



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Computação Aplicada

**POSICIONAMENTO ÓTIMO DE MÚLTIPLOS SINKS
MÓVEIS EM REDES DE SENSORES SEM FIO: UMA
PROPOSTA PARA CIDADES INTELIGENTES**

Emerson Santos de Oliveira

Feira de Santana

2018



Universidade Estadual de Feira de Santana
Programa de Pós-Graduação em Computação Aplicada

Emerson Santos de Oliveira

**POSICIONAMENTO ÓTIMO DE MÚLTIPLOS SINKS
MÓVEIS EM REDES DE SENSORES SEM FIO: UMA
PROPOSTA PARA CIDADES INTELIGENTES**

Dissertação apresentada à Universidade
Estadual de Feira de Santana como parte
dos requisitos para a obtenção do título de
Mestre em Computação Aplicada.

Orientador: Daniel G. Costa

Feira de Santana

2018

Ficha Catalográfica - Biblioteca Central Julieta Carteado

O46p Oliveira, Emerson Santos de
Posicionamento ótimo de múltiplos sinks móveis em redes de sensores sem fio : uma proposta para cidades inteligentes / Emerson Santos de Oliveira. - Feira de Santana, 2018.
75f.: il.

Orientador: Daniel G. Costa.
Dissertação (mestrado) - Universidade Estadual de Feira de Santana, Programa de Pós-Graduação em Computação Aplicada, 2018.

1. Rede de Sensores sem Fio (RSSF). 2. Sinks móveis. 3. Redes sem fio - Otimização. I. Costa, Daniel G., orient. II. Universidade Estadual de Feira de Santana. III. Título.

CDU: 004.73

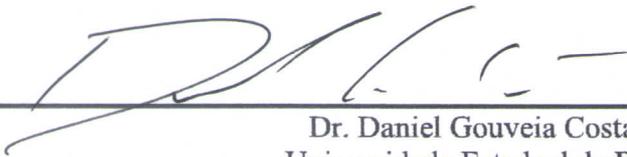
Emerson Santos de Oliveira

**POSICIONAMENTO ÓTIMO DE MÚLTIPLOS SINKS MÓVEIS
EM REDES DE SENSORES SEM FIO: UMA PROPOSTA PARA
CIDADES INTELIGENTES**

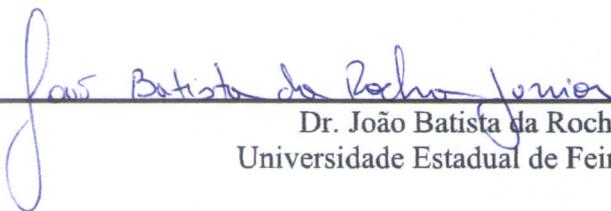
Dissertação apresentada à Universidade Estadual de Feira de Santana como parte dos requisitos para a obtenção do título de Mestre em Computação Aplicada.

Feira de Santana, 04 de maio de 2018

BANCA EXAMINADORA



Dr. Daniel Gouveia Costa (Orientador)
Universidade Estadual de Feira de Santana



Dr. João Batista da Rocha Júnior
Universidade Estadual de Feira de Santana



Dr. Ricardo Araújo Rios
Universidade Federal da Bahia

Abstract

The development of new technologies in the last years in the areas of wireless communications has brought many advantages to the development of many sectors of the city. Recent advances in sensor technology are providing an expansion of the possible applications, creating an environment of increasingly efficient data communication. With this, we can see a great advance in the scientific researches related to the development of techniques and improvement of communication in Wireless Sensor Networks (WSN). In fact, this type of network has been applied in several areas, such as: smart cities, internet of things (IoT), environmental monitoring, Vehicle tracking, health applications, among others. In this context, this work proposes to develop an algorithm for positioning node sinks in urban areas for intelligent cities, in order to optimize the communication between mobile sink nodes and fixed sensors clustered in a WSN. This approach allows the optimization of WSN for the urban environment, providing an improvement in the rates of energy consumption, processing and transmission of data in the network.

Keywords: Wireless Sensor Network, Positioning, Optimization, Smart Cities

Resumo

O desenvolvimento de novas tecnologias ocorrido nos últimos anos nas áreas das comunicações sem fio tem trazido grandes benefícios para o desenvolvimento de diversos setores da cidade. Avanços recentes na tecnologia de sensores estão proporcionando uma expansão das aplicações possíveis, criando um ambiente de comunicação de dados cada vez mais eficiente. Com isso, nota-se um grande avanço nas pesquisas científicas relacionadas ao desenvolvimento de técnicas e aprimoramento da comunicação em Redes de Sensores Sem Fio (RSSF). Com efeito, esse tipo de rede passou a ser aplicada em diversas áreas, tais como: cidades inteligentes, internet das coisas (IoT), monitoramento ambiental, rastreamento de veículos, aplicações nas áreas de saúde, entre outros. Nesse contexto, este trabalho propõe desenvolver um algoritmo de posicionamento de nós *sinks* em zonas urbanas para cidades inteligentes, com o propósito de otimizar a comunicação entre os nós *sinks* móveis e sensores fixos agrupados em *cluster* em uma RSSF. Essa abordagem, permite a otimização da RSSF para o ambiente urbano, proporcionando uma melhoria nas taxas de consumo de energia, processamento e transmissão de dados na rede.

Palavras-chave: Rede de Sensores sem Fio, Posicionamento, Otimização, Cidades Inteligentes

Prefácio

Esta dissertação de mestrado foi submetida à Universidade Estadual de Feira de Santana (UEFS) como requisito parcial para obtenção do grau de Mestre em Computação Aplicada.

A dissertação foi desenvolvida dentro do Programa de Pós-Graduação em Computação Aplicada (PGCA), tendo como orientador o Dr. Daniel G. Costa.

Agradecimentos

Agradeço a Deus por ter me dado forças para terminar mais essa etapa da minha vida. À minha esposa, filhos, família e a todos os meus amigos. Agradeço, aquelas pessoas especiais que diretamente me incentivaram, que me deram força para eu terminar esta jornada. Em especial aos amigos, João Paulo Just Peixoto, Ricardo de Jesus e Luiz Gustavo de Jesus Araújo, pelo apoio e incentivo durante o curso. Aos professores, funcionários e colegas do PGCA- Pós-Graduação em Computação Aplicada, por ter dividido a alegria desta experiência. E ao meu orientador Daniel G. Costa, por ter feito um excelente trabalho me auxiliando na pesquisa, desenvolvimento e principalmente, pela confiança e carinho de sempre. A todos, meu muito obrigado!

Sumário

Abstract	i
Resumo	ii
Prefácio	iii
Agradecimentos	iv
Sumário	vi
Lista de Tabelas	vii
Lista de Figuras	viii
Lista de Abreviações	ix
1 Introdução	1
1.1 Objetivos	4
1.2 Principais Contribuições	4
1.3 Organização do Trabalho	5
2 Fundamentação Teórica	7
2.1 Redes de Sensores sem Fio	7
2.1.1 Aplicações	8
2.1.2 Mobilidade em Redes de Sensores	9
2.1.3 Roteamento em RSSF	11
2.1.4 Protocolos de Roteamento	12
2.1.5 Comunicação por Clusters em RSSF	14
2.2 Cidades Inteligentes	15
2.2.1 Construindo Cidades Inteligentes: Exemplos	16
2.2.2 Tecnologias Envolvidas	16
2.3 Problema do Caminho Mais Curto	17
2.3.1 Algoritmos Especializados	18
2.3.2 Aplicações em Cidades Inteligentes	22
2.4 Relevância e Priorização	23

2.4.1	QoS (Quality of Service)	24
2.4.2	QoE (Quality of Experience)	24
3	Metodologia de Pesquisa	26
3.1	Escopo da Solução	26
3.2	Formalização do Problema	28
3.3	Descrição das Simulações	30
3.4	Etapas da Metodologia de Pesquisa	31
4	Solução Proposta	32
4.1	Conceitos Fundamentais	32
4.1.1	Definições Básicas	34
4.1.2	Abordagem Proposta	38
4.1.3	Algoritmo Proposto	41
4.1.4	Posicionamento dos <i>sinks</i>	43
5	Resultados Experimentais	45
5.1	Preparação dos cenários de testes	45
5.2	Cenário 1: Comportamento dos <i>sinks</i> em vias congestionadas	46
5.3	Cenário 2: Análise com dados da Av. Getúlio Vargas	52
5.4	Cenário 3: Perda de PDUs com muitos sensores	54
5.5	Análise dos Resultados	56
6	Considerações Finais	58
	Referências Bibliográficas	60

Lista de Tabelas

2.1	Algoritmos de caminho mais curto.	22
5.1	Situação do congestionamento.	47
5.2	Programação de atividades dos sensores.	48
5.3	Comparação entre rotas.	49
5.4	Tabela da velocidade média na Av. Getúlio Vargas.	52
5.5	Situação do congestionamento na Getúlio Vargas.	52
5.6	Comparação entre a simulação de rotas.	53

