3 apresenta os principais trabalhos relacionados, apresentando e discutindo diferentes aspectos destas soluções. No Capítulo 4, é apresentada a proposta deste trabalho. Após, o Capítulo 5 apresenta e discute a avaliação da técnica proposta. Por fim, o Capítulo 6, conclui este trabalho e aponta os trabalhos futuros.

2 FUNDAMENTAÇÃO

Neste capítulo realiza-se uma explanação quanto aos principais conceitos, técnicas e artefatos relacionados a este trabalho. Discorre-se inicialmente, na Seção 2.1, sobre redes de sensores sem fio, explorando os principais conceitos, características, e desafios. Na sequência, a Seção 2.2 aborda a fundamentação da técnica de supressão de mensagens em RSSF. Posteriormente, a Seção 2.3 apresenta e detalha o TSTP e a interface do SmartData, que juntos proveem mecanismos importantes para a concepção da solução proposta neste trabalho. Ambos, TSTP e SmartData, foram propostos anteriormente em trabalhos do grupo de pesquisa cujo autor desta dissertação é colaborador.

2.1 REDES DE SENSORES SEM FIO

Redes de sensores sem fio, como definem Potdar et al. (2009), compõem uma sub-classe das redes *ad-hoc*, e consistem em um conjunto de estações base (*sinks*) e eventualmente dezenas ou milhares de pequenos dispositivos de baixa potência, chamados nodos sensores (*motes*), distribuídos em um determinado espaço físico. Complementarmente, uma RSSF pode ser definida como uma infraestrutura composta por elementos de medição, de computação e de comunicação que, integrados, proveem a capacidade de monitorar aspectos do ambiente (e.g., temperatura, intensidade luminosa) e/ou de atuar neste ambiente por meio de atuadores (e.g., dimerização de lâmpadas, controle de dispositivos eletromecânicos).

Em comparação com as redes de computadores tradicionais, as RSSF se diferem em vários aspectos, como a grande quantidade de nodos, e as fortes restrições de energia, memória e processamento. Além disso, estas redes tendem a ser autônomas e requerem um alto nível de cooperação para operar e cumprir com os objetivos e requisitos da aplicação. Neste sentido, os desafios vão além dos comumente enfrentados em redes tradicionais (POTDAR et al., 2009).

Segundo Ruiz et al. (2003) as aplicações de RSSF podem ser classificadas, de acordo com a disseminação de dados, nas seguintes classes: programada, contínua, sob eventos, e sob demanda. Na abordagem programada, os nodos disseminam dados periodicamente, a uma taxa constante. No monitoramento contínuo, os nodos sensores

disseminam dados continuamente. A disseminação de dados, quando sob eventos, só é efetuada quando um evento ocorrer e uma determinada condição local for satisfeita. Já no modelo sob demanda, a disseminação de dados só ocorre quando houver uma solicitação do *sink*, por meio de uma mensagem explícita de consulta.

As RSSF focadas em monitoramento possuem, em sua grande maioria, a funcionalidade comum de sensoriamento periódico e contínuo a uma taxa constante. Neste modelo, a operação da rede induzirá um consumo mais rápido dos recursos disponíveis, restringindo a sua autonomia (RUIZ et al., 2003). Além disso, o fluxo de dados imposto por esta abordagem impacta no desempenho da rede de várias maneiras. Tal impacto pode ser percebido pela análise de métricas como carga da rede, latência, taxa de entrega de pacotes, entre outras.

O projeto de uma RSSF é significativamente dependente dos requisitos da aplicação, e devem ser considerados fatores como: tolerância a falhas, escalabilidade, custo de produção, ambiente de operação, topologia da rede, restrições físicas, meios de transmissão e consumo energético (AKYILDIZ et al., 2002). Neste contexto, os aspectos relacionados à eficiência energética, dentre outras métricas de desempenho, possuem relevante importância e apresentam-se críticos para a maioria das aplicações, necessitando assim de protocolos eficientes que busquem a otimização destas métricas. Dentre as principais métricas para avaliação de redes de sensores sem fio estão: eficiência de uso da energia e vida útil do sistema, carga da rede, escalabilidade, taxa de entrega de pacotes, tolerância a falhas, latência, *jitter*, e precisão dos dados (KHAN et al., 2012).

2.2 SUPRESSÃO DE MENSAGENS

Várias técnicas foram propostas para promover a conservação de energia em RSSF. Estas, segundo a taxonomia apresentada por Anastasi et al. (2009), podem ser classificadas como esquemas de *duty cycling* (fracionamento do período ativo do sistema de comunicação), esquemas orientados a dados, e esquemas baseados em mobilidade.

Várias abordagens para economizar energia em RSSF focam na otimização da pilha de rede, sem a consciência dos dados transmitidos (ISTOMIN et al., 2016). Normalmente, esquemas como os baseados em *duty cycling* e em mobilidade também apresentam esta característica. Neste sentido, abordagens baseadas em dados podem melhorar ainda mais a eficiência energética da rede.

Pelo prisma de sensoriamento, o consumo de energia nos nodos sensores é afetado de duas maneiras, pelo custo da aquisição de dados, e pela disseminação destes na rede. Os esquemas orientados a dados focam na eficientização da aquisição de dados de transdutores (e.g, amostragem adaptativa e hierárquica), ou na redução de dados (e.g, processamento em rede, compressão de dados e predição de dados).

Este trabalho concentra-se na sub-classe de redução de dados que explora a supressão de mensagens de dados por meio de técnicas de predição de séries temporais. Uma visão detalhada destas técnicas é apresentada por Anastasi et al. (2009) e Khan et al. (2015).

Grande parte das grandezas físicas monitoradas por RSSF possuem uma forte correlação espacial e temporal, como a temperatura, que de forma geral apresenta uma baixa variação espacial e uma tendência ao longo do tempo (e.g., sazonalidade). Neste caso, amostras redundantes resultam em consumo de energia não útil, mesmo quando o custo da amostragem é insignificante (i.e, desprezível). Além do custo energético de amostragem e transmissão destes dados na sua origem, é gerada uma perturbação na rede com comunicação desnecessária, aumentando o tráfego global e com isso a probabilidade de colisões, impactando na latência e, conseqüentemente, na taxa de entrega de pacotes. Ademais, o grande fluxo de dados em uma RSSF tende a restringir a escala da rede, por conta do efeito de afunilamento próximo ao *sink* (KHAN et al., 2015).

A supressão de mensagens, por meio de técnicas de predição de séries temporais, consiste na construção de uma abstração do fenômeno físico sensorado. Esta abstração é, comumente, desempenhada por um modelo matemático que descreve a tendência e evolução da série temporal (ANASTASI et al., 2009).

A ideia central da técnica de supressão de mensagens, ilustrada na Figura 1, é baseada em um mecanismo de predição de séries temporais incorporado nos nodos sensores e no *sink*. Para cada ponto de monitoramento (i.e., para cada grandeza monitorada por cada nodo sensor) existem duas instâncias de um modelo, um no próprio nodo sensor (i.e., na sua origem), e um no *sink*. Desta forma, a série temporal pode ser construída pela computação de uma sequência de modelos e não pelo envio explícito dos pontos da série temporal, sem exigir comunicação extra, reduzindo o tráfego da rede e contribuindo para a otimização do uso de recursos na rede.

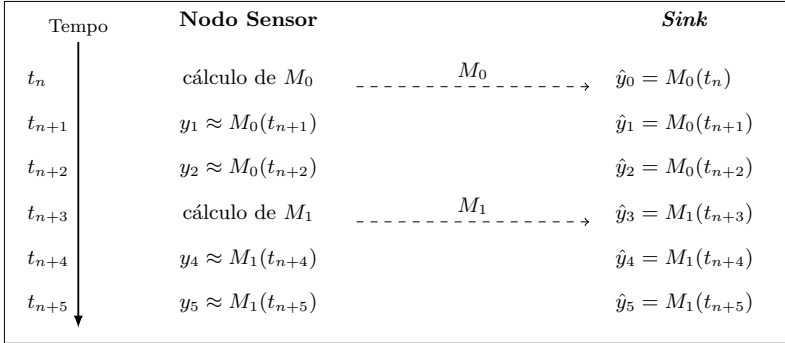


Figura 1 – Técnica de supressão de mensagens

Inicialmente, no instante t_n , o nodo sensor calcula um modelo matemático M_0 e o envia ao *sink*. A partir deste momento, a cada período (instantes t_n , t_{n+1} e t_{n+2}), ambos são capazes de estimar os mesmos valores pela avaliação deste modelo no tempo ($\hat{y}_t = M_0(t_{n+t})$).

A utilização do modelo gerado está condicionada a validade de sua representação do fenômeno em um dado instante. Assim como no mecanismo tradicional (i.e., sem supressão de mensagens), o nodo sensor continua coletando dados periodicamente, mas agora comparando estes dados reais com as estimativas ($y_t \approx? M(t)$), extraídas do modelo, com o intuito de avaliar a sua representatividade e, portanto, sua validade.

Quando o nodo sensor detecta um desvio do modelo em relação a realidade (i.e., o erro da estimativa em relação ao dado coletado está além da tolerância definida pelos requisitos da aplicação), um novo modelo M_1 é construído e enviado novamente ao *sink*, garantindo a sua eficácia, conforme ilustrado no instante t_{n+3} . Por outro lado, quando a estimativa satisfaz os requisitos da aplicação, nenhuma mensagem é necessária, como pode ser visto nos instantes de t_{n+1} à t_{n+2} e de t_{n+4} à t_{n+6} , reduzindo a taxa de comunicação sem diminuir a qualidade de serviço da aplicação.

O processo de construção do modelo, incluindo os recursos de *software* e *hardware* necessários para executá-lo são dependentes da técnica específica de predição dos dados. Um exemplo simples é a técnica, denominada por Tulone e Madden (2006b) como *approximate caching*, na qual o sensor considera um modelo constante, que é atualizado somente quando a diferença entre o dado extraído do modelo e o dado atual for maior que a tolerância definida pela aplicação, distanciando-se do modelo. Esta técnica requer pouco recurso de memória e processamento.

2.3 O TSTP E A API DO SMARTDATA

Trustful Space-Time Protocol (TSTP), é um protocolo *cross-layer*, orientado a aplicação, para Sistemas Ciber-Físicos em RSSF (RESNER, 2018). O TSTP concentra-se na entrega eficiente de funcionalidades recorrentemente necessárias por estes sistemas: dados confiáveis, temporizados, geo-referenciados, conformantes com o sistema internacional de unidades (SI), entregues para a aplicação por meio de uma interface que provê uma visão da rede centrada em dados. O seu *design* integra sincronização de tempo, localização, segurança, controle de acesso ao meio (MAC, do inglês *Medium Access Control*), e roteamento.

No TSTP, a comunicação ocorre entre os sensores e o *sink*, que possui interesse em informações de regiões do espaço-tempo. Os nodos possuem a capacidade de mensurar um ou mais tipos de variáveis do ambiente com uma certa precisão e taxa de amostragem, e são capazes de relatar essas medições ao *sink* (RESNER; FRÖHLICH, 2015). O interesse do *sink* é anunciado por uma mensagem de interesse, de acordo com a semântica do *SmartData* (FRÖHLICH et al., 2013).

Um *SmartData* (em português, dado inteligente) é um dado enriquecido com metadados suficientes para torná-lo independente em termos de semântica, localização espacial, tempo e confiança. Os aspectos semânticos do *SmartData* são descritos usando uma estratégia inspirada no padrão IEEE 1451 (IEEE 1451.0, 2007).

A interface do *SmartData*, apresentada na Figura 2, encapsula um poderoso mecanismo para abstrair qualquer tipo de sensor e atuador em um sistema, locais ou remotos. Em ambos os casos, a ligação de um objeto *SmartData* com um transdutor é feita por meio do parâmetro de classe *Transducer*, que deve declarar constantes, denominadas *UNIT* e *ERROR*. O primeiro identifica a quantidade SI correspondente, e o segundo define o erro absoluto do transdutor.

Por exemplo, um objeto *SmartData* pode abstrair um sensor de temperatura, ou um ar condicionado (i.e., um atuador), com precisão de $\pm 1,5 K$, com um transdutor que especifica K (Kelvin) como *UNIT* e 1,5 como *ERROR*. Além de um *Value* e uma *UNIT*, cada instância de *SmartData* possui uma coordenada do local onde o dado foi produzido.

O *SmartData* fornece uma abstração completa do sistema de sensoriamento/atuação e de comunicação. Todos os serviços no nível do sistema, incluindo comunicação, sincronização e interação com transdutores são implicitamente intermediados, considerando os parâmetros especificados no momento da instanciação.

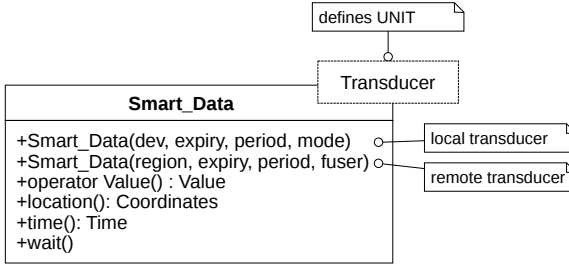


Figura 2 – Interface do *SmartData* (FRÖHLICH, 2018).

Um *SmartData* local é instanciado com o primeiro construtor, descrito na Figura 2. O primeiro parâmetro do construtor, *dev*, é usado para diferenciar várias instâncias do mesmo transdutor disponível em um determinado nodo. O segundo parâmetro, *expiry*, define a validade dos dados expressos em relação ao tempo em que foram produzidos. O último parâmetro, *mode*, determina a visibilidade do *SmartData* e a possibilidade de ser comandado, no caso de um atuador.

Um *SmartData* remoto é instanciado com o segundo construtor, representado na Figura 2, que são *proxies* locais para transdutores remotos. Todos os procedimentos em nível de sistema necessários para garantir uma semântica consistente para tais objetos são transparentemente executados pela implementação, incluindo comunicação, sincronização e escalonamento de tarefas.

O primeiro parâmetro do construtor é usado para especificar uma região de interesse no espaço-tempo como: (x, y, z, r, t_0, t_f) , onde (x, y, z) definem o centro da região de interesse, r o seu raio e $[t_0, t_f]$ o intervalo de tempo do interesse. A região de interesse é usada em combinação com a quantidade física associada à *UNIT*. Assim, sua instanciação, implica no anúncio de um interesse por uma quantidade física em uma determinada região de espaço (expressa por uma esfera) para um determinado intervalo de tempo. Os objetos *SmartData* (criados com o primeiro construtor) que correspondem aos critérios responderão ao interesse de acordo com o parâmetro de período.

Independentemente de como um *SmartData* é criado, três métodos comuns sempre podem ser invocados nele: um operador de conversão de tipo nativo (*operator Value()*), *location()* e *time()*. O primeiro permite conversões implícitas de (e para) o tipo de dado encapsulado (e.g, *integer*, *float*). O segundo e o terceiro retornam, respectivamente, onde e quando o dado foi produzido.

3 TRABALHOS RELACIONADOS

Diferentes abordagens para otimizar o uso dos recursos em RSSF foram propostas com foco na otimização da pilha de rede, sem a consciência dos dados transmitidos. A supressão de mensagens, por outro lado, é uma técnica, geralmente de nível de aplicação, utilizada para reduzir a energia gasta com comunicação que leva em consideração os dados transmitidos (ISTOMIN et al., 2016). Trabalhos recentes utilizaram a supressão de mensagens, com base em modelos lineares de tendência, se destacaram pela baixa complexidade e custo computacional, capacidade de supressão de mensagens, e consequentemente ganho em autonomia da rede (RAZA et al., 2015; XU; ZHANG, 2017). Assim, esta seção apresenta as técnicas do estado da arte, baseadas em modelos lineares de tendência cujo principal objetivo é a maximização do tempo de vida da rede.

Com o intuito de suportar a otimização da coleta de dados, Liu et al. (2013) propuseram o algoritmo *Distributed Spatial Correlation-based Clustering* (DSCC) (LIU et al., 2013), agrupando sensores com leituras semelhantes no mesmo *cluster*. O algoritmo é dividido em três etapas, a de seleção de *cluster heads*, a formação dos *cluster*, e a de manutenção destes. Dois parâmetros são definidos pela aplicação, um limite de tolerância a erros, e um raio de correlação.

O mecanismo de agrupamento inicia com uma mensagem do *sink*, que informa os parâmetros definidos pela aplicação. Nodos vizinhos, considerando o raio de correlação, trocam informações sobre energia remanescente e uma série de dados recentes. Cada nodo pode então se candidatar ao papel de *cluster head*, considerando métricas como o nível de energia remanescente e a sua capacidade de representar os dados dos vizinhos. Para definir os *cluster heads* finais, todos os candidatos executam um algoritmo distribuído de competição, baseado em *ranking*, que descobre os candidatos mais apropriados. Ao final, os nodos com *ranking* máximo tornam-se *cluster head*, e os demais decidem qual o *cluster* mais adequado para se associarem.

Raza et al. (2015) propuseram uma técnica de supressão de mensagens, denominada *Derivative-based Prediction* (DBP), baseada em modelo de tendência que parte da premissa de que uma sequência de valores medidos por um nodo sensor pode ser aproximada, em um curto espaço de tempo, por um modelo linear. Este modelo representa os dados com um erro que é limitado por uma tolerância, determinada pelos requisitos de qualidade dos dados da aplicação.

Cada nodo sensor executando o DBP constrói uma janela de aprendizagem \mathcal{W} , formada pelos últimos w valores medidos. A cada período de dado, o nodo sensor atualiza \mathcal{W} incluindo a última medição. Quando \mathcal{W} atinge o tamanho máximo w , o nodo sensor constrói o primeiro modelo \mathcal{M} , calculando a inclinação α da reta entre dois pontos. Estes pontos são obtidos, respectivamente, pelas médias aritméticas dos l pontos mais antigos de \mathcal{W} e dos l pontos mais recentes de \mathcal{W} , nomeados pontos de borda. Este modelo é então disseminado na rede com destino ao *sink*, que passa a usá-lo para predição de valores futuros, sem a necessidade de comunicação adicional (RAZA et al., 2015).

A cada período o nodo sensor realiza uma medição, atualiza a janela de aprendizagem e verifica a validade do modelo atual. Se o erro da predição do modelo for superior aos limites definidos pelos requisitos da aplicação, o nodo sensor cria um novo modelo baseado na janela de aprendizagem atual, e então dissemina-o na rede com destino ao *sink*, que ao recebê-lo irá substituir o modelo anterior pelo novo. Caso contrário, nenhuma comunicação é necessária.

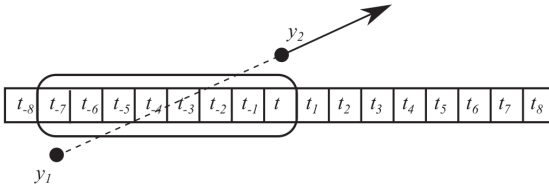


Figura 3 – DBP (BARTON; MUSILEK, 2016).

Raza et al. (2015) realizaram experimentos de nível de aplicação utilizando sete conjuntos de dados reais e DBP. Nestes experimentos, a DBP obteve uma taxa de supressão de mensagens de até 99%, apresentando o melhor desempenho em cinco dos sete conjuntos de dados, em comparação à outras técnicas apresentadas.

Duas variações da DBP foram propostas por Barton e Musilek (2016), chamadas *Delayed DBP* e *DBP with look ahead* (BARTON; MUSILEK, 2016). A *Delayed DBP* insere um atraso na janela de aprendizagem, com base na premissa de que o cálculo da inclinação de uma linha que começa no passado e termina no futuro resulta em melhores predições. Desta forma, é possível aumentar a taxa de supressão, uma vez que o *Delayed DBP* está ciente do futuro próximo.

O DBP original foi expandido, e um novo parâmetro, *delay*, foi definido. Este parâmetro determina de quantos pontos amostrais será

o atraso inserido na janela de aprendizagem. Assim como no DBP original, amostras reais são utilizadas para calcular os parâmetros do modelo, porém com o objetivo de prever pontos no passado.

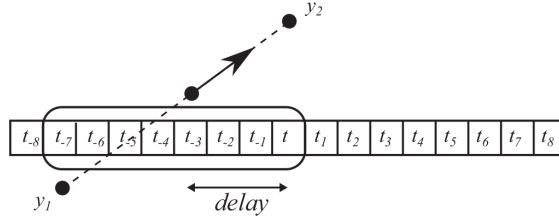


Figura 4 – Delayed DBP (BARTON; MUSILEK, 2016).

A variação *Delayed DBP* foi avaliada, em termos do número de atualizações de modelos, sob dados de temperatura. Nos experimentos conduzidos, o melhor resultado correspondeu à uma redução em 40% das atualizações de modelos comparado à versão original, ao custo da introdução de um atraso de 7,5 minutos, com uma tolerância de 0,24°C. Apesar de maximizar a supressão de mensagens, esta técnica não é aplicável à sistemas de tempo real, devido à percepção ligeiramente atrasada da realidade do ponto de vista do *sink*. Assim, sua aplicabilidade está diretamente relacionada às características e requisitos da aplicação.

Já a *DBP with look ahead*, é uma melhoria do *Delayed DBP* para viabilizar a sua utilização em aplicações de tempo real, que são sensíveis ao atraso imposto pelo *Delayed DBP* (BARTON; MUSILEK, 2016). Ao invés de atrasar a janela de aprendizado, ela usa uma Rede Neural Recorrente (RNN, do inglês *Recurrent Neural Network*) para prever os pontos futuros, que serão a base para construir o modelo linear.

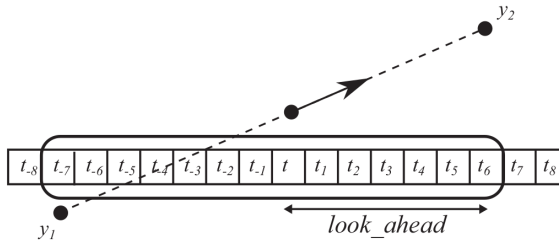


Figura 5 – DBP With Look Ahead (BARTON; MUSILEK, 2016).

Assim como no DBP original, e diferentemente do *Delayed DBP*, o ponto referência para geração do modelo permanece no tempo atual. A principal diferença está no deslocamento da borda direita da janela de aprendizagem para o futuro, cujo deslocamento é definido pelo novo parâmetro *look ahead* introduzido, e os valores futuros preditos pela RNN, uma vez que não se tem conhecimento dos valores futuros reais.

Para o mesmo conjunto de dados, *DBP with look ahead* foi capaz de reduzir as atualizações de modelos em 9% em comparação ao DBP original, fixando o parâmetro *look ahead* em 6. Com um ganho significativamente menor que a *Delayed DBP*, a *DBP with look ahead* possui a melhoria da taxa de supressão limitada pelo desempenho do *Delayed DBP*, uma vez que o resultado da execução do *Delayed DBP* é equivalente à execução do *DBP with look ahead* considerando uma RNN oráculo (i.e., uma RNN perfeita, cuja acurácia de predição é máxima em qualquer momento da execução). Nenhuma análise foi realizada em termos do custo computacional desta solução, que intuitivamente é maior, devido à RNN embarcada em dispositivos com limitações de recursos.

Xu e Zhang (2017) propuseram o *Hybrid Linear Model* (HLM), que consiste na combinação de reconstrução e predição de séries temporais. No sensor, o HLM possui uma fase de treinamento do modelo e uma de predição. Por outro lado, no *sink*, o HLM possui uma fase de reconstrução, e uma de predição da série temporal.

Na fase de treinamento, o modelo é construído considerando um atraso dinâmico limitado por um parâmetro de máximo atraso. O modelo é construído utilizando o método dos mínimos quadrados para minimizar o erro da predição. Inicialmente, são utilizados as últimas amostras, e então o modelo é refinado iterativamente, aumentando o atraso de treinamento a cada iteração, até que o erro da predição esteja dentro dos limites definidos pela aplicação, ou quando o atraso for o máximo permitido. A fase de predição usa o modelo para prever valores futuros e verificar a sua validade. Na fase de reconstrução, o *sink* o utiliza para reconstruir a série temporal considerando o atraso, e então passa a utilizá-lo para prever valores futuros.

Assim como no *Delayed DBP*, o HLM insere um atraso na série temporal, do ponto de vista do *sink*. Apesar do atraso ser dinâmico e limitado por um parâmetro de máximo atraso, ainda existem limitações relacionadas à sua aplicabilidade em aplicações de tempo real.

De forma geral, cada abordagem proposta pelos trabalhos relacionados apresenta algum inconveniente. Soluções baseadas em modelos de tendência, em particular os que utilizam modelos lineares,

apresentam um bom *trade-off* entre desempenho de supressão, precisão dos dados, e complexidade. No entanto, apesar de eficientes, soluções anteriores não exploraram a correlação espacial inerente de muitas grandezas físicas, comumente monitoradas em RSSF.

Soluções baseadas em *cluster* foram propostas para dar suporte a coleta de dados aproximada, eventualmente explorando esta correlação espacial. No entanto, tais soluções comumente apresentam custos adicionais significativos para criação e manutenção de *clusters*, além do *overhead* de eleição de *cluster heads*.

Dada a simplicidade e baixo custo computacional das técnicas baseadas em modelos de tendência, expandi-las para explorar também a correlação espacial das grandezas físicas sensoreadas poderia aumentar ainda mais a autonomia da rede. Quando a variação espacial da quantidade física sensoreada é baixa, modelos gerados por nodos sensores distintos, espacialmente próximos, tendem a apresentar uma forte correlação.

Assim, este trabalho consiste na proposição de uma solução para economia de energia em RSSF, por meio da supressão de mensagens, explorando também a correlação espacial destas séries temporais. A Tabela 1 apresenta um breve comparativo das técnicas apresentadas. O posicionamento deste trabalho em relação aos relacionados também é apresentado.

Neste sentido, as abordagens propostas são analisadas em relação: ao Gerenciamento, se a criação e avaliação de modelos é feita por nodo (localizada), entre nodos (distribuída), ou no *sink* (centralizada); a explorar a correlação Espacial e/ou Temporal; em relação aos mecanismos utilizados; e a sua aplicabilidade em soluções de Tempo Real.

Tabela 1 – Tabela comparativa dos trabalhos relacionados.

	Gerenciamento	Temporal	Espacial	Mecanismos	Tempo Real
Liu et al. (2013)	–	Não	Sim	<i>Clustering</i>	–
Raza et al. (2015)	Localizada	Sim	Não	Modelo Linear	Sim
Barton e Musilek (2016)*	Localizada	Sim	Não	Modelo Linear	Não
Barton e Musilek (2016)*	Localizada	Sim	Não	Modelo Linear e RNN	Não
Xu e Zhang (2017)	Localizada	Sim	Não	Modelo Linear	Dependente da característica dos dados e do parâmetro de limite
Proposta	Híbrido	Sim	Sim	Modelo Linear	Sim

**Delayed DBP*

**DBP with look ahead*

3.1 DERIVATIVE-BASED PREDICTION

Esta seção apresenta de forma mais detalhada alguns aspectos importantes do *Derivative-Based Prediction* (DBP), que é tomado como base para o desenvolvimento deste trabalho. Dentre os aspectos apresentados estão os níveis de tolerância definidos pela aplicação, a definição matemática do modelo de predição, e o processo de construção, manutenção e utilização deste modelo.

O DBP é um método de supressão de mensagens baseado em predição de séries temporais, compatível com aplicações que possuem restrições relacionadas à qualidade dos dados. Ela é adequada para aplicações de RSSF limitadas em energia, e é mais simples que outras técnicas existentes (RAZA et al., 2015).

O DBP possui o pressuposto de que o dado medido por um nodo sensor pode ser aproximado, em curto a médio intervalo, por um modelo de precisão linear. Este modelo linear representa os dados com um erro, limitado por uma margem de tolerância, a qual deve ser parte dos requisitos da aplicação.

Um intervalo de tempo, no qual um desvio do modelo pode ocorrer, é definido como \mathcal{E}_T . Já a margem de tolerância é definida como $\mathcal{E}_V = (\epsilon^{rel}, \epsilon^{abs})$, onde ϵ^{rel} é o erro relativo tolerado e ϵ^{abs} é o erro absoluto tolerado. Ambos os parâmetros de tolerância de valor e tempo estão ilustrados na Figura 6 e formalizados pelas Definições 1 e 2 (RAZA et al., 2015).