Lista de Figuras

1.1	Múltiplos sinks em Cidade Inteligente.	4
2.1	Exemplo dos principais componentes de uma RSSF.	9
2.2	Exemplo de otimização do <i>sink</i>	13
3.1	Exemplo de uma Cidade Inteligente com sensores para monitoramento.	27
3.2	MobSink - Simulador para múltiplos <i>sinks</i>	31
4.1	Modelo proposto.	33
4.2	Fluxograma do processo de obtenção de rotas.	34
4.3	Modelo de um grafo com posicionamento do nó <i>sink</i>	36
4.4	Exemplo do resultado do algoritmo (MSMA).	42
4.5	Fluxograma do algoritmo proposto.	43
4.6	Posicionamento do <i>sink</i>	44
5.1	Mapa da região simulada: cidade de Feira de Santana.	46
5.2	Sensores posicionados no mapa.	47
5.3	PDU's recebidos pelo <i>sink</i> por hora do dia.	49
5.4	Perda de PDU's por hora do dia.	50
5.5	PDU's recebidos por três <i>sinks</i>	51
5.6	Perda de PDU's por hora do dia com três <i>sinks</i>	51
5.7	PDU's perdidos por dia.	53
5.8	Mapa com 1000 sensores.	54
5.9	Perda de PDU's usando um <i>sink</i>	55
5.10	Perda de PDU's usando três <i>sinks</i>	55

Lista de Abreviações

Abreviação	Descrição
RSSF	Rede de Sensores Sem Fio
IoT	Internet of Things
QoS	Quality of Service
QoE	Quality of Experience
CH	Cluster Head
TIC	Tecnologia da Informação e Comunicação
MBS	Estação Base Móvel
MDC	Coletores de Dados Móveis
CSMA	Carrier Sense Multiple Access
ROT	Route Optimization
GPS	Sistema de Posicionamento Global
RL	Relevance Level
VANET	Veicular Ad-Hoc Network
PDU	Packet Data Units
MSMA	Algoritmo para Movimentação de Múltiplos <i>Sinks</i>

Capítulo 1

Introdução

Em virtude da evolução ocorrida nos últimos anos nas áreas das comunicações sem fio, circuitos eletrônicos e sensores, houve uma intensificação da pesquisa científica relacionada ao desenvolvimento de técnicas e aprimoramento da comunicação em Redes de Sensores Sem Fio (RSSF). Com efeito, esse tipo de rede passou a ser aplicada em diversas áreas, tais como: cidades inteligentes, internet das coisas (IoT), monitoramento ambiental, rastreamento de veículos, aplicações na área de saúde, entre outros.

Avanços recentes na tecnologia de sensores estão proporcionando uma expansão das possíveis aplicações das RSSF, criando um ambiente de comunicação de dados cada vez mais eficiente. Por meio de ferramentas de Tecnologia da Informação e Comunicação (TICs) os processos e infraestrutura de serviços, informações e gestão urbana têm a oportunidade de se tornar cada vez mais "inteligentes" e eficazes. E quanto mais eficiente forem as RSSF, os dispositivos que permeiam as margens dessa rede passam a se tornar cada vez mais ativos com os novos conceitos de IoT e Cidades Inteligentes.

Uma RSSF é formada por pequenos dispositivos eletrônicos alimentados por bateria que formam uma arquitetura *ad hoc* (comunicação entre os nós fonte e nó destino se dá através de múltiplos saltos). Esta arquitetura contribui para a redução do consumo de energia na transmissão de dados evitando que todos os nós da rede precisem transmitir informações diretamente ao nó *sink* que representa um nó coletor de dados da RSSF. Assim, os sensores transmitem apenas aos seus sensores vizinhos. No entanto, as restrições de consumo de energia, processamento e transmissão acabam criando muitas questões a serem investigadas [Peixoto e Costa 2015b].

De forma geral, as RSSF podem ser vistas como um tipo específico de rede móvel. Numa rede sem fio tradicional, a comunicação entre os dispositivos computacionais é feita através de estações de rádio conhecidas como "ponto de acesso", que constituem uma infraestrutura de comunicação específica. Já nas RSSF, os elementos computacionais trocam dados diretamente entre si, salto a salto. Do ponto de vista organizacional, as RSSF possuem elementos computacionais que comunicam diretamente entre si através de enlaces de comunicação sem fio, onde cada elemento provê dados, que são processados por nós

específicos chamados de *sinks*. Um dos principais objetivos de uma RSSF é produzir, durante o máximo de tempo possível, informações importantes a partir de dados coletados na rede [Peixoto e Costa 2015b]. Esse objetivo deve ser alcançado em um contexto que prolongue ao máximo o tempo de vida útil de um nó sensor, que terá como consequência o aumento da disponibilidade da RSSF, fazendo com que a rede continue a fornecer informações por mais tempo. Desta forma, as RSSF refletem um ambiente de desafios, referente ao desenvolvimento de padrões e protocolos capazes de minimizar as barreiras hoje existentes.

As redes de sensores são compostas por diversos nós sensores. De acordo com [Akyildiz et al. 2007], o posicionamento dos sensores pode ser aleatório ou "manipulado", permitindo assim a implantação aleatória em terrenos inalcançáveis, como em regiões vulcânicas ou seguindo um plano de implantação. Para isso os protocolos de rede de sensores e algoritmos devem ser capazes de se auto organizar para cada tipo de aplicação e topologia de rede. Uma outra característica é a capacidade de processar e transmitir apenas os dados necessários e parcialmente processados. Desta forma, pode-se economizar recursos essenciais para a vida útil do sensor. De fato, eficiência energética é um ponto crucial para a otimização de RSSF.

As redes tradicionais de sensores sem fio podem ser compostas por um grande número de nós sensores e *sinks* estáticos espalhados pelo ambiente. Os nós sensores geram relatórios sobre diversos eventos e encaminham para um *sink*. No entanto, este modelo de nós estáticos pode restringir a vida útil da rede, levando-se em consideração que os nós que compõem a rede de sensores têm limitações computacionais, de armazenamento e de capacidades de comunicação [Akyildiz et al. 2007]. O nó *sink* é responsável por controlar a rede e repassar tanto os dados coletados, quanto informações para os nós que estão no grupo de sensores [Akyildiz et al. 2007]. Em uma outra abordagem [Oh et al. 2010] pode-se desejar que o nó *sink* deixe de ser um elemento fixo em um local e passe a ter a liberdade de se mover pela RSSF, de movimentar-se aleatoriamente pela RSSF e, com isso, ampliar as aplicações desta.

Um *sink* móvel tem dois tipos de mobilidade entre os agrupamentos de sensores estáticos na rede. Um tipo é a mobilidade do *sink* dentro dos grupos de sensores "*intra-cluster*" [Oh et al. 2010], ou seja, o nó se move dentro de um grupo de sensores, reduzindo a sobrecarga de um nó que estiver realizando um evento com prioridade alta. Já em um outro modelo, a movimentação e transferência de dados é feita entre grupos de sensores. Neste caso, o nó *sink* movimentar-se entre os *cluster* de origem e de destino usando um algoritmo de roteamento, sendo definida como "*inter-cluster*" [Oh et al. 2010].

De maneira diferente para [Silva et al. 2014] existem três classes de mobilidade do *sink* diferenciada pelo autor como estação base móvel (MBS), coletores de dados móveis (MDC) e soluções baseadas em "Rendezvous", ou seja, soluções híbridas, que é uma mistura das duas primeiras classes. Nas estações de base móvel, o *sink* está apto a se mover por toda a rede, aumentando desta forma a cobertura e diminuindo o número de saltos para alcançar cada nó. Já os coletores de dados móveis aproveitam a capacidade de nós mais potentes para realizar coleta sob demanda, evitando que os dados coletados

trafeguem através de vários saltos, consumindo recursos importantes da rede. Usar um *sink* móvel para coletar informações de uma RSSF contribui para a economia de energia e perda de pacotes nos nós sensores. Porém, em redes de grande escala com um grande número de nós sensores, a implantação de múltiplos *sinks* é essencial para aumentar a capacidade de gerenciamento da rede e reduzir a dissipação de energia em cada nó [Oyman e Ersoy 2004]. Portanto, no trabalho de [Oyman e Ersoy 2004] o autor propõe ainda que durante a fase de projeto de uma rede de sensores de grande escala, deve ser estabelecido o número de *clusters* e os locais ótimos dos nós *sinks*. Esta metodologia é definida pelo autor como "rede de sensores com múltiplo *sinks*". Entretanto, este método apresenta alguns problemas clássicos como de posicionamento e movimentação múltipla dos *sinks* espalhados em um ambiente urbano.

Este presente trabalho, propõe desenvolver um algoritmo para otimizar o posicionamento de múltiplos *sink* móveis em uma rede de sensores sem fio, visando a eficiência energética da rede e redução nos níveis de perda de pacotes transmitidos para o nó *sink* na rede, com base em um ambiente urbano de uma Cidade Inteligente. Para esta otimização serão levados em conta parâmetros de priorização de nós fontes com o propósito de mover os *sinks* para regiões que possuam sensores mais relevantes para a comunicação da rede em determinado momento.