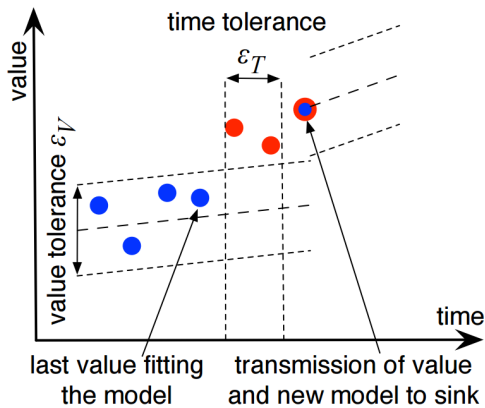


Figura 6 – Tolerâncias de valor e tempo (RAZA et al., 2015).

Definição 1. Seja V_i um valor sensoreado obtido no tempo t_i e \mathcal{E}_V a tolerância de valor. Da perspectiva da aplicação, ler um valor V_i se torna equivalente a ler qualquer valor \hat{V}_i em um intervalo R_V definido pelo erro máximo aceitável, $\hat{V}_i \in R_V$ se, e somente se, $\hat{V}_i \in [V_i - \epsilon, V_i + \epsilon]$, onde $\epsilon = \max\{\frac{V_i}{100} \epsilon^{rel}, \epsilon^{abs}\}$. Em outras palavras, a aplicação considera o valor $\hat{V}_i \in R_V$ como correto.

Definição 2. Seja $\Delta T = (t_k - t_j)$ um intervalo de tempo, e $\hat{V}_{\Delta T} = \{\hat{V}_j, \dots, \hat{V}_k\}$ o conjunto de valores reportados para a aplicação durante ΔT . A tolerância de tempo \mathcal{E}_T é o valor máximo aceitável de ΔT tal que todos os valores reportados neste intervalo são considerados incorretos, i.e., $\hat{V}_i \notin R_V, \forall \hat{V}_i \in \hat{V}_{\Delta T}$.

Em muitas aplicações, a amplitude de variação de uma série temporal pode ser alta, de modo que os valores medidos podem ser muito baixos ou muito altos. Um exemplo disto é a irradiação global, que pode ter uma amplitude de $1 W/m^2$ até mais de $1000 W/m^2$. Esta é a razão pela qual Raza et al. (2015) definiram a tolerância de valor em termos de erros relativos e absolutos. Assim, é possível especificar erros máximos absolutos e relativos, para ambos os extremos.

Cada nodo sensor executando DBP constrói um modelo linear que, ao ser mantido sincronizado com o *sink*, permite a supressão de mensagens por meio da estimativa dos valores reais lidos pelo nodo. Este modelo é gerado a partir de uma janela de aprendizagem \mathcal{W} , que é formada pelas w medições mais recentes. Este processo está ilustrado na Figura 7, e um detalhamento matemático é apresentado a seguir.

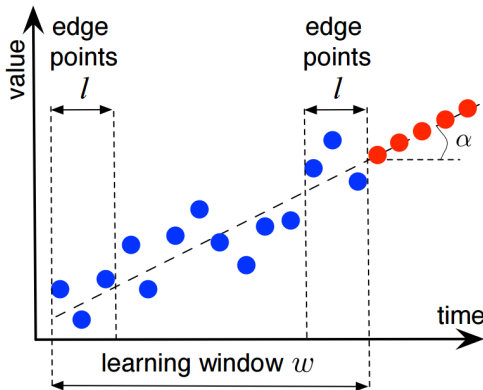


Figura 7 – Construção do modelo linear de predição (RAZA et al., 2015).

Seja \mathcal{W}_t a janela de aprendizagem em um dado tempo t , w' é o tamanho atual da janela, e p é um dado período, temos que: $\mathcal{W}_t = \{V_{t-((w'-1)*p)}, \dots, V_t\}$, para todo $w' > 1$. Quando \mathcal{W} atinge o tamanho máximo w , o nodo sensor constrói o primeiro modelo \mathcal{M} , calculando a inclinação α da linha que atravessa dois pontos definidos. O primeiro ponto corresponde à média dos l pontos mais antigos de \mathcal{W} e o segundo ponto à média dos l mais recentes pontos de \mathcal{W} , denominados pontos de borda. Assim, um modelo linear \mathcal{M} gerado no instante $t_0 + p$ pode ser descrito em função do tempo $y(t)$ como: $\hat{V}_t(t) = \alpha * (t - t_0) + \beta$, onde β é a média dos l pontos mais antigos de \mathcal{W} .

Este modelo então é enviado ao *sink*, o qual é considerado válido. A cada período, o nodo sensor atualiza a janela de aprendizagem como uma nova medida V_t e compara esta com o valor obtido pelo DBP \hat{V}_t utilizando o modelo atual \mathcal{M} . Se o valor predito está dentro da tolerância \mathcal{E}_V , \mathcal{M} permanece sendo um modelo válido e o *sink* continua gerando valor com base neste modelo. Por outro lado, se as medidas reais estão fora dos intervalos de tolerância \mathcal{E}_V e \mathcal{E}_T , um novo modelo \mathcal{M}' é gerado pelo nodo sensor, com base nas últimas w medidas, e enviado ao *sink*, que passa a considerá-lo em detrimento do modelo anterior.

4 SPACE-TIME DERIVATIVE-BASED PREDICTION

Este capítulo apresenta a proposta do ST-DBP, uma extensão da técnica de supressão de mensagens, baseada em modelos de tendência, para considerar supressão de mensagens em regiões do espaço e tempo. Neste capítulo, após uma introdução à proposta, duas seções a detalham. A Seção 4.1 contém o detalhamento da concepção do método e suas propriedades. Já a Seção 4.2 apresenta aspectos relacionados ao projeto e implementação da técnica, além da sua integração tanto à uma plataforma real de RSSF quanto a um ambiente de simulação.

No contexto das RSSF de monitoramento, as séries temporais de grandezas físicas sensorizadas possuem uma forte correlação espacial e temporal. Várias técnicas foram propostas principalmente para reduzir o consumo de energia, por meio da supressão de mensagens, com predição de dados, explorando esta característica.

Propostas mais recentes, baseadas em modelos de tendência, mostraram simplicidade e eficiência. No entanto, tais propostas não consideram a correlação espacial das quantidades físicas. Assim, a proposta deste trabalho é a expansão de uma técnica do estado-da-arte de supressão de mensagens baseada em modelos de tendência. Ao invés de aplicar a supressão de mensagens nodo a nodo, a proposta é aplicar a supressão de mensagens em regiões contendo vários nodos.

A ideia central desta proposta é agrupar nodos que possuem modelos de predição semelhantes (i.e., gerados a partir de dados que possuem correlação espacial), geograficamente próximos. Para tanto, propõe-se um mecanismo de agrupamento geográfico, dividido em três partes: a inicialização (*bootstrap*), o particionamento de regiões; e a aglutinação de regiões particionadas. O método assemelha-se à técnicas clássicas de *clustering*, porém sem o *overhead* de criação e manutenção de clusters, além da eleição de *cluster heads*.

Na inicialização, cada nodo gera seu próprio modelo, baseado nas suas amostras, e o envia na rede. Nodos vizinhos recebem este modelo, e o avaliam com o objetivo de detectar uma correlação. Cada nodo que detectar uma correlação, assume este modelo como referência, formando uma região de supressão, composta por estes nodos, e representada por este modelo. O particionamento desta região ocorre, quando algum nodo pertencente a esta, detecta um desvio deste modelo, disseminando um modelo próprio. Novamente,

nodos mais próximos, avaliam este novo modelo para detectar correlação. Por fim, a aglutinação ocorre quando o *sink* detecta correlações entre regiões já particionadas, enviando uma mensagem, com um modelo, para aglutinação destas regiões.

A técnica Derivative-Based Prediction (DBP), proposta por Raza et al. (2015), a qual é tomada como ponto de partida para este trabalho, é limitada à supressão de apenas um nodo sensor por modelo de predição. No entanto, da perspectiva da aplicação executando TSTP, existe a premissa de que se o dado obtido provém de uma região do espaço e intervalo de tempo desejados, é confiável, e está dentro dos níveis de tolerância definidos pelos requisitos da aplicação, não importa em qual sensor o dado tenha sido gerado (RESNER; FRÖHLICH, 2015). Esta premissa, que é válida para a maioria das aplicações de sensoriamento, permite que o DBP seja expandido para alcançar a supressão de mensagens para regiões de espaço-tempo e não para nós individuais.

4.1 CONCEPÇÃO DO MÉTODO

Esta seção contém o detalhamento da concepção do método e suas propriedades. Mais especificamente, são apresentados: as premissas gerais para sua aplicação; o novo modelo de supressão; o conceito de dissimilaridade entre modelos; o problema da Bola de Fechamento Mínima e a sua aplicação; e o conceito de supressão de mensagens em regiões do espaço e tempo.

4.1.1 Premissas

O método proposto possui uma série de propriedades que são dependentes das características da aplicação, e demais componentes que compõem a RSSF. Neste sentido, para que seja possível implementar o método proposto em uma RSSF real, algumas premissas precisam ser estabelecidas. A seguir estão elencadas as principais premissas:

- A aplicação suporta que os dados monitorados possuam um pequeno erro, definido por um parâmetro de tolerância;
- O protocolo de rede prevê sincronização temporal;
- Os nodos da RSSF que executam o ST-DBP são estáticos e possuem conhecimento da sua posição geográfica;

- O *sink* é um dispositivo computacional de maior robustez;
- O protocolo é *cross-layer* e permite explorar e atuar nos seus diferentes níveis.

Além destas questões, existem aspectos de segurança, que não são foco deste trabalho. Com o objetivo de preservar princípios de segurança dos dados, como confidencialidade, estes são transmitidos criptografados, no entanto para que o ST-DBP funcione corretamente, outros nodos precisam ter acesso a estes dados. Neste caso, uma possível solução é a utilização de chaves de grupo, de modo a permitir que um grupo de nodos em uma determinada região do espaço possam ler dados de determinadas mensagens.

4.1.2 Modelo de Supressão

No ST-DBP, cada nodo $n \in \mathcal{N}^{\mathcal{R}}$, pertencente à região de interesse \mathcal{R} , cria uma tupla $\langle \alpha, \beta, \tau, \theta \rangle$, onde α e β são os coeficientes do modelo, τ é o instante no qual o modelo foi gerado, e θ é a coordenada (x, y, z) da origem do modelo. Neste ponto, a diferença entre o ST-DBP em comparação ao DBP é a componente espacial θ presente no modelo, além da forma com que os nodos realizam sua sincronização, que ocorre entre os nodos e o *sink*, cujo mecanismo será apresentado ao final desta seção.

4.1.3 Dissimilaridade Entre Modelos

O ST-DBP conta com um mecanismo de aglutinação de regiões, que será apresentado na Seção 4.1.5, onde ocorre a fusão de modelos de supressão. Estes modelos são avaliados por meio de uma métrica de dissimilaridade que, quanto mais alto for o seu valor, maior será a dissimilaridade entre os modelos.

Neste contexto, a métrica denominada *Manhattan Distance* (md), definida na Equação 4.1, foi adotada como métrica de dissimilaridade. A md é definida pela média aritmética dos valores absolutos das diferenças entre os termos de duas séries numéricas, que é adotada neste trabalho como o grau de dissimilaridade.

$$md(\mathcal{M}_a, \mathcal{M}_b) = \frac{\sum_{i=1}^k |\hat{V}_i^a - \hat{V}_i^b|}{k} \quad (4.1)$$

Quando aplicada sobre modelos lineares de supressão, a série numérica composta pelos módulos das diferenças das previsões destes modelos pode ser expressa pela Equação 4.2. Tal série numérica corresponde à uma progressão aritmética (PA), uma vez que os modelos são lineares e possuem, conseqüentemente, coeficientes angulares constantes.

$$S = (|\hat{V}_1^a - \hat{V}_1^b|, |\hat{V}_2^a - \hat{V}_2^b|, \dots, |\hat{V}_k^a - \hat{V}_k^b|) \quad (4.2)$$

O somatório de todos os elementos de uma PA pode ser expresso como $(n * (a_1 + a_n)/2)$, onde n é o número de elementos da série, e a_1 e a_n são, respectivamente, o primeiro e o último termo desta série. Deste modo, a métrica de dissimilaridade, introduzida pela Equação 4.1, pode ser tomada como a média aritmética de uma PA, e reescrita conforme a Equação 4.3, onde S_1 e S_n são, respectivamente, o primeiro e o último termo da PA definida pela Equação 4.2.

$$md(n_a, n_b) = \frac{S_1 + S_n}{2} = \frac{|\hat{V}_1^a - \hat{V}_1^b| + |\hat{V}_k^a - \hat{V}_k^b|}{2} \quad (4.3)$$

No processo de aglutinação de regiões, que será apresentado na Seção 4.1.5, onde ocorre a fusão de modelos, estes são avaliados de acordo com a métrica de dissimilaridade. Dois modelos, \mathcal{M}_a e \mathcal{M}_b , são considerados similares se, e somente se, a métrica de dissimilaridade for menor ou igual à um coeficiente máximo de dissimilaridade, conforme Equação 4.4

$$md(n_a, n_b) \leq \psi \varepsilon \quad (4.4)$$

O coeficiente máximo de dissimilaridade é definido como uma porção ψ do erro máximo aceitável pela aplicação. Este parâmetro ψ é dependente da aplicação e das características específicas da grandeza física sensorada e, portanto, deve ser definido criteriosamente, em tempo de projeto, por um especialista no comportamento de tal grandeza, com base em ensaios experimentais.

4.1.4 Smallest Enclosing Balls

O problema denominado Bola de Fechamento Mínima (SEB, do inglês *Smallest Enclosing Balls*), consiste na construção da menor esfera que englobe um determinado conjunto de pontos. Este problema foi

apresentando inicialmente em 1857 (GÄRTNER, 1999). Desde então diferentes soluções foram propostas para solucionar tal problema, tais como as apresentadas por Welzl (1991), Gärtner (1999), Fischer et al. (2003) e Martinetz et al. (2006).