Em qualquer aplicação de RSSF, a conectividade entre os sensores é considerada uma importante métrica para medir a qualidade da comunicação [Akyildiz et al. 2007]. Sendo assim, a organização dos sensores em uma RSSF, o posicionamento, métodos de priorização e a mobilidade de múltiplos *sinks* é de extrema importância para uma maior vida útil da rede, motivando o desenvolvimento de algoritmos que minimizem estas variáveis, aumentando o tempo de vida do sensor e, com isso, mantendo por mais tempo a rede ativa. Portanto, o algoritmo deverá calcular rotas para os *sinks* móveis, quando houver um evento urbano (monitorado pelos sensores), que indique a mudança de relevância e definir o posicionamento final dos *sinks*. Entretanto, como o *sink* se posicionará em locais diferentes em uma área urbana, serão levados em consideração parâmetros como:

1. Dados sobre o comportamento do trânsito em ambientes urbanos;
2. Prioridade dos nós sensores espalhados no ambiente urbano;
3. O número de *sinks* móveis necessários para a coleta e transmissão dos dados;
4. Parâmetros de configuração das vias urbanas.

Assim, este trabalho apresenta uma forma otimizada para a comunicação entre nós *sinks* móveis e nós sensores organizados em grupos, encontrando o melhor caminho e posicionamento dos nós *sinks* móveis, levando em conta técnicas de roteamento e restrições de uma RSSF com o foco em Cidades Inteligentes, como representada na Figura 1.1.

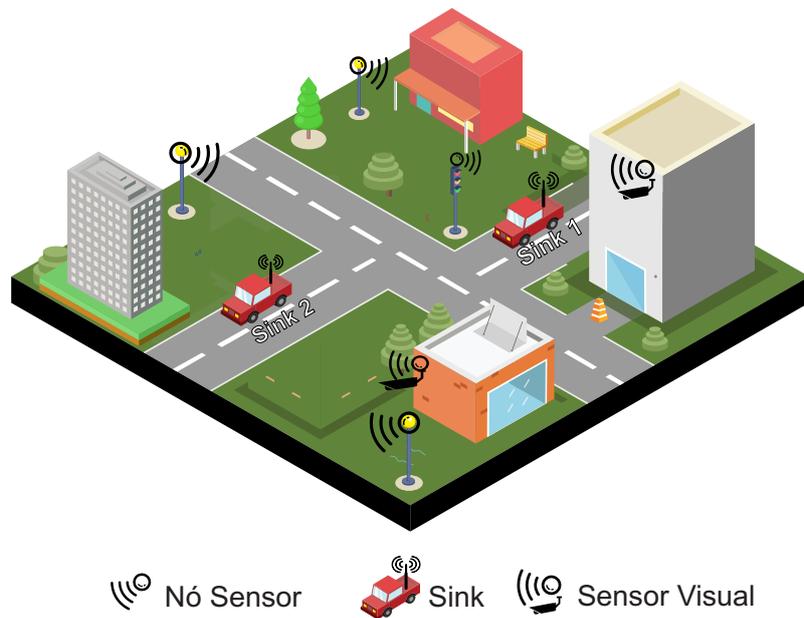


Figura 1.1: Múltiplos sinks em Cidade Inteligente.

1.1 Objetivos

Esta pesquisa tem como objetivo geral desenvolver um algoritmo de posicionamento de nós *sinks* móveis em zonas urbanas para Cidades Inteligentes, com a finalidade de otimizar a comunicação entre *sinks* móveis e sensores em uma RSSF, quando houver um evento urbano (monitorado pelos sensores), que indique a mudança de relevância dos grupos de sensores. Para isso, foram estabelecidos os seguintes objetivos específicos:

- Definir o posicionamento final dos *sinks* móveis em uma RSSF com base em um ambiente urbano de uma Cidade Inteligente;
- Calcular rotas para os *sinks* móveis instalados em veículos urbanos;
- Reduzir a perda de pacotes na RSSF em uma Cidade Inteligente, quando houver um evento urbano.

1.2 Principais Contribuições

Nesta dissertação foram estudadas diversas tecnologias que tem objetivos semelhantes ao abordado neste trabalho. Dentre eles, pode-se verificar que a perspectiva considerada apresenta modelos de posicionamento dos nós *sinks* que adotam diferentes paradigmas para aplicações em uma RSSF. Como resultado do trabalho apresentado nesta dissertação, as seguintes contribuições podem ser destacadas.

[Peixoto e Costa 2017a] descrevem uma abordagem para RSSF com o objetivo de posicionar múltiplos *sinks* móveis em um cenário de Cidade Inteligente. A proposta descrita pelos autores considera a relevância de cada nó sensor na rede e as restrições de movimentação do nó *sink* em uma cidade para apresentar soluções de mobilidade dos *sink* móveis ao longo das vias. Assim, foram testados dois cenários, um de configuração conhecida e outro de configuração aleatória para obter o máximo de realidade em relação a um ambiente urbano.

[Costa et al. 2014] apresentam um trabalho que visa definir diferentes parâmetros de priorização como a relevância de monitorização, detecção de eventos críticos, alvos de visualização, codificação multimídia, entre outros. E relatam que os níveis de priorização são de extrema importância na implantação de redes de sensores sem fio no mundo real.

[Peixoto e Costa 2015b] propõe uma abordagem de mobilidade para otimizar o posicionamento dos *sinks* móveis, considerando as prioridades dos nós sensores. Dependendo da aplicação e características, a prioridade de cada sensor pode ser considerada a fim de determinar as posições dos *sinks* móveis, reduzindo o número de saltos dos nós com prioridade alta. Assim, apresentaram novos métodos de posicionamento dos múltiplos *sinks* móveis com QoE, visando a redução do consumo de energia no sensor sem fio, reduzindo a distância dos nós de alta prioridade.

Já em [Peixoto 2015] foi apresentada, uma nova abordagem para o posicionamento de múltiplos *sinks* em uma RSSF, o que também levou ao desenvolvimento de um método de inicialização para o algoritmo de setorização chamado de k-means. Os algoritmos desenvolvidos fazem uso dos níveis de relevância de cada sensor, onde os nós mais importantes devem ser priorizados, mantendo-os em funcionamento pelo maior tempo possível.

Em [Souza et al. 2016] é apresentado o SPARTAN, uma solução que notifica os condutores sobre áreas congestionadas através de comunicação veículo a veículo e emprega um mecanismo de tomada de decisão em tempo real utilizado para reencaminhar veículos e evitar as áreas congestionadas. Reduzindo assim o tempo da viagem e o tempo de congestionamento, evitando que os veículos passem por áreas congestionadas, proporcionando, conseqüentemente um bom fluxo de tráfego em cenários de Cidades Inteligentes.

Portanto, essas contribuições trouxeram várias perspectivas para o desenvolvimento deste trabalho. Vale ressaltar, que os autores produziram valiosas abordagens sobre métodos de priorização, setorização e movimentação dos *sink* móveis aplicados a RSSF. Desta forma, estes conceitos contribuíram com esta pesquisa, trazendo ideias de como aplicá-los em um ambiente urbano de uma Cidade Inteligente.

1.3 Organização do Trabalho

Esse texto está organizado da seguinte forma: O Capítulo 2 descreve a fundamentação teórica do trabalho, abordando conceitos referente as RSSF e Cidades Inteligentes, já no Capítulo 3, é apresentada a metodologia do trabalho. O Capítulo 4, trata da solução

proposta e apresenta modelos matemáticos para a resolução do problema, enquanto no Capítulo 5 será apresentado os resultados da pesquisa, Por último, o Capítulo 6 apresenta as conclusões finais do trabalho.

Capítulo 2

Fundamentação Teórica

Vários trabalhos foram desenvolvidos nos últimos anos com o intuito de investigar diversas questões relacionadas a topologia, meios de transmissão, processamento, armazenamento de dados e mobilidade dos nós sensores instalados em uma RSSF. Essas abordagens procuram desenvolver as melhores soluções para problemas de eficiência energética e desempenho do tráfego da informações, com o objetivo de aumentar a vida útil das redes de sensores. Nas sessões seguintes serão apresentados trabalhos relacionados à Rede de Sensores sem Fio, Cidades Inteligentes, algoritmos especializados, relevância e priorização.

2.1 Redes de Sensores sem Fio

Uma Rede de Sensores Sem Fio (RSSF) é formada por vários sensores sem fio transmitindo dados de fenômenos diversos através de transmissores de rádio, com uma fonte de alimentação autônoma [Yick et al. 2008]. Um atributo muito importante deste tipo de rede é a rápida implantação e a capacidade de se auto-organizar, principalmente pelo fato de utilizar protocolos tolerantes a falhas. Em [Yick et al. 2008] foi relatado que atualmente as RSSF proporcionam um debate abrangente sobre a implantação e a aplicação deste tipo de rede em diversas áreas, devido as variadas restrições de hardware que limita recursos essenciais como: energia das baterias, alcance do sinal, largura de banda, processamento e armazenamento de dados nos nós sensores.