Neste trabalho, foi utilizada a solução proposta por Gärtner (1999), uma melhoria de Welzl (1991) que resolve o problema em tempo linear e apresenta bom desempenho para dimensão menores ou igual à vinte. Uma implementação escrita em linguagem C++ foi disponibilizada pelo autor¹, a qual foi refatorada e acoplada à solução proposta neste trabalho.

4.1.5 Supressão de Mensagens em Regiões

Neste contexto, foi introduzido o conceito de região espaço-tempo de supressão, a qual corresponde a uma aglutinação de nodos representados por um mesmo modelo, denominado "modelo de região". Este conceito é formalizado pela Definição 3.

Definição 3. Seja $\mathcal{N}^{\mathcal{R}}$ um conjunto finito de nodos em uma região de interesse \mathcal{R} . Uma região espaço-tempo de supressão T é um conjunto de nodos $T \in 2^{\mathcal{N}^{\mathcal{R}}} \setminus \emptyset$ tal que $\forall n \in T, \neg \exists m : \mathcal{M}_m^r$ que tenha sido gerado mais próximo de n do que \mathcal{M}_T , onde n e m são nodos, \mathcal{M}_m^r é o modelo de região de m , e \mathcal{M}_T é o modelo da região de supressão T .

O ST-DBP assume, inicialmente, que $T = \mathcal{N}^{\mathcal{R}}$ e que \mathcal{M}_T é o primeiro modelo gerado e anunciado por um nodo $n \in \mathcal{R}$. Assim, $\forall n' \in \mathcal{R}$ que ouvir este anúncio, o modelo de região de n' será \mathcal{M}_T . Consequentemente, o primeiro nodo n a anunciar um modelo local \mathcal{M}_n^l (i.e., modelo local de n) representará a região de supressão à qual pertence com o seu modelo local (i.e., $\mathcal{M}_T = \mathcal{M}_n^r = \mathcal{M}_n^l$).

Este nodo, ao anunciar as atualizações de seu modelo local, manterá a sincronização, do modelo de região, com os outros nodos. Por outro lado, cada um dos demais nodos verifica, a cada período, o seu modelo de região, descartando o anúncio do seu modelo local enquanto não ocorrer um desvio do modelo de região.

O particionamento da região ocorre de forma distribuída, quando o modelo de região \mathcal{M}_T desvia-se da realidade de um nodo $n \in T$ se, e somente se, $\mathcal{M}_T.\theta \neq n.\theta$ (i.e., a origem do modelo é diferente da posição do nodo). Quando isto ocorre, o nodo n constrói e anuncia um novo modelo local \mathcal{M}_n^l , baseado na sua janela de aprendizagem, e

¹<http://www.inf.ethz.ch/personal/gaertner/miniball.html>

passa a representar uma nova região de supressão. Neste caso, cada nodo que ouve este anúncio deve decidir se permanece na região de supressão atual ou se muda para a recém criada. Assim, um nodo n' só altera sua região de supressão se, e somente se, a sua distância até a origem do novo modelo de região for menor que a sua distância até a origem do seu modelo de região atual (i.e., se, e somente se, $D(n'.\theta, \mathcal{M}_n^r.\theta) < D(n'.\theta, \mathcal{M}_{n'}^r.\theta)$, onde $D(x, y)$ é a distância geográfica entre dois pontos x e y).

Como consequência, cada nodo terá a origem do modelo de região o mais próximo possível de sua localização. Na pior das hipóteses, haverá tantas regiões quanto nodos. Neste caso, o ST-DBP iguala-se ao DBP em termos de mensagens de modelo suprimidas (i.e., aplica supressão de mensagens por nó, em vez de região).

O *sink* é um nodo mais poderoso, com menos restrições de recursos que os nodos sensores. Assim, é capaz de analisar todos os modelos da região de interesse e decidir como e quando realizar uma aglutinação das regiões particionadas. Quando o *sink* detecta que uma determinada quantidade de regiões possuem modelos próximos e estáveis, calcula um novo modelo baseado nestas regiões e o anuncia como parte de uma nova região de interesse (que inclui todas as regiões analisadas). Quando os nós sensores receberem esta mensagem de interesse, deverão assumir o novo modelo.

4.2 PROJETO E IMPLEMENTAÇÃO

Apresenta-se nesta seção, os detalhes do projeto e implementação do método proposto, incluindo a sua integração à uma interface centrada a dados (SmartData) que provê uma abstração das camadas mais baixas de software, tanto em relação ao sistema operacional embarcado EPOS, quanto em relação ao protocolo de rede TSTP. Tais implementações e outras dependências de software foram integradas ao *framework* de simulação Castalia, de forma a preservar ao máximo a sua compatibilidade com o EPOS.

O ST-DBP foi implementado como um nova classe *ST_DBP*, agregando a classe *SmartData*, conforme demonstrado pela Figura 8. Ambas as classes abstraem todas as tarefas relacionadas ao mecanismo de supressão, que é transparente para a aplicação.

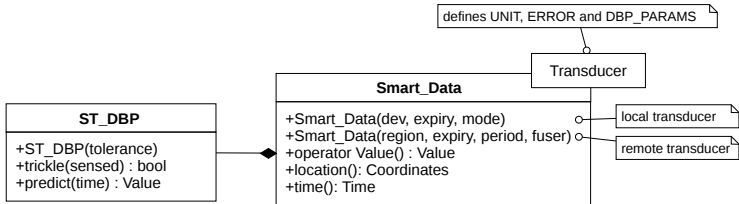


Figura 8 – Integração com SmartData.

O construtor da classe *ST_DBP* (Figura 8) possui um parâmetro (*tolerance*) que corresponde aos níveis de tolerância ao erro da predição, definidos pela aplicação. Este parâmetro é extraído da precisão requisitada na mensagem de interesse do *sink*.

Um novo parâmetro de *template*, *DBP_PARAMS*, foi definido por *Transducer*. Esta define os parâmetros do *ST_DBP* (i.e., tamanho total w e das bordas l da janela de aprendizagem). Assim, o *ST_DBP* abstrai todas as operações necessárias para manipular a janela de aprendizagem, implementada como uma fila circular, cujo tipo e tamanho são definidos em tempo de compilação por *Transducer*.

A classe *ST_DBP* implementa os métodos *trickle(sensed)* e *predict(time)*. O primeiro é usado no nodo sensor para informar um novo valor sensoreado (*sensed*) e retorna se houve desvio do modelo ou não. Já o segundo, computa o modelo de região atual para obter o valor predito \hat{V}_i em um determinado instante de tempo t_i . Este método é usado pelo *sink* em um objeto *SmartData*, que abstrai um *Transducer* remoto, para obter um dado previsto quando solicitado pela aplicação.

Para o funcionamento completo do mecanismo de supressão foram propostos e implementados três algoritmos principais. O primeiro diz respeito ao método *trickle(sensed)*, que manipula a janela de aprendizagem, checa a validade dos modelos de região e local, gera novos modelos, e é também responsável por anunciar atualizações de modelo por meio da API do TSTP. O segundo algoritmo diz respeito à atualização e sincronização dos modelos, incluindo o particionamento das regiões. Já o terceiro algoritmo trata do mecanismo de aglutinação das regiões que possuem modelos próximos.

Tais algoritmos, e outros que compõem a solução, são apresentados na seção a seguir.

4.2.1 Algoritmo de Trickle

O Algoritmo 1, denominado *ST-DBP Trickle Algorithm*, é executado por cada nodo sensor em cada período de amostragem. Inicialmente, o algoritmo insere um novo dado sensoreado (*sensed*) na janela de aprendizagem. Então é verificado se o nodo sensor já possui um modelo de região (linha 4) e, caso não possua (*i.e.*, $\mathcal{M}_i^T = null$), o nodo cria um primeiro modelo local (linha 6) e considera este modelo como sendo o modelo de sua região, somente se a janela de aprendizagem estiver cheia (linha 5). Neste caso, o nodo sensor passa automaticamente a representar a sua região.

Algoritmo 1 ST-DBP Trickle Algorithm

```

1: procedure Trickle(sensed)
2:   sendModel  $\leftarrow$  false
3:    $\mathcal{W}_i.insert(t, sensed)$ 
4:   if  $\mathcal{M}_i^T = null$  then
5:     if  $\mathcal{W}_i.size \geq w$  then
6:        $\mathcal{M}_i^T \leftarrow \mathcal{M}_i^L \leftarrow buildModel(\mathcal{W}_i)$ 
7:       sendModel  $\leftarrow$  true
8:     end if
9:   else
10:    if deviates( $\mathcal{M}_i^T, sensed, t$ ) then
11:       $\mathcal{M}_i^T \leftarrow \mathcal{M}_i^L \leftarrow buildModel(\mathcal{W}_i)$ 
12:      sendModel  $\leftarrow$  true
13:    end if
14:  end if
15:  if sendModel then
16:     $Q_i.insert(buildModelMsg(\mathcal{M}_i^L))$ 
17:  end if
18: end procedure

```

Se o nodo sensor já possui o modelo de sua região \mathcal{M}_i^T , o algoritmo avalia este modelo comparando o resultado de sua predição com o valor real sensoreado no instante t . Esta avaliação é realizada pela função *deviates* (linha 10). Esta função retorna *false* quando todos os requisitos do mecanismo de *Trickle* são satisfeitos, e *true* caso contrário. Quando os requisitos não são satisfeitos, significa que o modelo de região atual desviou-se além da tolerância definida pela aplicação, gerando e anunciando um novo modelo local. Além disso, o

nodo assume este novo modelo como sendo o modelo de uma nova região, à qual passa automaticamente a representar.

Ao final de sua execução, o algoritmo verifica se um novo modelo foi gerado, o qual é candidato a representar a região do nodo sensor. Neste caso, uma mensagem contendo o modelo é construída e inserida na fila de mensagens Q_i para que seja posteriormente transmitida pelo TSTP (linha 16).

A sincronização de um nó sensor, que possui um modelo local representando sua região, com outros nodos e o *sink* não implica em *overhead* para tal nodo. Sempre que o modelo de uma região se desvia da tolerância aceitável, o algoritmo atualiza o modelo de região e o dissemina na rede, de tal forma que nenhuma ação adicional no nodo sensor é necessária.

4.2.2 Manutenção dos Modelos

O algoritmo denominado *ST-DBP Model Update Algorithm* é executado em cada nodo sensor sempre que uma mensagem de modelo ou interesse é recebida pelo TSTP, com o objetivo de atualizar o modelo do nodo sensor ou as configurações do preditor. O processo de atualização é implementado de acordo com o Algoritmo 2, o qual lida com dois casos possíveis:

1. Caso 1: se m é uma mensagem de modelo, o algoritmo avalia o modelo recebido e sua origem (linha 7). Se a origem do modelo recebido for mais próxima que a origem do modelo atual do nodo sensor, o algoritmo atualiza o modelo.
2. Caso 2: se m é uma mensagem de interesse e possui um modelo, o algoritmo verifica se o nodo sensor está dentro da região de interesse (linha 11). Neste caso, o modelo recebido é obrigatoriamente considerado como o modelo de região do nodo sensor em questão. Este é o comportamento esperado quando o *sink* envia uma mensagem de interesse com um modelo, com o objetivo de aglutinar regiões.

Ao final do algoritmo, se houve uma atualização de modelo, o algoritmo verifica se existe alguma nova mensagem de modelo prestes a ser transmitida, neste caso a mensagem é removida da fila de transmissão do MAC e destruída. Caso o modelo tenha sido enviado pelo *sink*, o nodo sensor em questão deve, obrigatoriamente, assumir este modelo como sendo o seu modelo de região.

Algoritmo 2 ST-DBP Model Update Algorithm

```

1: procedure ModelUpdate( $m$ )
2:    $modelUpdate \leftarrow false$ 
3:    $fromSink \leftarrow false$ 
4:   if  $m.isModelMessage$  then
5:      $D \leftarrow distance(here(), m.\mathcal{M}.\theta)$ 
6:      $D_i \leftarrow distance(here(), \mathcal{M}_i^T.\theta)$ 
7:     if  $D < D_i$  or  $m.\mathcal{M}.\theta = \mathcal{M}_i^T.\theta$  then
8:        $modelUpdate \leftarrow true$ 
9:     end if
10:  else if  $m.isInterestMessage$  and  $m.\mathcal{M} \neq null$  then
11:    if  $contains(m.sphere, here())$  then
12:       $modelUpdate \leftarrow true$ 
13:       $fromSink \leftarrow true$ 
14:    end if
15:  end if
16:  if  $modelUpdate$  then
17:    if  $\exists e_Q \in Q_i : e_Q$  is a ModelMsg and  $e_Q.\mathcal{M} = \mathcal{M}_i^L$  then
18:      if  $fromSink$  or not  $deviates(m.\mathcal{M}, getSensed(), t)$  then
19:         $\mathcal{M}_i^T \leftarrow m.\mathcal{M}$ 
20:         $delete\ Q_i.remove(e_Q)$ 
21:      end if
22:    else
23:       $\mathcal{M}_i^T \leftarrow m.\mathcal{M}$ 
24:    end if
25:  end if
26: end procedure

```

4.2.3 Bootstrap

Dado um interesse do *sink* em uma determinada região do espaço e do tempo, todos os nodos sensores que respondem a este interesse medem dados de uma mesma unidade SI e consideram o mesmo período de amostragem. No *bootstrap* do ST-DBP, todos estes nodos começam a inserir medições periódicas em suas janelas de aprendizagem, até que esta esteja cheia. Uma vez cheia, o nodo sensor verifica a existência de um modelo para representar sua região, como o modelo ainda não existe, cada nodo sensor cria um modelo local e assume este como sendo o seu modelo de região, anunciando-o para outros nodos.

No entanto, todos os nodos possuem o mesmo período de amostragem e seus relógios estão sincronizados pelo TSTP, portanto existe uma possibilidade de todos os nodos executarem estas tarefas ao mesmo tempo, causando um particionamento generalizado da região já no instante inicial t_0 .