Além disso, esses sensores podem desenvolver várias medições e monitoramento em diversos tipos de ambientes, e são compostas por uma variedade de dispositivos instalados de forma aleatória, que podem medir variados tipos de fenômenos como: luz, temperatura, umidade, pressão barométrica, velocidade, aceleração, etc. Alguns dispositivos podem ser selecionados para ter a função de *gateway* de sensores que tem a atribuição de receber e retransmitir dados para estações de coleta de informações, esses dispositivos possui um maior poder de processamento, armazenamento de energia e potência de transmissão do rádio [Yick et al. 2008]. Sendo assim, as redes de sensores são formadas pelo

conjunto de vários nós sensores que tem a função de coletar e transmitir dados para os *gateway* [Yick et al. 2008]. Entretanto, para realizar estas funções, [Akyildiz et al. 2007] relata que o hardware básico de um sensor é composto dos seguintes componentes: uma unidade de sensoriamento, uma unidade de processamento, um emissor e receptor de rádio e de potência. Esses dispositivos também podem conter outros componentes adicionais que depende da aplicação da rede, tais como um sistema de localização via satélite, um gerador de energia e um mobilizador pode ser necessário para mover os nós sensores para a realização das tarefas. Os nós sensores sem fio são dispositivos eletrônicos equipados com uma fonte de energia limitada [Yick et al. 2008]. Em alguns cenários [Akyildiz et al. 2007] a reposição de recursos pode ser impossível, tendo assim, um tempo de vida útil a ser monitorado [Akyildiz et al. 2007].

Para [Baronti et al. 2007] sensores possuem dois tipos de transmissão via rádio. No primeiro caso, pode-se usar um modelo mais simples de sensor que opera usando o protocolo CSMA (*Carrier Sense Multiple Access*) cuja a transmissão é feita em uma faixa que possui licença livre variando entre (315/433/868/916 MHz) com largura de banda entre 20-50 kbps. No segundo caso, modelos mais atuais de sensores que fazem parte do padrão IEEE 802.15.4, operando na faixa de 2,4 GHz, com largura de banda de 250 kbps. Segundo [Baronti et al. 2007], estes modelos oferecem a possibilidade de uso de antena interna ou externa para aumentar alcance de transmissão na rede. Já o alcance nos dois modelos pode variar de acordo com o tipo de frequência usada no rádio, características físicas e ambientais relacionadas à instalação do sensor. Assim, os nós sensores podem ser posicionados de forma pré-determinada ou aleatória, permitindo, desta forma, a instalação destes dispositivos em ambientes inacessíveis. Para isso, os algoritmos e protocolos devem ter a capacidade de se auto-organizar devido aos problemas como falhas de comunicação e perda de nós da rede, levando em consideração o tipo de aplicação que será monitorada [Akyildiz et al. 2007]. Esses sensores são gerenciados por um nó *sink*, que é o dispositivo responsável por repassar os dados coletados pela RSSF, como mostra a Figura 2.1, esses dispositivos podem ser instalados usando um posicionamento fixo em um local estratégico da rede, ou móvel, passando a se movimentar pela rede, buscando a melhor localização para coletar e transmitir dados [Baronti et al. 2007].

2.1.1 Aplicações

As redes de sensores sem fio RSSF permitem o desenvolvimento de uma gama de aplicações, sendo empregados em diversos cenários e setores das Cidades Inteligentes. Em face da facilidade de comunicação entre os nós sensores, que é o elemento principal do funcionamento das redes de sensores e das diversas possibilidades de instalações em diferentes tipos de topologias.

Alguns pontos importantes impactam na grande gama de aplicações das RSSF como a possibilidade da mobilidade dos nós *sinks*. Deixando assim, as topologias mais dinâmicas. Desta forma, o trabalho em [Mottola e Picco 2011] relata que as melhorias na conectividade e a busca por padrões de protocolos mais eficientes são pontos chave para

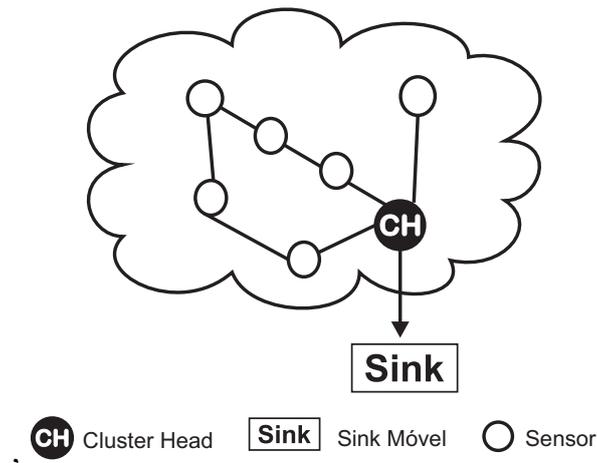


Figura 2.1: Exemplo dos principais componentes de uma RSSF.

estender a vida útil da rede. No entanto, o desenvolvimento de técnicas de priorização e posicionamento proporcionam um aumento no grau de dinamismo, devido à necessidade de posicionar os dispositivos fisicamente móveis [Peixoto 2015, Peixoto e Costa 2015b].

Redes de Sensores atuais podem ser implantadas em diversos ambientes como no solo, no subsolo e debaixo d'água [Baronti et al. 2007]. Assim, dependendo do tipo de ambiente escolhido, a rede de sensores enfrenta diversas imposições e desafios. Existem cinco tipos de RSSFs: terrestre, subterrâneo, subaquática, multimídia e móvel [Baronti et al. 2007]. Estes tipos de redes podem ser aplicadas em diferentes contextos e classificadas em duas categorias: monitoramento e rastreamento. As aplicações de monitoramento podem ser internas ou externas e se aplicam em setores ambientais: saúde, energia, fábrica e automação de processos, monitoramento sísmico e estrutural [Yick et al. 2008]. Já as aplicações de rastreamento possuem diversas abordagens como o rastreamento de animais, seres humanos e veículos [Yick et al. 2008].

Dada a variada gama de aplicação em RSSF, existe uma grande quantidade de requisitos que devem nortear a utilização desta tecnologia. Assim, o desenvolvimento de técnicas de comunicação que envolvam a mobilidade e relevância dos dispositivos móveis é um aspecto importante a ser considerado. Além disso, destaca-se a importância da escolha do padrão a ser adotado de acordo com a aplicação e tipo de rede.

2.1.2 Mobilidade em Redes de Sensores

As redes de sensores são constituídas por uma variedade de dispositivos que possuem na sua maioria limitações de recursos, onde suas características físicas ainda estão distantes de suportar a realidade dos requisitos exigidos pelas aplicações das rede de sensores. Assim, a otimização de algoritmos de roteamento, mecanismos de descoberta da rede, requisitos de segurança e mobilidade, são exemplos de soluções que estão sendo desen-

volvidas para aumentar o tempo de vida de cada dispositivo móvel em uma rede de sensores [Silva et al. 2014]. Para [Anastasi et al. 2009] uma rede de sensores é composta por quatro componentes principais:

- um subsistema de detecção, incluindo um ou mais sensores para coleta de dados;
- um subsistema de processamento de dados, que inclui um microcontrolador e memória para processamento de dados locais;
- um subsistema de rádio para comunicação de dados sem fios;
- uma fonte de alimentação.

O mesmo autor relata ainda que o consumo de energia nos nós sensores, que transmitem em uma frequência de rádio, é da mesma ordem de magnitude tanto na transmissão como na recepção e o estado ocioso reduz o consumo de energia dos nós. Portanto, o rádio deve ser desligado sempre que possível aumentando desta forma a vida útil de cada subsistema [Anastasi et al. 2009]. Desta forma, em alguns casos as técnicas de mobilidade em redes de sensores sem fio devem ser usadas como uma alternativa eficaz para reduzir e otimizar o consumo de energia de modo geral.

Segundo [Silva et al. 2014], a mobilidade em redes de sensores sem fio são justificadas pelos seguintes aspectos: elemento móvel, tipo de movimento, nível de protocolo em que a mobilidade é suportado e a entidade que lida com o processo de mobilidade. Estes aspectos representam características físicas e arquitetônicas da mobilidade em uma rede. Projetos de mobilidade podem ser classificados como mobilidade do *sink* e mobilidade do nó transmissor, dependendo do tipo da entidade móvel envolvida. Para [Anastasi et al. 2009] os nós móveis podem ser divididos em duas grandes categorias: concebida como parte da infraestrutura da rede, ou como parte do ambiente.

Em um modelo de mobilidade descrito em [Camp et al. 2002] sugere que os nós sensores devem reproduzir as características reais de movimentação em tempos razoáveis e velocidade na movimentação do dispositivo. Desta forma, a mobilidade pode ser abordada de várias formas visando objetivos diferentes, levando a uma grande variedade de soluções. Em [Silva et al. 2014], são classificados os elementos móveis de uma RSSF classificando o *sink* como uma estação base móvel para coleta de dados dos sensores. Os *sink* móveis tem o objetivo de movimentar-se para perto de cada *cluster* de nós sensores, a fim de economizar energia e otimizar o caminho da comunicação entre os *sink* e os sensores fixos. Segundo [Silva et al. 2014] os *sinks* móveis podem ser classificados em: (MBS) Estação Base Móvel; (MDC) Coletores de Dados Móvel, e uma solução híbrida fazendo as duas funções. Desta forma, cada estação base é capaz de se mover pela rede aumentando o desempenho e o tempo de vida útil da rede. Os (MDC) usam a capacidade dos nós mais potentes para fazer a coleta dos dados, diminuindo, desta forma, a quantidade de saltos na comunicação até o coletor. As soluções híbridas MBS e MDC posicionam o *sink* com o propósito de cobrir com mais eficiência a demanda da rede. Ainda de acordo com [Silva et al. 2014] a mobilidade do nó sensor pode ser dividida em dois modos básicos: mobilidade fraca e mobilidade forte. A mobilidade fraca é caracterizada pela morte de alguns nós sensores da rede por esgotamento de recursos como bateria. Já a mobilidade

forte, é associada a agentes externos como fenômenos da natureza. Desta forma, outros nós sensores devem movimentar-se para suprir as necessidades de comunicação e gestão de energia da rede gerando assim uma nova topologia.

Para [Achour et al. 2015], a mobilidade dos dispositivos móveis em uma topologia de uma RSSF pode ser classificada em micro e macro mobilidade. A micro mobilidade, refere-se à mobilidade do nó sensor dentro do mesmo domínio de rede. Assim, essa mobilidade muda seu ponto de fixação de um roteador para outro dentro da abrangência da mesma rede. Desta forma, o endereçamento de rede não é alterado. Já a macro mobilidade, refere-se à mudança entre domínios diferentes, alterando desta forma, os endereços já estabelecidos de cada nó da rede. Os protocolos usados nas redes de sensores devem levar em conta estes tipos de mobilidade, por causa de seu impacto sobre os endereços dos dispositivos. Além destas características abordadas, soluções de mobilidade devem considerar o tipo de movimento realizado pelo dispositivo.

Para [Silva et al. 2014], são relacionados três tipos de movimentação dos *sinks* móveis em uma rede que podem ser divididos em: movimento aleatório, pré-definida e controlado. O movimento aleatório significa que o *sink* ou nó sensor se movimenta de forma aleatória sem controle dentro da rede. Já o movimento pré-definido sugere que o dispositivo se movimenta por um caminho específico pré-determinado com velocidade conhecida. Por fim, o movimento controlado, indica que o movimento é controlado por uma entidade externa em tempo real. A mobilidade das redes de sensores estão passando por constantes evoluções, principalmente quando se fala ao nível de protocolo. Desta forma, mobilidade em RSSF podem ser tratada nas camadas MAC, de controle de acesso ao meio ou na camada de rede do modelo de referência em camadas [Silva et al. 2014]. As redes mais tradicionais não suportam qualquer tipo de mobilidade, os sensores devem estar sempre dentro do alcance de seu ponto de acesso local. Já nas RSSF, a mobilidade foi incluída em vários níveis de protocolos de rede. Entretanto, quando a conectividade é essencial para o desempenho e otimização da rede, um maior suporte a mobilidade dos dispositivos deve ser considerado e um maior planejamento se torna necessário, principalmente quando se trata em melhorias nas rotas dos pacotes [Achour et al. 2015].

2.1.3 Roteamento em RSSF

Seja qual for a aplicação de uma rede de sensor, a conectividade entre os nós sensores e os *sinks* deve ser considerada como uma importante medida de desempenho para mensurar a qualidade do serviço na comunicação entre os dispositivos de uma rede de sensores. Os sensores realizam vigilância visual, medem os estresses da construção, analisam a composição do solo entre outras funções. Portanto, [Musznicki et al. 2012] relata que estes tipos de redes são frequentemente implantadas em diferentes tipos de ambientes hostis. Assim, dispositivos altamente limitados em recursos de energia têm que operar por muitos anos sem uma maneira de recarregar ou substituir suas baterias. No entanto, garantir a comunicação direta entre um sensor e o *sink* pode forçar os nós a emitir suas mensagens com um poder tão alto que seus recursos poderiam ser rapidamente esgotados. Desta

forma, os dados são propagados por sensores intermediários e passam por várias rotas até chegar ao *sink* [García Villalba et al. 2009]. Entretanto para [García Villalba et al. 2009], os algoritmos de roteamento que podem ser usados em RSSF podem ser divididos em três tipos: os centralizados, que são executados num nó da rede e detém o conhecimento da estrutura de toda a rede. Segundo o autor este tipo é pouco usado por possuir um custo de transmissão de dados para fazer o sensor ter informações do status de toda a rede. Algoritmos distribuídos onde a comunicação é recebida pela transmissão de mensagens e os algoritmos baseados em locais onde os dispositivos usam informações adquiridas de uma área próxima ao sensor, com as informações dos sensores locais o algoritmo é executado em um nó.

Já [García Villalba et al. 2009] classifica os protocolos de roteamento em protocolos hierárquicos e de coleta de dados. Os protocolos de roteamento hierárquicos selecionam os sensores de acordo com as suas funcionalidades. Com isso, a rede é dividida em grupos ou *clusters*. Um *cluster* é selecionado através da análise de recursos como: energia residual e localização, para coordenar as atividades dentro do grupo de sensores para se comunicar com outros grupos de sensores. Já na coleta de dados, a interação entre os dispositivos da rede pode ser feito de várias formas. O modelo de entrega de dados indica o fluxo de informação entre os nós do sensor e o *sink*. Desta forma, o protocolo de roteamento devem otimizar a comunicação entre os sensores vizinhos aumentando o tempo de vida útil e melhorando a conectividade da rede entre os sensores [García Villalba et al. 2009].

2.1.4 Protocolos de Roteamento

Para [Musznicki et al. 2012], cada algoritmo de roteamento tem características únicas. Assim, os resultados podem ter parâmetros bem diferentes, especialmente em redes densas onde o desempenho do algoritmo pode ser relacionado com a natureza do algoritmo de roteamento. Foi proposto em [Gupta et al. 2015] uma técnica de otimização do roteamento entre sensores agrupados em *cluster* denominada (ROT) (*Route Optimization*). Com os sensores agrupados em *cluster* e considerando a existência de obstáculos, o algoritmo propõe uma forma de estabelecer um caminho eficiente em termos energéticos entre os principais sensores de um grupo de sensores chamado de CH (*Cluster Head*) ou nó cabeça, que foi selecionado por uma técnica que prioriza o nó baseado na estimativa da energia residual de cada sensor, comparando com outros sensores do *cluster*, como mostra a Figura 2.2. O algoritmo ROT usa o menor caminho entre o CH e os *sinks* móveis, usando para isso o algoritmo *Dijkstra*.

Para [Petrioli et al. 2014], a otimização do roteamento geográfico de um pacote em direção ao *sink* proporciona uma diminuição dos saltos e conseqüentemente uma maior eficiência na comunicação da rede. Em redes densas, esta abordagem é eficiente, uma vez que os sensores tem que encontrar o melhor caminho para o *sink* passando por um número limitado de sensores intermediários. O protocolo proposto em [Petrioli et al. 2014] denominado ALBA (*Adaptive Load-Balancing Algorithm*), é uma solução *crosslayer*, onde os protocolos não se comunicam em camadas de rede como na arquitetura tradicional,

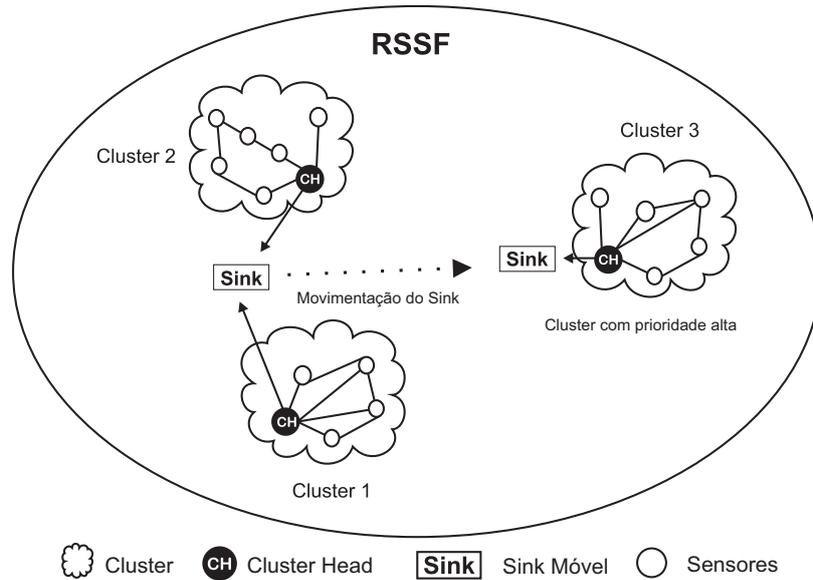


Figura 2.2: Exemplo de otimização do *sink*.

tornando o roteamento mais eficiente em redes de sensores. A proposta prevê integrar horários definidos para transmissão, roteamento geográfico, balanceamento de carga e transmissões de pacotes de *back-to-back*, para otimizar a comunicação com um *sink* móvel. O encaminhamento de pacotes é implementado de forma que o emissor inicia executando uma pesquisa da disponibilidade dos seus sensores vizinhos. Os nós vizinhos disponíveis respondem com um pacote de dados que permite que o remetente escolha o melhor sensor para iniciar a rota. O roteamento geográfico em direção ao *sink* é feito usando os sensores que têm o melhor desempenho para o encaminhamento dos pacotes. Para isso, é usada informações de posicionamento fornecidas por GPS (*Sistema de Posicionamento Global*) a um sensor, cada sensor calcula índice de priorização denominado pelo autor de "GPI", que é referente ao número da região geográfica de encaminhamento do sensor que está enviando os dados. A numeração das regiões pode variar de 0 a $n - 1$. Os números são atribuídos de modo que quanto maior o número da região, mais longe do coletor estão os sensores, ou seja, os nós na região 0 fornecem um caminho mais rápido em direção ao *sink* [Petrioli et al. 2014].