Para evitar este cenário, explora-se características *cross-layer* do TSTP, que permite buscar e remover uma mensagem da fila de transmissão do MAC antes que ela seja realmente transmitida. O TSTP-MAC verifica o canal antes de iniciar uma transmissão, e detecta uma mensagem sendo transmitida. Assim, se a mensagem for um modelo, o TSTP notifica as camadas superiores, de modo a executar o Algoritmo 2. Quando o algoritmo identifica um novo modelo (linha 16), ele verifica se há uma mensagem recente contendo seu modelo local armazenado na fila de transmissão (linha 17). Neste caso, o algoritmo verifica se o novo modelo atende aos requisitos de qualidade (i.e., tolerância aceitável), removendo a mensagem da fila de transmissão caso positivo. Caso contrário, o algoritmo mantém o modelo local como sendo o modelo de região e a mensagem não é removida da fila de transmissão.

4.2.4 Aglutinação de Regiões

Já o algoritmo de aglutinação, denominado *ST-DBP Regions Agglutination Algorithm*, é executado pelo *sink* e é responsável pela aglutinação das regiões particionadas por meio da avaliação dos modelos atuais. Este algoritmo é uma versão otimizada do algoritmo de aglutinação apresentado pelo autor em (RICHA et al., 2017), cuja solução é não escalável para uma rede com grande número de nodos.

Este algoritmo de aglutinação baseia-se na construção de um dendograma, cuja construção é norteadada por uma métrica de dissimilaridade, apresentada anteriormente na Seção 4.1.3. A Figura 9 apresenta a estrutura geral de um dendograma de dissimilaridade.

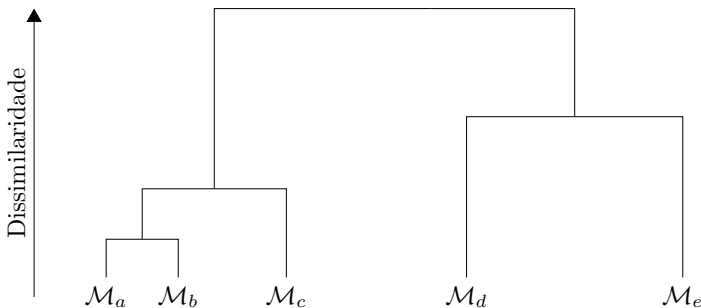


Figura 9 – Estrutura geral do dendograma de dissimilaridade.

Algoritmo 3 Dendogram Algorithm

```

1: function Dendogram
2:    $t_0 \leftarrow \text{now}()$ 
3:    $t_1 \leftarrow t_0 + (w * \mathcal{I}.\text{period})$ 
4:   // Inicializando o dendograma
5:   for each  $\mathcal{M} \in \mathcal{S}$  do
6:      $d \leftarrow \langle \mathcal{M}, \text{null}, \text{null}, 0, \text{false} \rangle$ 
7:      $dgr.\text{insert}(d)$ 
8:   end for
9:   // Gerando pares candidatos à aglutinação
10:  for  $nd_l \leftarrow dgr.\text{begin}()$  to  $dgr.\text{end}()$  do
11:    for  $nd_r \leftarrow nd_l.\text{next}()$  to  $dgr.\text{end}()$  do
12:       $\text{dissimilarity} \leftarrow \text{dissimilarity}(nd_l.\mathcal{M}, nd_r.\mathcal{M})$ 
13:      if  $\text{dissimilarity} \leq \lambda$  then
14:         $e_C \leftarrow \langle \text{null}, nd_l, nd_r, \text{dissimilarity}, \text{false} \rangle$ 
15:         $C.\text{insert}(e_C)$ 
16:      end if
17:    end for
18:  end for
19:  // Gerando o dendograma
20:  while  $|C| > 0$  do
21:     $nd_a \leftarrow C.\text{remove\_head}()$  // Menor dissimilaridade
22:    if  $nd_a.\text{left}.\text{agglutinated}$  or  $nd_a.\text{right}.\text{agglutinated}$  then
23:       $\text{delete } nd_a;$ 
24:    else
25:       $dgr.\text{remove}(nd_a.\text{left})$ 
26:       $dgr.\text{remove}(nd_a.\text{right})$ 
27:       $nd_a.\text{left}.\text{agglutinated} \leftarrow \text{true}$ 
28:       $nd_a.\text{right}.\text{agglutinated} \leftarrow \text{true}$ 
29:       $nd_a.\mathcal{M} \leftarrow \text{fuse}(nd_a.\text{left}.\mathcal{M}, nd_a.\text{right}.\mathcal{M})$ 
30:      // Verificando novos pares candidatos
31:      for each  $nd' \in dgr$  do
32:         $\text{dissimilarity} \leftarrow \text{dissimilarity}(nd'.\mathcal{M}, nd_a.\mathcal{M}, t_0, t_1)$ 
33:        if  $\text{dissimilarity} \leq \lambda$  then
34:           $e_C \leftarrow \langle \text{null}, nd', nd_a, \text{dissimilarity}, \text{false} \rangle$ 
35:           $C.\text{insert}(e_C)$ 
36:        end if
37:      end for
38:       $dgr.\text{insert}(nd_a)$ 
39:    end if
40:  end while
41:  return  $dgr$ 
42: end function

```

O dendograma de dissimilaridade, implementado como uma árvore binária, expressa em cada nodo não folha o nível de dissimilaridade entre as sub-árvores esquerda e direita. O Algoritmo 3 apresenta a construção do dendograma, partindo de uma lista \mathcal{S} de

modelos, que representam as diferentes regiões existentes. Cada nodo não folha representa um agrupamento de nodos, que correspondem às sub-árvores esquerda e direita do respectivo nodo.

Inicialmente, o algoritmo calcula o intervalo de tempo no qual a dissimilaridade dos modelos será avaliada (Linhas 2 e 3). Este intervalo corresponde ao período que inicia no instante atual (t_0) e tem a sua extensão calculada pelo tamanho (w) da janela de aprendizagem e pelo período de amostragem (*period*), definido pelo interesse da aplicação, conforme explanado na Seção 4.1.3.

Cada um dos nodos da árvore é representado por uma tupla $\langle \mathcal{M}, nd_l, nd_r, dissimilarity, agglutinated \rangle$, onde: \mathcal{M} é um modelo externo, ou o modelo fusão que representa as sub-árvores esquerda e direita; nd_l e nd_r que são, respectivamente, as sub-árvores esquerda e direita; *dissimilarity* é o nível de dissimilaridade entre as sub-árvores esquerda e direita; e *agglutinated* indica se determinado nodo já foi (*true*), ou não (*false - default*), considerado no processo de construção do dendograma (i.e., possui um nodo superior que representa a sua aglutinação).

Deste modo, o algoritmo constrói uma lista de nodos folhas não conectados (Linhas 5 a 8), cujo tamanho é equivalente ao número de modelos atuais ($|\mathcal{S}|$). Para cada modelo $\mathcal{M} \in \mathcal{S}$, o algoritmo cria uma tupla (i.e., um nodo) $\langle \mathcal{M}, null, null, 0, false \rangle$, de modo que seja um nodo folha (i.e., sub-árvores nulas), possua dissimilaridade nula, e tenha anotação de não aglutinado, para que seja processado em seguida.

Na sequência, entre as Linhas 10 e 18, o algoritmo constrói uma lista C contendo pares de modelos candidatos à aglutinação e ordenada pela dissimilaridade destes, de forma descendente (i.e., menos dissimilares antes). Esta lista terá tamanho correspondente à uma combinação simples de todos os modelos tomados dois a dois ($C_{|\mathcal{S}|,2}$), conforme a Equação 4.5.

$$C_{n,p} = \frac{n!}{p! \times (n-p)!} \quad (4.5)$$

Fixando o número de elementos a serem agrupados ($p = 2$), a Equação 4.5 por ser reescrita conforme a Equação 4.6. Sendo n o número de elementos do conjunto de entrada, temos que a complexidade do algoritmo é quadrática com o número de modelos ($O(|\mathcal{S}|^2)$).

$$C_{n,2} = \frac{n!}{2 \times (n-2)!} = \frac{n \times (n-1) \times (n-2)!}{2 \times (n-2)!} = \frac{n \times (n-1)}{2} \quad (4.6)$$

Existe uma melhoria significativa em relação à versão preliminar proposta em (RICHA et al., 2017), que parte da construção do conjunto potência ($2^{\mathcal{S}}$) do conjunto de modelos \mathcal{S} , também denominado conjunto das partes do conjunto \mathcal{S} . Por conseguinte, tal algoritmo apresenta uma complexidade exponencial com o número de modelos ($O(2^{|\mathcal{S}|})$).

Apesar da melhoria proporcionada, o algoritmo ainda apresenta uma certa limitação frente à necessidade de escalar a solução. Esta limitação pode ser controlada pela limitação do volume da esfera de interesse e do raio de correlação, dependendo da densidade de nodos na rede, reduzindo o número de pares candidatos à aglutinação.

Após construir a lista C de pares candidatos, o algoritmo remove iterativamente o primeiro elemento de C (Linha 21), até que a lista C esteja vazia. Este primeiro elemento corresponde ao nodo candidato de menor dissimilaridade (i.e., maior similaridade entre sub-árvores). Assim, a cada iteração, as sub-árvores mais próximas são aglutinadas e passam a constituir um novo nodo, cujo modelo é a fusão dos modelos das sub-árvores (Linha 29). Este modelo é então combinado com os modelos que já constituem o dendograma para formar novos candidatos que são incluídos em C (Linhas 31 a 37). Após, o novo nodo aglutinado é inserido no dendograma (Linha 38)

Ao final da execução, o algoritmo retorna o dendograma de dissimilaridade de modelos, representado por uma árvore binária. A raiz de tal árvore será entrada para o algoritmo de aglutinação (Algoritmo 4), que a utilizará para determinar as operações de aglutinação a serem realizadas.

O algoritmo de aglutinação possui uma abordagem recursiva na qual, a cada recursão, avalia a existência de sub-árvores para aglutinação. Enquanto não encontrar (i.e., enquanto $coords = null$), o algoritmo avança recursivamente pelos diferentes ramos da árvore. Se $root$ não for um nodo folha (sub-árvores não nulas), e a dissimilaridade das suas sub-árvores for menor ou igual ao *threshold* (λ) (Linha 5), então as sub-árvores de $root$ podem ser aglutinadas. Neste caso, uma lista vazia de coordenadas é criada ($coords$: Linha 6), na qual serão inseridas as coordenadas de origem dos modelos aglutinados.

Uma vez que $coords \neq null$, o algoritmo inicia o avanço recursivo até os nodos folhas das sub-árvores de $root$ (Linha 7), que possuem a referência para os modelos aglutinados. Deste modo, para cada nodo folha encontrado, o algoritmo insere as coordenadas do modelo aglutinado na lista $coords$ (Linha 14).

Algoritmo 4 ST-DBP Regions Agglutination Algorithm

```

1: procedure RegionsAgglutination(root, coords = null)
2:   if root = null then
3:     return
4:   end if
5:   if coords = null and (root.dissimilarity ≤ λ and ¬ root.isLeaf) then
6:     coords' ← new CoordinatesList()
7:     RegionsAgglutination(root, coords')
8:     sphere ← Miniball(coords')
9:     Qi.insert(buildInterestMsg(root.M, sphere))
10:  else
11:    RegionsAgglutination(root.left, coords)
12:    RegionsAgglutination(root.right, coords)
13:    if coords and root.isLeaf then
14:      coords.insert(root.M)
15:    end if
16:  end if
17: end procedure

```

Após percorrer as sub-árvores e inserir na lista *coords* as coordenadas de todos os modelos aglutinados, é gerada uma esfera que contém todas estas coordenadas (Linha 8). A construção desta esfera corresponde ao problema SEB, apresentado na Seção 4.1.4. Para tanto, foi utilizada a implementação em C disponibilizada por Gärtner (1999), citada na mesma seção. Esta implementação foi adaptada e integrada à solução proposta.

Por fim, na Linha 9 é gerada uma mensagem de interesse contendo o novo modelo de aglutinação (*root.M*), cujo destino é a região do espaço endereçada pela esfera gerada (*sphere*). Esta mensagem é então inserida na fila de mensagens a serem transmitidas pelo TSTP (Linha 9).

4.2.5 Roteamento Geográfico

O algoritmo de roteamento, denominado *Greedy Forwarding Algorithm* (RESNER et al., 2016), foi proposto para evitar retransmissões de mensagens para destinos incorretos. Neste algoritmo as transmissões sempre possuem um progresso positivo em direção ao destino. Este comportamento pode introduzir problemas no ST-DBP, uma vez que um modelo gerado em qualquer ponto da região de interesse e enviado ao *sink* pode não ser recebido por um ou mais nodos que estejam na direção oposta ao *sink*.

A solução proposta para este problema consiste em um processo de *flooding* geograficamente controlado, no qual as propagações de mensagens são regidas de acordo com diferentes condições. Neste sentido, foi proposta uma extensão do protocolo original para tratar tais condições, de modo a possibilitar o funcionamento adequado dos mecanismos adotados pelo ST-DBP.

O mecanismo de roteamento é implementado de acordo com o Algoritmo 5, o qual assume por padrão que a mensagem m não será retransmitida (Linha 2). Ainda assim, toda mensagem recebida passa por uma lista circular de mensagens vistas recentemente, que é mantida e consultada por este algoritmo (Linhas 3 a 7), o que possibilita o controle e finalização do *flooding*, explicado posteriormente.

Naturalmente, toda mensagem possui um destino que está fortemente relacionado ao seu tipo. Para o ST-DBP, as mensagens de modelo são destinadas, não somente ao *sink*, mas também a todos os nodos inseridos na região do respectivo interesse e que respondem ao mesmo. Neste caso, o algoritmo utiliza-se do artifício de alternar entre estes destinos para garantir o roteamento adequado destas mensagens. Se m é uma mensagem de modelo (Linha 9) e o nodo atual responde ao interesse correspondente à m (Linhas 10 e 11), então o algoritmo assume que o destino de m corresponde à esfera vinculada ao respectivo interesse. Em qualquer outro caso o algoritmo assume o destino natural na mensagem (Linha 8).

Uma vez definido o destino, o algoritmo trata de dois casos, sendo o primeiro sub-dividido em dois. O primeiro, quando o nodo atual está inserido na região de destino, e o segundo caso corresponde ao caso geral, no qual o nodo está localizado entre a origem e o destino e, portanto, é um possível retransmissor da mensagem.