Segundo [Musznicki et al. 2012], a utilização de roteamento *multicast* não otimiza a comunicação entre sensores. Porém, utilizar algoritmo de roteamento *multicast* associado ao algoritmo de menor caminho de *Dijkstra* pode alcançar ótimos resultados para o consumo de energia em uma RSSF. Para [Musznicki et al. 2012], métodos de entrega de informações *multicast* podem ser divididos em vários tipos, dependendo da forma como o grupo de destino é identificado. O primeiro método estudado é a inundação em rede, onde cada sensor recebe uma mensagem processa e transmite a cada vizinho dentro da sua faixa de alcance do sinal de rádio, porém este método pode levar ao congestionamento e esgotamento de recursos de rede, causando replicações de mensagens na rede. Um

outro método de distribuição *multicast* descrito em [Musznicki et al. 2012] é o encaminhamento geográfico, que consiste em direcionar o fluxo de dados a um grupo de sensores específicos, este grupo pode receber um identificador dedicado ou um endereço do grupo com base em posições geográficas, para determinar as coordenadas, alguns sensores podem ser equipados com receptores GPS. Portanto, quando as coordenadas forem conhecidas, os grupos de sensores podem ser relacionados por meio da sua localização. Desta forma, [Musznicki et al. 2012] propôs em seu trabalho o algoritmo DLEMA (*Dijkstra-based Localized Energy-Efficient Multicast Algorithm*). o DLEMA usa uma árvore de caminho mais curto calculada com o algoritmo *Dijkstra*. Com isso, cada caminho entre o sensor atual e o destino é considerado o caminho de menor custo em uma árvore *multicast* criada localmente. Assim as rotas criadas pelo algoritmo de *Dijkstra*, geralmente mostram um menor número de conexões, o que reduz o atraso médio da propagação de mensagens relacionadas à estrutura da árvore *multicast* melhorando desta forma o consumo de energia dos dispositivos da rede.

2.1.5 Comunicação por Clusters em RSSF

Seja qual for a aplicação de uma rede de sensores, a infraestrutura da rede é de extrema importância para a performance da rede, assim como a disposição dos nós sensores na formação da rede. Os nós sensores são geralmente instalados em uma área para o monitoramento de alguma atividade e a disposição destes sensores pode ser feita de várias formas. Assim, antes de iniciarem as atividades de sensoriamento, os sensores devem iniciar realizando atividades de descoberta da rede, localização e formação de *cluster* [Gupta et al. 2015]. Os autores afirmam que técnicas de agrupamentos de sensores aumentam a vida útil da rede, principalmente quando os sensores são instalados em ambientes hostis. Vários protocolos têm sido apresentados na literatura sobre o tema, onde abordam melhorias na infraestrutura da rede de sensores em *cluster*. A abordagem proposta em [Gupta et al. 2015], relata que quando acontece um agrupamento é selecionado um CH (*Cluster Heads*) e alguns protocolos, definem a melhor forma para esta seleção, por exemplo, a seleção do CH pode ser baseado na energia residual de cada sensor e uma estimativa de quantos dos seus vizinhos irá se beneficiar com ele sendo um CH.

A transmissão de dados em rede baseada em *cluster* pode alcançar um maior nível de escalabilidade e de gestão, o que maximiza a vida útil da rede e reduz o consumo da largura de banda. Desta forma, em uma rede baseada em *cluster*, cada *cluster* tem um nó sensor líder já definido como CH. O CH coleta os dados dos nós sensores e encaminhando para a uma estação de base [Lu et al. 2014]. Apresentado por [Lu et al. 2014], o algoritmo (LEACH) foi desenvolvido para minimizar o consumo de energia dos sensores, o algoritmo gira aleatoriamente o sensor eleito como CH entre todos os nós sensores na rede, proporcionando um aumento no tempo de vida da rede.

2.2 Cidades Inteligentes

As Cidades Inteligentes (*Smart Cities*) constituem um sistema de pessoas interagindo para o desenvolvimento urbano e econômico usando tecnologias de automação para a melhoria da qualidade de vida de seus habitantes [Peixoto e Costa 2017a]. As *Smart Cities* representam um campo multidisciplinar, sempre modelada por avanços na tecnologia e desenvolvimento urbano [Angelidou 2015].

De acordo com [Letaifa 2015] as *Smart Cities* são definidas como uma cidade que monitora e integra as condições de todas as suas infraestruturas críticas, incluindo estradas, pontes, túneis, trilhos, metrô, aeroportos, portos, comunicações, água, energia, até mesmo grandes edifícios, pode organizar melhor seus recursos e planejar as suas atividades, monitorando assim, os aspectos de segurança e melhorando os serviços aos seus cidadãos. Esses modelos de interação são classificados como inteligentes por fazer uso estratégico da infraestrutura de serviços, informação, comunicação, planejamento e gestão urbana para dar resposta às necessidades sociais e econômicas [Peixoto e Costa 2017a].

Segundo [Angelidou 2015], durante os anos 1960, 1970 e 1980, houve um grande aumento no desenvolvimento de trabalhos científicos publicados que envolviam temas relacionados as Cidades Inteligentes. Inicialmente na década de 80, surgiu a ideia de instrumentar a cidade com redes cabeadas o que levou à popularização do conceito de cidades com fio e a generalização de concepções relacionada as cidades, como as Cidades Inteligentes, Cidades Criativas, Cidades Digitais e Cidades Virtuais. Essas ideias representam perspectivas de um futuro com o desejo de aprimorar a infraestrutura urbana existente e ampliar a popularização das Tecnologias de Comunicação e Informação (TIC).

Para [Letaifa 2015], o nascimento de Cidades Criativas revelam uma tendência oposta. Cidades Criativas delegam iniciativas baseadas na comunidade urbana e no setor privado, laboratórios vivos e empreendedorismo social. Desta forma, Cidades Criativas enfrentam o desafio de transformar o processo de ideação em inovação e mudança estrutural. Iniciativas pequenas e independentes muitas vezes não conseguem crescer e se tornar sustentável. Cidades em todo o mundo enfrentam problemas de crescimento urbano e econômico. Os líderes mundiais devem conceber novas estratégias para melhorar o desempenho de cidades e sustentabilidade urbana. Assim, governantes de todo o mundo passaram a optar por transformar as cidades em Cidades Inteligentes, para que os cidadãos, empresas e governos possam acessar de forma mais eficiente, serviços e recursos [Letaifa 2015].

A estrutura básica de uma Cidade Inteligente requer a implementação e integração das redes urbanas e a criação de redes inteligentes conectando os diversos setores em uma cidade. Segundo [Mattoni et al. 2015], os nós da rede são compostos pelos humanos e elementos urbanos que juntos se transformam em nós das redes inteligentes. Se cada nó da rede executa mais de uma função, as atividades são otimizadas e os resultados e performances mais eficientes são alcançadas. Já [Mattoni et al. 2015] divide a estrutura conceitual de uma Cidade Inteligente em seis eixos: Pessoas, como capital social e humano, economia representando a competitividade, meio ambiente com os recursos naturais, mobilidade relacionada a transportes, as (TICs), governança e de estar representada pela qualidade

de vida dos habitantes da cidade. O autor relata ainda que o principal objetivo de um projeto para uma *Smart City* parte de imaginar a cidade como um organismo único, a fim de gerenciar de forma integrada as suas estruturas básicas.

2.2.1 Construindo Cidades Inteligentes: Exemplos

Um exemplo de Cidade Inteligente relatado por [Angelidou 2015] está sendo implementado em Barcelona, Espanha. Onde, uma estratégia inteligente inclui tanto o uso de alta tecnologia e avanço do capital humano através do desenvolvimento da economia e do conhecimento. Outro exemplo abordado por [Angelidou 2015], é o plano de Londres para a Cidade Inteligente. A ideia estratégica é usar o poder criativo das pessoas e novas tecnologias para servir e melhorar a vida dos Londrinos. Tendo os avanços tecnológicos como alvo para a realização de futuras áreas urbanas. O projeto será desenvolvido no *Queen Elizabeth Olympic Park*, usando os edifícios da imprensa que fizeram parte dos centros dos Jogos Olímpicos de 2012. Será um campus para apoiar o crescimento do setor de tecnologias de Londres através da combinação de negócios, tecnologia, mídia, educação e dados, tendo como objetivo criar um sistema local de inovação. Este espaço, deverá proporcionar um ambiente propício para o desenvolvimento de projetos relacionados às *startups*, educação e pesquisa de pós-graduação.

2.2.2 Tecnologias Envolvidas

As arquiteturas urbanas combinam um conjunto sistemático das (*TICs*) para implementar serviços e recursos nas Cidades Inteligentes [Yin et al. 2015]. Os avanços nos sistemas computacionais, redes de telecomunicações, telefonia móvel, sistemas de armazenamento de informações se tornam mais eficientes com o avanço da internet, além de toda tecnologia embarcada nos produtos eletrônicos, isso possibilita o desenvolvimento de diversas tecnologias no âmbito das Cidades Inteligentes [Yin et al. 2015]. Diante disto, as Cidades Inteligentes tornaram-se um elemento central para o uso de tecnologias da informação ao abordar vários problemas em ambientes urbanos, como mobilidade, segurança pública, eficiência energética e resposta emergente [Su et al. 2011, Batty et al. 2012]. Muitas soluções foram criadas para diferentes cenários empregando novos desenvolvimentos de tecnologias como a Internet de Coisas (IoT) [Ahmed et al. 2016], *Big Data* [Hashem et al. 2016], algoritmos de Aprendizagem de Máquina [Qiu et al. 2016], serviços baseados na nuvem [Botta et al. 2016], entre outros. A integração do IoT com um ambiente inteligente amplia as capacidades dos objetos inteligentes. O trabalho em ambientes inteligentes baseados em IoT geralmente pode ser classificado nas seguintes áreas: Cidades Inteligentes, Casas Inteligentes, Redes Inteligentes, Edifícios de Arte, Transporte Inteligente, Saúde Inteligente e Indústria Inteligente [Ahmed et al. 2016]. Os sistemas de *Big Data* auxiliam as Cidades Inteligentes armazenando processando e mineração dados de forma eficiente para produzir informações e aprimorar diferentes serviços urbanos usando as ferramentas e métodos adequados para uma análise de dados eficiente

e efetiva [Hashem et al. 2016]. Uma outra tecnologia existente nos projetos de Cidade Inteligente são os algoritmos de Aprendizagem de Máquina [Qiu et al. 2016]. Estes algoritmos foram usados em áreas complexas com um intensivo processamento de dados, como medicina, astronomia, biologia, etc. [Qiu et al. 2016]. Já a computação nas nuvens fornece serviços para executar tarefas complexas de computação em larga escala, como a mineração de dados de redes sociais [Botta et al. 2016]. Os serviços de computação em nuvem podem ser combinados com o IoT integrando dispositivos de monitoramento, dispositivos de armazenamento, ferramentas de análise, plataformas de visualização entre outros [Botta et al. 2016]. Para integrar todas estas tecnologias as RSSF será o bloco de construção fundamental dos sistemas de Cidades Inteligentes [Rawat et al. 2014]. A coleta de informações relevantes sobre o comportamento dinâmico de diferentes elementos relacionados à maneira de viver em cidades modernas, como temperatura, poluição, nível de chuva, fluxo de tráfego e radiação UV, os sensores fornecem informações para apoiar serviços que possam melhorar a qualidade de vida e o suporte ações de emergência em áreas urbanas. No entanto, a grande quantidade esperada de dados produzidos por sensores potencialmente estressará os canais de comunicação disponíveis, o que promoverá o desenvolvimento de soluções de otimização nesse sentido. Portanto, as Cidades Inteligentes são baseadas em toda uma interpretação sobre o desenvolvimento dos processos urbanos nas cidades, baseado na popularização das (TIC) que continua a influenciar as políticas de desenvolvimento urbano em uma escala global.

2.3 Problema do Caminho Mais Curto

Nos últimos anos, tem-se observado um avanço nos processos de resolução de problemas em arquiteturas de redes de comunicação, solucionados usando o problema de caminho mais curto, que é um dos problemas genéricos de diversas áreas como Engenharia de Transportes, Automação, Computação e Inteligência Artificial [Lassabe et al. 2007], [Khan et al. 2014]. De fato, com a evolução dos computadores, principalmente como se fala de memória, velocidade de execução e de resolução de modelos matemáticos, como por exemplo, os algoritmos para determinar os caminhos mais curtos de uma rede, é hoje possível determinar de forma eficiente (para certos tipos de redes) todas as soluções dominadas de um problema de caminho mais curto. Outras possibilidades de aplicação incluem quaisquer problema envolvendo redes ou grafos em que se tenha grandezas como: distâncias, tempo, perdas, ganhos e despesas.

O problema do caminho mais curto consiste em encontrar o caminho mais curto entre dois pontos em uma rede ou, em última análise, um grafo, que consiste em descobrir o caminho de menor custo entre dois pontos considerando a soma dos custos associados aos caminhos percorridos entre a origem e o destino [Khan et al. 2014]. Os algoritmos de caminho mais curto são frequentemente usados em aplicações de transporte e se adaptam a diversas situações práticas [Lassabe et al. 2007]. Muitos desses algoritmos são otimizados e dão desempenhos satisfatórios. Entretanto, outras possibilidades de aplicação incluem problemas envolvendo redes ou grafos que possuam medidas como distâncias e tempo, onde

estas variáveis se acumulam ao longo do percurso da rede. Para [Divoky e Hung 1990] os algoritmos de busca do caminho mais curto são utilizados para resolver problemas de emparelhamento, problemas do fluxo de custo mínimo e problemas de atribuição de rota.

As diferentes situações, que na prática implicam em encontrar o caminho mais curto têm feito com que este problema tenha sido dividido em diferentes casos, a fim de facilitar seu estudo, pesquisas e desenvolvimento. Segundo [Dreyfus 1969], existem cinco tipos de problemas envolvendo o caminho mais curto:

- estabelecer o caminho mais curto entre dois nós específicos de uma rede;
- estabelecer o caminho mais curto entre todos os pares de nós da rede;
- definir o segundo, terceiro, quarto, etc., caminho mais curto;
- localizar o caminho mais rápido numa rede com tempos de viagem e dependendo de uma hora de saída;
- encontrar o caminho mais curto entre dois nós específicos precisando-se atravessar obrigatoriamente determinados nós intermediários.

O problema do caminho mais curto é um tema atual destinado a diversas aplicações práticas, pois sua resolução permite obter soluções eficientes para resolução de problemas reais ao determinar de modo confiável a forma mais rápida e econômica de realizar uma atividade que requer a definição da melhor rota. Para [Ahuja et al. 1990], outras aplicações para resolver o problema do caminho mais curto estão relacionadas com a verificação ótima relacionada a substituição de equipamento, transmissão de mensagens em sistemas de comunicação e fluxo de tráfego em cidades com grandes congestionamentos. Para [Aini e Salehipour 2012], o problema do caminho mais curto é um problema fundamental e bem conhecido na pesquisa de operações relacionada com a busca de um caminho entre dois nós (vértices) de um grafo, de modo que a soma dos pesos (custo, distância, tempo, etc.) de suas bordas de conexão seja minimizada. O problema do caminho mais curto tem muitas aplicações do mundo real, uma das mais comuns é a de encontrar o caminho mais rápido através de uma rede rodoviária [Aini e Salehipour 2012]. Desta forma, foram desenvolvidos vários algoritmos especializados em solucionar o problema em encontrar o caminho mais curto entre dois pontos em uma rede. Esta estratégia consegue reduzir significativamente o número de nós visitados em um grafo, economizando recursos essenciais para o aumento da disponibilidade dos dispositivos na rede.

2.3.1 Algoritmos Especializados

Para resolver o problema do caminho mais curto foram desenvolvidos muitos algoritmos com resultados e eficiência diversa, o mais encontrado na literatura é o algoritmo de *Dijkstra*. Para [Musznicki et al. 2012], este algoritmo permite determinar uma solução ótima através da adição de vértices a árvore de caminho mínimo. Em outras palavras, consiste em verificar se há a possibilidade de melhorar o caminho obtido em uma rota.

Em [Musznicki et al. 2012], os autores mencionam que os caminhos criados pelo algoritmo de *Dijkstra* geralmente mostram um menor número de conexões, o que diminui o atraso médio da disseminação de mensagens relacionadas à estrutura da árvore *multicast* inicial melhorando o gerenciamento de energia.

O algoritmo de *Dijkstra* concentra-se em descobrir os caminhos mais curtos através de nós que fornecem o máximo de avanço geográfico para destinos desejados [Sarkar e Murugan 2016]. Já [Ahuja et al. 1990] afirma que o algoritmo de *Dijkstra* é conhecido teoricamente como o mais eficiente para resolver o problema de caminho mais curto. Em uma análise feita por [Ahuja et al. 1990] no algoritmo de *Dijkstra*, foi verificado que sua ordem de complexidade é $O(n^2)$, onde n é o número de nós, ou seja, na pior das hipóteses, o tempo de execução cresce. Além disso, relata que trabalhos posteriores demonstraram que a complexidade pode ser reduzida para $O(m \log n^2)$, onde m é o número de arcos, simplesmente substituindo a lista que recebe os nós de um conjunto F por uma fila de prioridades.

Para [Ahuja et al. 1990], a chave para a implementação eficiente do algoritmo de *Dijkstra* é o uso de uma estrutura de dados chamada *heap* (ou fila de prioridade). O *heap* consiste em um conjunto de itens, cada um com uma chave de valor real associada, ou seja, é uma alternativa de implementação do algoritmo de *Dijkstra* a fim de encontrar o nó com etiqueta de menor custo. *Dijkstra* é a base de vários algoritmos heurísticos que traz considerável incremento de desempenho, mantendo a complexidade e garantindo a originalidade do código inicial. Um dos problemas do algoritmo de *Dijkstra* é o fato do algoritmo não admitir pesos negativos, em uma comparação com o algoritmo *Bellman-Ford* que admite ou reconhece as bordas com pesos negativos, o que gerará inúmeros caminhos do ponto de partida para o destino final onde cada ciclo minimizará o comprimento do caminho mais curto [Magzhan e Jani 2013].