1. O nodo atual está contido na região destino (Linhas 15 a 34): neste caso o algoritmo utiliza uma abordagem na qual o progresso da mensagem é calculado no sentido radial da esfera (região destino), tanto em direção ao centro (*forward*), quanto em direção às bordas (*backward*), seja este progresso promovido pela sua própria transmissão ou pela de outro nodo (recepção).
 - 1.1. Nodo do último *hop* está contido na região destino: neste caso ambos os nodos, atual e anterior, estão participando do *flooding* e por isso é feito um controle de terminação baseado no número de vezes que o nodo detectou o progresso da mensagem. Na recepção de uma mensagem, o algoritmo acumula progresso em direção ao centro, se o lastHop está

mais próximo do centro, e acumula progresso em direção à borda no caso contrário. O mesmo é realizado em relação à retransmissão, quando o nodo irá retransmitir a mensagem em questão. Deste modo, o nodo só retransmite a mensagem se ainda não detectou um progresso em cada direção.

- 1.2. Nodo do último *hop* não está contido na região destino: neste caso o nodo anterior (*lastHop*) não está participando do *flooding* e por isso irá retransmitir a mensagem até que perceba um *ack* implícito (i.e., aquele que é observado quando um nodo mais próximo do destino transmite a mensagem). Assim, o nodo que está dentro da esfera só retransmite, não só com o objetivo de fazer progresso em direção ao destino, mas também para notificar o *sender*, se, e somente se, a origem da mensagem não está contida na região destino.

2. O nodo atual não está contido na região destino (Linhas 35 a 39): neste caso o critério de roteamento é igual aquele definido pelo algoritmo original, no qual um nodo retransmite a mensagem sempre que está mais próximo do destino do que o último *hop*.

Além da decisão de roteamento realizada pelo algoritmo, o TSTP aplica um distorção espacial, que consiste em um *offset*, responsável por determinar quais nodos provavelmente terão acesso primeiro ao canal de comunicação. Uma das métricas para definição deste *offset* é a distância geográfica do nodo atual até o destino final, fazendo com que nodos mais próximos tenham prioridade no acesso ao canal para transmissão de uma determinada mensagem.

Neste sentido, foi proposta a utilização de um coeficiente de distorção híbrido, cuja definição é dependente da relação entre o nodo e o interesse que corresponde à mensagem que está sendo transmitida. Este coeficiente de distorção é definido em termos da distância do nodo em relação as bordas da esfera de interesse, ou em relação à sua distância do destino. Deste modo, é possível promover um bom espalhamento das mensagens nas regiões, fazendo com que a propagação das mensagens se dê de forma mais eficiente no contexto do ST-DBP.

Algoritmo 5 Roteamento Baseado em Progresso Radial

```

1: procedure Radial_Greedy_Forward(m)
2:   toRelay  $\leftarrow$  false
3:   eS  $\leftarrow$  S.search(m.id) // Lista de mensagens vistas recentemente
4:   if eS = null then
5:     eS  $\leftarrow$  new S.El(0,0)
6:     S.insert(eS)
7:   end if
8:   dst  $\leftarrow$  m.dst
9:   if m.isModelMessage then
10:    eI  $\leftarrow$  I.search(m) // Lista de interesses sendo respondidos pelo nodo
11:    if eI = null then
12:      dst  $\leftarrow$  eI.sphere
13:    end if
14:  end if
15:  if contains(dst, here()) then
16:    if contains(dst, m.LastHop) then
17:      if distance(here(), dst) > distance(m.LastHop, dst) then
18:        eS.forward++
19:        if eS.backward < 1 then
20:          toRelay  $\leftarrow$  true
21:          eS.backward++
22:        end if
23:      end if
24:      if distance(here(), dst) < distance(m.LastHop, dst) then
25:        eS.backward++
26:        if eS.forward < 1 then
27:          toRelay  $\leftarrow$  true
28:          eS.forward++
29:        end if
30:      end if
31:    else
32:      toRelay  $\leftarrow$  not contains(dst, m.Origin)
33:      m.ACK  $\leftarrow$  toRelay
34:    end if
35:  else
36:    if distance(here(), dst) < distance(m.LastHop, dst) then
37:      toRelay  $\leftarrow$  true
38:    end if
39:  end if
40:  eQ  $\leftarrow$  Q.search(m.id)
41:  if eQ = null and toRelay then
42:    if distance(here(), dst) < distance(m.LastHop, dst) then
43:      m.progressBits  $\leftarrow$  true
44:    end if
45:    Q.insert(m)
46:  else
47:    delete m
48:  end if
49: end procedure

```

5 AVALIAÇÃO

Este capítulo apresenta a avaliação do mecanismo de supressão de mensagens em RSSF, apresentado no Capítulo 4 como uma extensão da técnica proposta por Raza et al. (2015), para considerar a correlação espacial da quantidade física sensoreada. Para tanto, a Seção 5.1 apresenta as métricas adotadas para avaliação e a Seção 5.2 discute os resultados obtidos na avaliação.

5.1 MÉTRICAS AVALIADAS

Para conduzir a avaliação da técnica proposta, algumas métricas de avaliação foram adotadas. Uma das métricas utilizadas é a taxa de supressão (SR, do inglês *Suppression Ratio*) (RAZA et al., 2015), definida pela Equação 5.1, que é o complemento da razão entre a quantidade de mensagens geradas com supressão pela quantidade sem supressão, variando de 0 à 1. Quanto maior o seu valor, mais eficiente é a técnica.

$$SR = 1 - \frac{\text{mensagens geradas com supressão}}{\text{mensagens geradas sem supressão}} \quad (5.1)$$

Foi considerada também uma métrica de erro entre os valores observados pela aplicação (i.e., valores preditos no *sink*) e o valor real medido pelo nodo. Adotou-se, assim como Raza et al. (2015), a Raiz do Erro Quadrático Médio (RMSE, do inglês *Root Mean Squared Error*), definida pela Equação 5.2, como métrica de qualidade da série temporal predita no *sink*. Esta métrica foi utilizada apenas na avaliação do estudo de caso 2.

$$RMSE = \sqrt{\frac{1}{k} \sum_{i=1}^k (V_i - \hat{V}_i)^2} \quad (5.2)$$

Onde V_i e \hat{V}_i são, respectivamente, os i -ésimos valores das séries temporais real e predita, e k corresponde ao total de valores reais medidos pelo nodo sensor. Outras três métricas foram utilizadas em conjunto para inferir a efetividade do mecanismo de particionamento e aglutinação adotados pelo método proposto. Sendo elas o número de particionamentos e aglutinações, e o número de regiões particionadas ao longo do tempo.

5.2 RESULTADOS

Nesta seção são apresentados e discutidos os resultados obtidos por meio de simulação. A solução foi implementada¹, integrada e avaliada, por meio de simulação, utilizando o Castalia Framework (BOULIS, 2017), versão 3.3, um *framework* de simulação de RSSF desenvolvido sobre o simulador de eventos discretos OMNET++ (OPENSIM, 2017) versão 4.6.

5.2.1 Estudo de Caso 1

Uma avaliação preliminar foi executada, comparando a proposta com a versão original da técnica. Esta avaliação foi conduzida em nível de aplicação, em um ambiente que simula a comunicação entre os nodos, e todas as propriedades do mecanismo descrito anteriormente. A avaliação usou dados reais de temperatura, coletados durante dez dias, por dez sensores em uma fazenda solar. O período de amostragem foi de um minuto, resultando em um número total de 14.400 amostras.

Os parâmetros utilizados para a avaliação, sumarizados na Tabela 2 foram os mesmos utilizados em (RAZA et al., 2015) para as tolerâncias de valor \mathcal{E}_V e tempo \mathcal{E}_T , além do tamanho total w e das bordas l da janela de aprendizagem. A única exceção é o erro relativo ϵ^{rel} , para o qual foi definido um intervalo de valores. Os parâmetros ϕ e ψ são, respectivamente, o número mínimo de regiões a serem aglutinadas e a porção do erro máximo aceitável pela aplicação, que corresponde ao coeficiente máximo de dissimilaridade para o qual dois modelos podem ser considerados suficientemente similares.

Tabela 2 – Parâmetros usados durante a avaliação.

ϵ^{rel}	ϵ^{abs}	\mathcal{E}_T	w	l	ϕ	ψ
(0 à 10 a cada 0,5)%	0,5°C	2	6	3	2	0,2

Foram comparadas duas versões da proposta com o DBP original. A primeira não implementa o mecanismo de aglutinação. Portanto, só usa a proximidade geográfica dos modelos anunciados. O mecanismo de aglutinação é avaliado na segunda versão, onde é possível observar o ganho relacionado ao mecanismo de aglutinação.

¹A implementação está disponível no repositório do Laboratório de Integração de Software e Hardware (LISHA), em: <https://epos.lisha.ufsc.br/svn>.

A Figura 10 apresenta os resultados obtidos referentes à taxa de supressão. O método apresentou resultados significativamente melhores em comparação ao DBP original. Com um ϵ^{rel} baixo de 1,5%, o ST-DBP apresentou uma diferença de 3,17%, totalizando 97,95%, o que é um ganho significativo, uma vez que a taxa de supressão do DBP já é alta. Com ϵ^{rel} de 10% o ST-DBP completo apresentou a maior taxa de supressão, de 99,7%. Quando comparado ao trabalho anterior (RAZA et al., 2015), com ϵ^{rel} de 5%, o ST-DBP atingiu uma taxa de supressão de 99,32% contra uma razão de 90,18% da versão original. A diferença entre a versão do ST-DBP sem o mecanismo de aglutinação e a versão com, evidenciando a contribuição do mecanismo de particionamento e aglutinação.

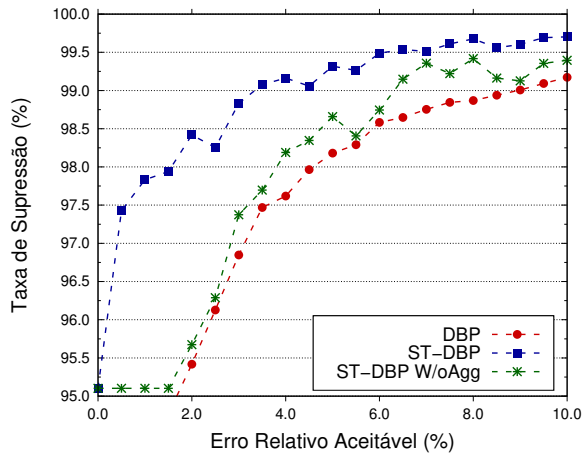


Figura 10 – Taxa de supressão dos parâmetros avaliados no DBP, ST-DBP W/oAgg, e ST-DBP Full.

Os resultados da avaliação para dez dias, com ϵ^{rel} de 1,5%, sumarizados na Tabela 3. Com o ST-DBP completo, nesta configuração, ocorreram 2838 atualizações de modelos, 428 particionamentos de regiões, e 130 aglutinações. O tempo máximo de supressão de mensagens atingiu um intervalo de 5 horas e 20 minutos, contra um máximo de 2 horas e 47 minutos da versão original. Em média, o ST-DBP aumentou os intervalos de supressão por 0,92 vezes, quando comparado versão original, o que indica menor frequência média de atualização de modelos

Tabela 3 – Resumo das simulações com ϵ^{rel} de 1,5%.

Métricas	DBP	ST-DBP	
		W/oAgg	Full
Média de regiões particionadas	–	9,17	3
Aglutinações / 2 h	–	–	1,08
Atualizações de modelo / 5 min	2,62	2,45	0,98
Partições / 30 min	NA	0,02	0,89
Máximo intervalo de supressão	10020 s	19200 s	19200 s
Intervalo médio de supressão	413,79 s	430,28 s	639,29 s

Uma explicação para esse ganho promovido pelo ST-DBP é a sua principal característica, de agrupar os nós sensores em regiões de supressão. O gráfico de frequência apresentado na Figura 11 apresenta uma comparação da distribuição do número de regiões particionadas ao longo da simulação.

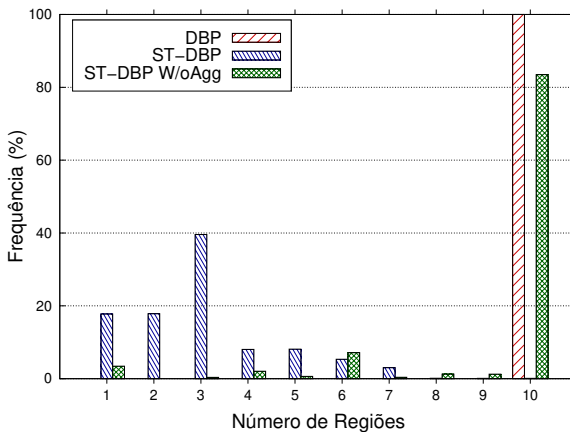


Figura 11 – Distribuição do número de regiões particionadas, com ϵ^{rel} de 1,5%.

Para essa comparação, o número de regiões particionadas com o DBP foi considerada máxima, pois há um modelo para cada nodo sensor. Com ST-DBP sem aglutinação a região de interesse alcançou o máximo de particionamento após 39,66 horas, o que corresponde a 83,47% do tempo de avaliação. ST-DBP completo permaneceu 75,18% do tempo com um número de regiões menor ou igual a três, 24,52% entre quatro e sete regiões e menos de 0,3% acima de sete regiões.

5.2.2 Estudo de Caso 2

Além da avaliação preliminar previamente realizada em (RICHA et al., 2017), apresentada na seção anterior, uma avaliação da técnica foi realizada por meio de simulação, utilizando uma rede de grande escala, densa, e com uma frequência de amostragem significativa, com o intuito de investigar quanta energia o mecanismo realmente economiza. Para tanto, foi utilizado o conjunto de dados produzido pelo Lausanne Urban Canopy Experiment (LUCE), mantido pelo projeto Sensorscope (NADEAU et al., 2009).

No projeto, executado entre outubro de 2006 e abril de 2007 no campus universitário da École Polytechnique Fédérale de Lausanne (EPFL), uma rede com 110 nodos foi implantada em uma área que corresponde à aproximadamente 120 mil metros quadrados. Objetivando realizar o monitoramento meteorológico do campus, diferentes grandezas físicas foram observadas, tais como a temperatura ambiente, irradiação solar, umidade relativa do ar, volume de precipitação, dentre outras.

Por questões práticas, e para limitação da abrangência deste trabalho, as simulações consideraram um intervalo de tempo equivalente à 24 horas, que correspondem ao dia 07 de fevereiro de 2017. Duas séries temporais foram consideradas, sendo uma delas de temperatura com 88 nodos sensores e outra de irradiação solar, composta por 18 dos 88 nodos, todos operantes no intervalo citado, além de um nodo *sink*.

Os dados disponibilizados pelo projeto Sensorscope apresentam um período de amostragem de trinta segundos. Para sua utilização nas simulações foi implementado um componente com a mesma API do transdutor do EPOS no simulador, com o intuito de preservar ao máximo a compatibilidade da codificação dos componentes entre o simulador e o sistema operacional EPOS.

No *setup* de simulação, os parâmetros de configuração de rádio aplicados foram os do CC2538, o qual é um SoC da Texas Instruments, projetado para redes de sensores sem fio de baixo consumo energético, utilizado no EPOSMoteIII (LISHA, 2017b). Além disso, a quantidade de energia inicial dos nodos da rede foi configurada em 18720 joules, equivalente à energia disponível em duas baterias AA.

O projeto Sensorscope disponibilizou, junto aos dados, as coordenadas dos nodos sensores, que foram convertidas e utilizadas para o posicionamento dos mesmos na topologia da rede, ilustrado na Figura 12. Por simplicidade, os nodos são identificados em dois tipos,

sendo o tipo A os nodos que medem apenas temperatura, e o tipo B os nodos que medem, além da temperatura, a irradiação solar. O *sink* foi posicionado aproximadamente no centro da área monitorada.

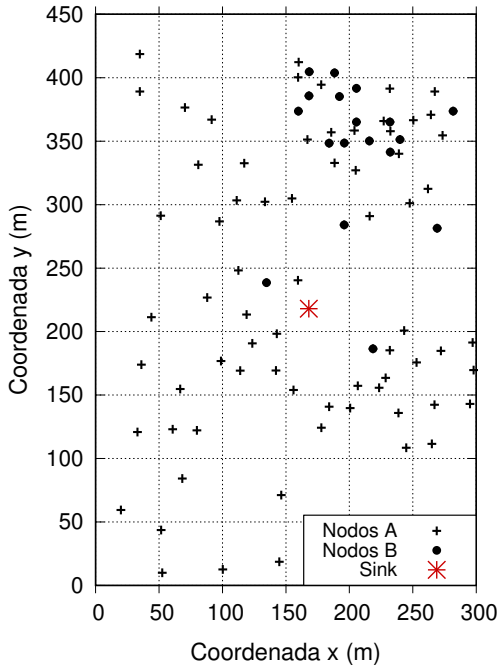


Figura 12 – Topologia da rede implantada pelo projeto Sensorscope.

Os parâmetros utilizados para esta avaliação, sumarizados na Tabela 4, são os mesmos utilizados na avaliação descrita na seção anterior, com exceção dos parâmetros w e ϕ . Durante avaliações preliminares foi constatado que construir um modelo baseado em $w = 10$ leituras resulta em um modelo mais adequado para este cenário. Em relação ao parâmetro $\phi = 5$, percebeu-se que em uma rede de maior densidade, como a apresentada, considerar valores menores pode resultar em mais aglutinações ineficazes, uma vez que existem muitos nodos próximos.

Tabela 4 – Parâmetros usados durante a avaliação.

Série	ϵ^{rel}	ϵ^{abs}	\mathcal{E}_T	w	l	ϕ	ψ
T	(0 à 10 a cada 0,5)%	0,5 °C	2	10	3	5	0,2
I	(0 à 10 a cada 0,5)%	0,5 W/m ²	2	10	3	5	0,2

T - Temperatura I - Irradiação Solar

Iniciou-se as avaliações pela série de temperatura. A Figura 13 apresenta a taxa de supressão de mensagens proporcionada pela técnica aplicada. É possível observar que a técnica proposta obteve um ganho máximo de aproximadamente 2,7% em taxa de supressão, com erro relativo de 2%, quando comparada à técnica original. Quando analisada a taxa de supressão em relação aos diferentes níveis de erro relativo aceitável, ambas as técnicas apresentaram um comportamento semelhante, não sendo observado nenhum distanciamento significativo. Além disso, a técnica proposta apresenta, com intervalo de confiança de 95%, um erro não nulo da média, ao contrário da técnica original, cuja taxa de supressão depende unicamente da série temporal.

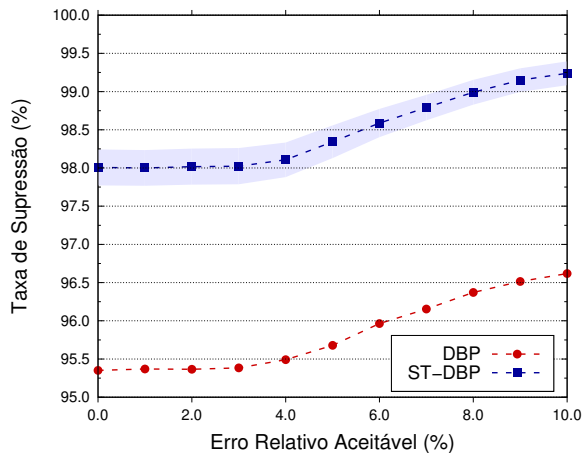


Figura 13 – Taxa de supressão para a série de temperatura.

O aumento da taxa de supressão está relacionado ao aumento do erro relativo aceitável, o que pode ser observado para erro relativo maior que 3%. Tal comportamento é esperado, uma vez que ao admitir um erro relativo maior, é considerado um intervalo de

tolerância maior, o que implicará no aumento do tempo de validade do modelo e, conseqüentemente, da taxa de supressão.

Em relação a qualidade das predições realizadas pelos modelos de supressão, espera-se que a técnica proposta apresente uma perda de qualidade, quando comparada às predições observadas com a técnica original. Isto ocorre devido ao mecanismo de aglutinação, já que vários nodos utilizam o mesmo modelo de supressão, mesmo que este não tenha sido gerado com base em seus próprios dados, ocasionando assim, um maior desvio das predições. Este comportamento pode ser observado pela análise da métrica Raiz do Erro Quadrático Médio (RMSE), apresentada na Figura 14.

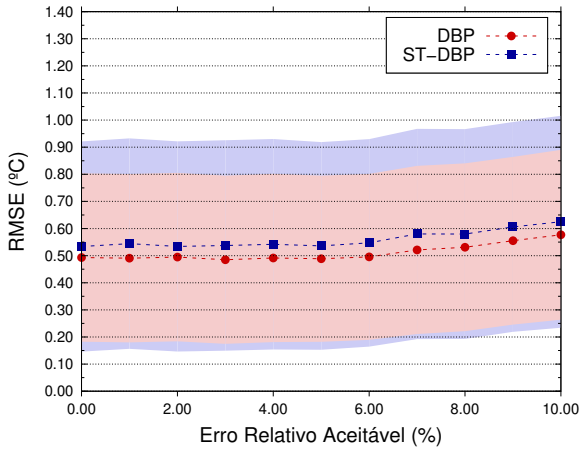


Figura 14 – Raiz do erro quadrático médio.

Além disso, considerando um intervalo de confiança de 95%, é possível observar que a variação da média para a técnica proposta, é maior do que a da técnica original. Isto também ocorre devido à utilização de modelos construídos por nodos vizinhos ou pelo *sink*.

A Figura 15 relaciona o erro relativo aceitável com a métrica de tempo de vida da rede. Apesar de observado um ganho de aproximadamente 2,7% na taxa de supressão, o mesmo não é observado em relação à estimativa de tempo de vida da rede. Tal comportamento é justificado pelo mecanismo de aglutinação, que apesar de objetivar a supressão de mensagens, incorre no envio de uma nova mensagem de interesse, do *sink* em direção à uma dada região do espaço, cujos nodos serão aglutinados.

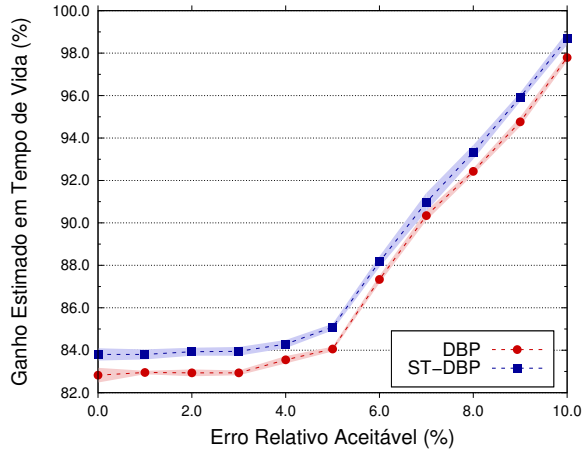


Figura 15 – Ganho estimado em tempo de vida da rede para a série de temperatura.

A Figura 16 retrata a distribuição do número de regiões particionadas (i.e., quantidade de modelos), a cada período de amostragem no nodo sensor, ou predição no *sink*, em um histograma cujo eixo ordenado é apresentado em escala logarítmica de base dois, para facilitar a sua visualização. A técnica proposta manteve o número de regiões particionadas entre 37 e 78 regiões, ao contrário da técnica original que, assim como para o estudo anterior, considerou-se a rede totalmente particionada (i.e., um modelo por nodo).

Nestas análises, é observado um ganho de até 2,7% em supressão de mensagens, que é considerado significativo, uma vez que a taxa de supressão já apresentava-se alta, acima de 95%. Por outro lado, apresentou um ganho menos significativo em tempo de vida da rede, mas superou a técnica original em até 1,4%. Em relação à qualidade dos dados observados pela aplicação (i.e., dados preditos), os resultados são estatisticamente equivalentes e se mantiveram, na média, dentro dos requisitos especificados.

Com o objetivo de avaliar a utilização do mecanismo proposto em uma série temporal, cujas leituras apresentam uma variação menos estável, uma rede com sensores de irradiação solar foi considerada, conforme citado anteriormente. A Figura 17 apresenta a taxa de supressão obtida, onde é possível observar inicialmente um ganho inferior ao obtido quando considerada a série de temperatura (Figura 13).

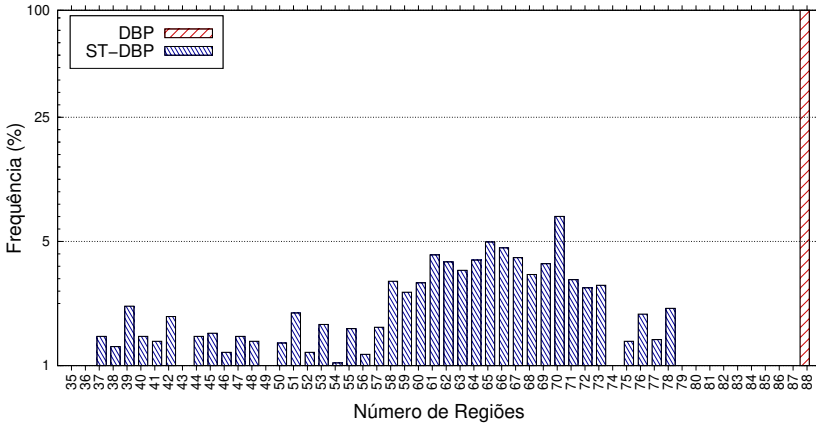


Figura 16 – Distribuição do número de regiões particionadas para a série de temperatura, com ϵ^{rel} de 5%.

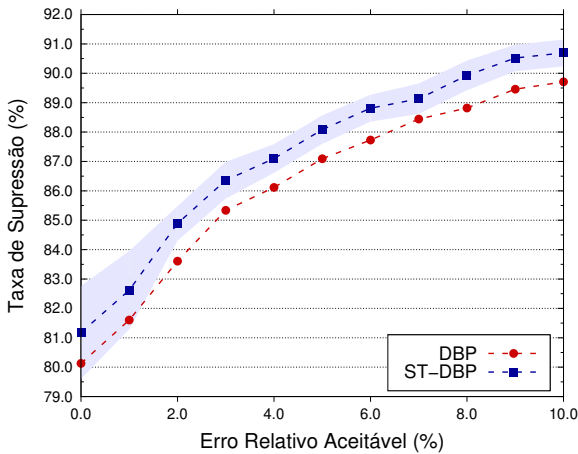


Figura 17 – Taxa de supressão para a série de irradiação solar.

Além disso, considerando um intervalo de confiança de 95%, o erro da média apresentou-se maior com erro relativo menor que 2%, sendo estes os casos em que a técnica proposta não apresentou-se melhor do que a técnica original em todas repetições das simulações. Isto se deve ao fato de que, com baixo erro relativo, o mecanismo de supressão torna-se muito sensível às variações que ocorrem no espaço.

O ganho estimado em tempo de vida está ilustrado na Figura 18. Assim como na análise anterior, o ganho em tempo de vida apresenta-se menor em relação ao ganho obtido com a supressão. Com a série de irradiação solar, a técnica proposta não apresentou ganhos significativos em relação à técnica original. Considerando um intervalo de confiança de 95%, pode-se considerar que ambas as técnicas apresentam um mesmo ganho em tempo de vida da rede.

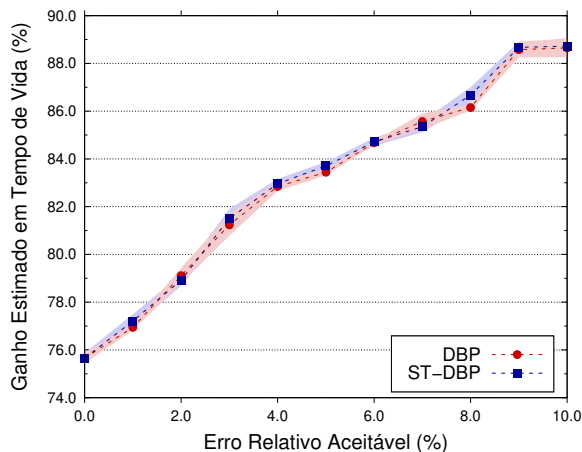


Figura 18 – Ganho estimado em tempo de vida da rede para a série de irradiação solar.

A distribuição do número de regiões particionadas para a série de irradiação solar, considerando o erro relativo de 5% está apresentada no histograma da Figura 19, em escala logarítmica de base dois. Novamente, a rede foi considerada totalmente particionada quando submetida à técnica original. Ao aplicar a técnica proposta, a rede apresentou-se particionada entre 11 e 17 regiões, sendo que 15 regiões apresentou a maior ocorrência, que corresponde à 47,52% do tempo de simulação.

Em análise a este gráfico é possível observar que para a série de irradiação solar, a técnica mostrou-se menos eficiente, uma vez que apresentou uma redução pouco significativa no número de regiões. Esta limitação ocorre, principalmente, devido à escala da rede, mas também devido às variações observadas entre nodos distintos em um mesmo instante do tempo, o que demonstra uma variabilidade mais significativa da grandeza física no espaço.

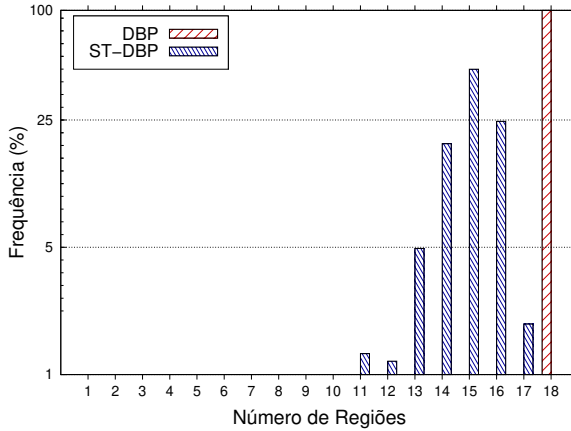


Figura 19 – Distribuição do número de regiões particionadas para a série de irradiação solar, com ϵ^{rel} de 5%.

A partir destas análises, observa-se que a técnica proposta, apresentou um maior ganho quando aplicada à série de temperatura, com uma taxa de supressão de até 2,7% e 1,4% em ganho em tempo de vida. Por outro lado, quando aplicada à série de irradiação solar, a técnica apresentou um ganho menos significativo de até 1,2% em taxa de supressão e pôde ser considerada equivalente à técnica original em termos do ganho em tempo de vida, o que indica que o cenário não é adequado para aplicação da técnica proposta, principalmente devido à escala e densidade da rede de monitoramento de irradiação solar.

6 CONSIDERAÇÕES FINAIS

Técnicas de supressão de mensagens baseadas em modelos de tendência simplificados, recentemente propostas para predição de séries temporais, não exploram a variação espacial da grandeza física sensoreada, o que poderia atingir níveis de supressão de mensagens ainda maiores. Assim, esta dissertação propôs a extensão da técnica de supressão de mensagens DBP, baseada em modelos de tendência, para explorar a baixa variabilidade espacial destas grandezas.

A técnica proposta, denominada Space-Time Derivative-based Prediction (ST-DBP), representa dados de um conjunto de sensores em uma determinada região do espaço com um único modelo de predição. O cerne da proposta consiste em um mecanismo de particionamento e aglutinação de regiões, denominadas neste trabalho de regiões de supressão, que compartilham um mesmo modelo de predição enquanto aglutinadas. O particionamento ocorre sempre que um nó pertencente a uma região de supressão detecta um desvio do modelo de região, e a aglutinação ocorre quando o *sink* detecta a possibilidade de aglutinação de regiões cujos dados apresentam-se similares no período em questão.

Para isto, foi proposta uma alteração no modelo de supressão, a qual consiste na inclusão da coordenada de origem do modelo (i.e., posição geográfica do nó que o gerou). Para o mecanismo de aglutinação, foram adotadas algumas soluções como: uma métrica de dissimilaridade, para determinação das regiões aptas a serem aglutinadas; e um algoritmo para solução do problema da Bola de Fechamento Mínima, para determinar a região de aglutinação que contém um dado conjunto de modelos similares.

Como parte integrante da proposta, foi apresentado um conjunto de algoritmos que compõem a técnica. Estes algoritmos englobam a criação e manutenção dos modelos, além do mecanismo de particionamento e aglutinação propriamente ditos. Também foi proposto um ajuste no algoritmo de roteamento, de modo a habilitá-lo a rotear mensagens destinadas às regiões de forma a satisfazer os requisitos da técnica proposta.

Foi apresentado o detalhamento do projeto e implementação da solução, incluindo a sua integração ao SmartData, uma interface centrada a dados que provê uma abstração das camadas mais baixas de software, tanto em relação ao sistema operacional embarcado EPOS, quanto em relação ao protocolo de rede TSTP. Tais

implementações e outras dependências de software foram integradas ao framework de simulação Castalia, de forma a preservar ao máximo a compatibilidade da codificação com o sistema operacional embarcado EPOS.

Uma avaliação foi conduzida por meio de simulação, utilizando a topologia e os dados obtidos por uma rede real de larga escala e densidade de nodos, tendo como principais métricas a taxa de supressão e o ganho em tempo de vida da rede. Nesta análise, considerando uma série de temperatura, a técnica proposta apresentou um ganho significativo de 2,7% na taxa de supressão, visto que as taxas de supressão já apresentavam-se acima de 95%. Um ganho menos significativo foi obtido em relação ao tempo de vida da rede, que superou em até 1,4% à técnica original. Por outro lado, em relação a qualidade dos dados, as técnicas apresentaram-se estatisticamente equivalentes, e mantiveram-se dentro dos requisitos especificados pela aplicação. A técnica proposta também foi avaliada em uma rede monitorando a irradiação solar, e apresentou um ganho de até 1,2% em taxa de supressão e pode ser considerada equivalente à técnica original em termos do ganho em tempo de vida.

Considera-se que os objetivos deste trabalho foram atingidos, apresentando como principal contribuição, a proposta e análise de um mecanismo de supressão de mensagens em redes de sensores sem fio, explorando a baixa variação espacial das grandezas físicas sensoreadas. Além disso, os resultados demonstraram a aplicabilidade da técnica para tal finalidade em RSSF.

Por fim, identifica-se como trabalhos futuros a melhoria da métrica de dissimilaridade, com o intuito de tornar a decisão de aglutinação mais acertada. Além disso, com esta melhoria, é possível melhorar a qualidade dos dados, fator que não apresentou nenhuma melhoria neste trabalho. Também a integração da proposta com um protocolo MultiSink, com o objetivo de analisar o comportamento desta proposta em um cenário mais complexo e com maior demanda de comunicação. Além disso, como apresentado na seção de avaliação, modelos mais sofisticados podem ser necessários para séries temporais cujos perfis no espaço apresentam padrões mais complexos. Neste caso, ficam apontados como trabalhos futuros a busca por estes modelos, uma vez que este aspecto esteve fora do escopo deste trabalho.

REFERÊNCIAS

- AKYILDIZ, I. F.; SU, W.; SANKARASUBRAMANIAM, Y.; CAYIRCI, E. A survey on sensor networks. **IEEE Communications Magazine**, v. 40, n. 8, p. 102–114, aug 2002. ISSN 0163-6804.
- ANASTASI, G.; CONTI, M.; FRANCESCO, M. D.; PASSARELLA, A. Energy conservation in wireless sensor networks: A survey. **Ad Hoc Networks**, v. 7, n. 3, p. 537–568, 2009. ISSN 1570-8705.
- BARTON, T.; MUSILEK, P. Derivative based prediction with look ahead. In: **2016 International Joint Conference on Neural Networks (IJCNN)**. Vancouver, Canada: IEEE, 2016. p. 2118–2123.
- BOULIS, A. **Castalia A simulator for Wireless Sensor Networks and Body Area Networks**. 2017. Disponível em: <<https://github.com/boulis/Castalia>>.
- FISCHER, K.; GÄRTNER, B.; KUTZ, M. Fast smallest-enclosing-ball computation in high dimensions. In: SPRINGER. **European Symposium on Algorithms**. Budapest, Hungria, 2003. p. 630–641.
- FRÖHLICH, A. A.; OKAZAKI, A. M.; STEINER, R. V.; OLIVEIRA, P.; MARTINA, J. E. A Cross-layer Approach to Trustfulness in the Internet of Things. In: **9th Workshop on Software Technologies for Embedded and Ubiquitous Systems (SEUS)**. Paderborn, Germany: IEEE, 2013. p. 1–8.
- FRÖHLICH, A. A. SmartData: an IoT-Ready API for Sensor Networks. **International Journal of Sensor Networks**, v. 28, n. 3, p. 202–210, 2018. ISSN 1748-1279.
- GÄRTNER, B. Fast and robust smallest enclosing balls. In: SPRINGER. **European Symposium on Algorithms**. Praga, República Tcheca, 1999. p. 325–338.
- GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. 1. ed. Rio Grande do Sul, Brasil: Editora da UFRGS, 2009.
- IEEE 1451.0. **IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data**

Sheet (TEDS) Formats. New York, NY, EUA, September 2007. 335 p.

ISTOMIN, T.; MURPHY, A. L.; PICCO, G. P.; RAZA, U. Data Prediction + Synchronous Transmissions = Ultra-low Power Wireless Sensor Networks. In: **Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM.** New York, NY, USA: ACM, 2016. (SenSys '16), p. 83–95. ISBN 978-1-4503-4263-6.

KHAN, J. A.; QURESHI, H. K.; IQBAL, A. Energy management in Wireless Sensor Networks: A survey. **Computers & Electrical Engineering**, v. 41, p. 159–176, 2015. ISSN 0045-7906.

KHAN, W. Z.; SAAD, N. M.; AALSALEM, M. Y. An overview of evaluation metrics for routing protocols in wireless sensor networks. In: **2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012).** Kuala Lumpur, Malásia: IEEE, 2012. v. 2, p. 588–593.

LISHA. **EPOS - Embedded Parallel Operating System.** 2017. Disponível em: <<https://epos.lisha.ufsc.br/>>.

LISHA. **EPOSMote III.** 2017. Disponível em: <<https://epos.lisha.ufsc.br/EPOSMote+III>>.

LIU, Z.; XING, W.; ZENG, B.; WANG, Y.; LU, D. Distributed Spatial Correlation-based Clustering for Approximate Data Collection in WSNs. In: **2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA).** Barcelona, Spain: IEEE, 2013. p. 56–63. ISSN 1550-445X.

MARTINETZ, T.; MAMLOUK, A. M.; MOTA, C. Fast and easy computation of approximate smallest enclosing balls. In: **2006 19th Brazilian Symposium on Computer Graphics and Image Processing.** Manaus, Amazonas, Brasil: IEEE, 2006. p. 163–170. ISSN 1530-1834.

NADEAU, D. F. et al. Estimation of urban sensible heat flux using a dense wireless network of observations. **Environmental fluid mechanics**, Springer, v. 9, n. 6, p. 635–653, 2009.

OPENSIM. **OMNeT++ - Objective Modular Network Testbed in C++.** 2017. Disponível em: <<https://omnetpp.org/>>.

POTDAR, V.; SHARIF, A.; CHANG, E. Wireless sensor networks: A survey. In: **2009 International Conference on Advanced Information Networking and Applications Workshops**. Bradford, Reino Unido: IEEE, 2009. p. 636–641.

RAZA, U.; CAMERRA, A.; MURPHY, A. L.; PALPANAS, T.; PICCO, G. P. What does model-driven data acquisition really achieve in wireless sensor networks? In: **2012 IEEE International Conference on Pervasive Computing and Communications**. Lugano, Suíça: IEEE, 2012. p. 85–94.

RAZA, U.; CAMERRA, A.; MURPHY, A. L.; PALPANAS, T.; PICCO, G. P. Practical Data Prediction for Real-World Wireless Sensor Networks. **IEEE Transactions on Knowledge and Data Engineering**, v. 27, n. 8, p. 2231–2244, aug 2015. ISSN 1041-4347.

RESNER, D. **Performance Evaluation of the Trustful Space-Time Protocol**. 192 p. Dissertação (Mestrado) — Federal University of Santa Catarina, Florianópolis, 2018. M.Sc. Thesis.

RESNER, D.; ARAUJO, G. M. de; FRÖHLICH, A. A. On the Impact of Dynamic Routing Metrics on a Geographic Protocol for WSNs. In: **Brazilian Symposium on Computing Systems Engineering**. João Pessoa, Brazil: IEEE, 2016.

RESNER, D.; FRÖHLICH, A. A. Design Rationale of a Cross-layer, Trustful Space-Time Protocol for Wireless Sensor Networks. In: **20th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)**. Luxembourg, Luxembourg: IEEE, 2015. p. 1–8.

RICHA, C. H.; GRACIOLI, G.; ; FROHLICH, A. A. Space-Time Derivative-Based Prediction: A Novel Trickling Mechanism for WSN. In: **Brazilian Symposium on Computing Systems Engineering**. Curitiba, Brazil: IEEE, 2017. ISBN 978-1-5386-3590-2.

RUIZ, L. B.; NOGUEIRA, J. M.; LOUREIRO, A. A. F. Manna: a management architecture for wireless sensor networks. **IEEE Communications Magazine**, v. 41, n. 2, p. 116–125, Feb 2003. ISSN 0163-6804.

TULONE, D.; MADDEN, S. An Energy-efficient Querying Framework in Sensor Networks for Detecting Node Similarities. In: **Proceedings of the 9th ACM International Symposium on**

Modeling Analysis and Simulation of Wireless and Mobile Systems. New York, NY, USA: ACM, 2006. (MSWiM '06), p. 191–300. ISBN 1-59593-477-4.

TULONE, D.; MADDEN, S. PAQ: Time Series Forecasting for Approximate Query Answering in Sensor Networks. In: RÖMER, K.; KARL, H.; MATTERN, F. (Ed.). **Wireless Sensor Networks.** Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 21–37. ISBN 978-3-540-32159-0.

WELZL, E. Smallest enclosing disks (balls and ellipsoids). In: **New results and new trends in computer science.** Graz, Austria: Springer, 1991. p. 359–370.

XU, X.; ZHANG, G. A Hybrid Model for Data Prediction in Real-World Wireless Sensor Networks. **IEEE Communications Letters**, PP, n. 99, p. 1, 2017. ISSN 1089-7798.