Outro algoritmo proposto em [Liu e Hong 2006] foi desenvolvido para o roteamento de tráfego e energia eficiente, sendo definido como *Traffic-Aware Energy Efficient* (TAEE). O algoritmo explora informações de carga do tráfego, além de níveis de resíduo de potência para otimizar a distribuição de carga da rede RSSF, podendo desta forma, aumentar o tempo de vida da rede. Para isso, é incorporado ao algoritmo uma métrica para quantificar a carga do tráfego em potencial e esta métrica é usada para determinar o caminho com menor custo para a rede. O TAEE maximiza o tempo de vida da rede aproveitando as reservas de energia proativa dos nós que fazem parte da rota. Além disso, para redes de sensores com grande escala, os autores sugerem usar uma formação aleatória de zona (chamadas de grupos) como parte do roteamento hierárquico heurístico que executa o algoritmo TAEE localmente entre os nós no interior dos grupos e de forma aleatória.

Segundo [Liu e Hong 2006] o algoritmo TAEE tem duas operações principais. Uma operação é para construir uma árvore geradora mínima de menor caminho e a segunda operação é para nós de árvore de acordo com o valor do PL (o valor de PL reflete a probabilidade de que um nó pode participar do roteamento). Assim, a árvore é gerada e atualizada por cada nó e será usada para disseminação de mensagem em *multicast* e para a estimativa de carga de tráfego. Em sua pesquisa, [Liu e Hong 2006] usa para calcular o caminho

mais curto entre os pares na árvore um algoritmo existente, definido como algoritmo de *Floyd-Warshall*, considerando que para calcular o custo para caminho mais curto, nível de energia residual e o consumo de energia de transmissão ao longo da rede. A seleção do caminho para o roteamento será ainda mais otimizada para balanceamento de carga usando o valor do PL. Especificamente, os autores usaram o PL para aparar nós da árvore acima mencionada.

Segundo [Khan et al. 2014], o algoritmo *Warshall* é usado para encontrar o fechamento transitivo de um grafo. Mas, esse algoritmo não é exatamente um algoritmo de roteamento de caminho mais curto. Há outro algoritmo chamado de *Floyd* que é usado para encontrar o caminho mais curto da matriz de caminhos. Se os pesos são atribuídos aos caminhos no algoritmo de *Warshall* e os caminhos mais curtos entre todos os pares são descobertos, então isso se chama algoritmo de *Floyd*. Assim os algoritmos de *Warshall* e *Floyd* juntos são chamados de algoritmo de *Floyd-Warshall*. Para [Khan et al. 2014] o algoritmo de *Floyd-Warshall* não garante transmissão segura de dados em um RSSF, devido à ausência de confirmação de recebimento de dados. Os autores sugerem uma modificação no algoritmo para torna-lo mais eficiente em relação ao roteamento de caminho mais curto em uma rede de sensores sem fio. As modificações implementadas no algoritmo de *Floyd-Warshall* existentes são as seguintes:

- um caminho entre dois nós tem outro caminho semelhante do mesmo peso na direção inversa, entre os mesmos;
- se nenhum caminho de reconhecimento direto ou caminho reverso estiver presente entre dois nós, então não pode haver um caminho direto entre eles;
- se o peso de um caminho não é igual ao do seu caminho inverso, então esse caminho bem como o seu caminho inverso é inválido.

Assim, qualquer nó pode comportar-se como um nó sensor ativo, e qualquer nó pode ser o coordenador de acordo com a escolha do usuário ou de acordo com os requisitos da infraestrutura da rede. Nesta nova abordagem do algoritmo de *Floyd-Warshall*, [Khan et al. 2014] considerou que se qualquer nó roteador estiver presente, o algoritmo não vai reconhecer o recebimento de dados (embora recebido com sucesso) para o remetente, em seguida o remetente não vai perceber a presença de nó receptor. Esta situação pode ocorrer devido as interferências de RF (rádio frequência) ou qualquer outro tipo de obstrução presente entre os dois nós. Se o nó estiver ausente, então não vai ser capaz de transmitir os dados para o próximo roteador ou coordenador. Desta forma, este algoritmo modificado torna-se um algoritmo de roteamento de caminho mais curto, útil em redes de sensores sem fio.

Em [Aini e Salehipour 2012], foi introduzido uma nova abordagem para calcular o caminho mais curto em redes com ciclos. Esta contribuição gerou um novo algoritmo que reduz substancialmente a quantidade de cálculo exigida pelo algoritmo *Floyd-Warshall*. Na abordagem proposta por [Aini e Salehipour 2012], um algoritmo retangular insere melhorias no *Floyd-Warshall*, considerado pelos autores como um dos melhores algoritmos disponíveis para tratar este problema. O algoritmo de *Floyd-Warshall* e o algoritmo re-

retangular tem exatamente o mesmo desempenho na determinação da matriz, no entanto, o algoritmo retangular deriva das matrizes associadas muito mais rapidamente devido à quantidade reduzida de cálculo.

Em uma outra abordagem descrita por [Magzhan e Jani 2013], algoritmos inteligentes foram desenvolvidos para encontrar caminhos mais curtos em situações que exigem um grande espaço de busca, dentro de prazo limitado e também para acomodar um ambiente em constante mudança. Portanto, [Magzhan e Jani 2013] relata que o principal algoritmo inteligente usado para o problema do caminho mais curto é o Algoritmo Genético (AG). Portanto, os Algoritmos Genéticos são usados principalmente para resolver problemas de otimizações. Ele usa várias operações genéticas, tais como seleção de passagem e mutação, a fim de gerar uma nova geração da população, que representa um conjunto de soluções para o problema. Cada indivíduo ou cromossomo dentro da população receberá um valor de aptidão, que é calculado com base em uma função de aptidão predeterminada que mede o quanto é ótima para resolver o problema atual.

Segundo [Magzhan e Jani 2013], para resolver o problema do caminho mais curto usando o AG, é necessário gerar uma série de soluções e em seguida, escolher a mais ideal entre o conjunto fornecido de possíveis soluções. Na solução proposta por [Magzhan e Jani 2013], uma população inicial é criada de forma aleatória formando um primeiro conjunto de cromossomos a ser usado no AG. Cada cromossomo representa uma solução possível para o problema atual em questão. Depois disso, eles (cromossomos) são estimados usando determinada função, determinando desta forma, as melhores soluções. Considerando o valor da aptidão de cada solução ou cromossomo, serão selecionados alguns cromossomos ou indivíduos, e as operações genéticas básicas, como cruzamento e mutação, são aplicadas nestes cromossomos. Então, o valor de aptidão de cada cromossomo é recalculado e as melhores soluções são selecionadas para serem consideradas para a próxima geração. Este processo continua até que os critérios do problema dado não sejam alcançados.

A complexidade de cada um dos algoritmos descritos abaixo na Tabela 2.1, mostra que esses algoritmos são satisfatórios em termos de performance e usabilidade na solução do problema do caminho mais curto. Entretanto, foi observado que várias pesquisas procuram modificar os algoritmos em busca de um melhor desempenho para uma solução específica. No entanto, o algoritmo de *Dijkstra* foi o que mais se destacou nas literaturas analisadas, sendo modificado para atender a diversas soluções de rede, especialmente quando se trata de RSSF.

Tabela 2.1: Algoritmos de caminho mais curto.

Algoritmos Especializados		
Algoritmos	Descrição	Aplicado em RSSF
Dijkstra	Algoritmo utilizado em buscas de menor caminho em uma rota, partindo de um ponto qualquer.	Sim
TAAE	Algoritmo explora informações de carga do tráfego, além de níveis de resíduo de potência para otimizar a distribuição de carga da rede.	Sim
Floyd-Warshall	Algoritmo utilizado em buscas de menor caminho em grafos orientados e valorados.	Sim
Algoritmo Genético	Algoritmos genéticos são usados principalmente para resolver problemas de otimizações e encontrar caminhos mais curtos em situações que exigem um grande espaço de busca.	Sim

2.3.2 Aplicações em Cidades Inteligentes

As Cidades Inteligentes podem fornecer benefícios substanciais que melhoram a vida diária das pessoas. Nos dias atuais é possível aplicar algoritmos para resolver problemas de rota relacionados à arquitetura das redes de sensores estabelecidas nas Cidades Inteligentes. O congestionamento do tráfego de veículos, por exemplo, é um problema significativo que surge em muitas cidades. Isto é devido ao número crescente de veículos que estão dirigindo nas vias das cidades. Segundo [Amer et al. 2016], com o crescimento explosivo da economia global, houve uma grande expansão das cidades e como consequência um aumento no número de veículos que dirigem em redes rodoviárias das cidades. Isso provocou um aumento extremo do congestionamento do tráfego. Portanto, os algoritmos de caminho mais curto podem ser usados em aplicações de transporte e se adaptam a diversas situações práticas das cidades, muitos desses algoritmos são otimizados e dão desempenhos satisfatórios [Lassabe et al. 2007]. Para resolver o problema de congestionamento de tráfego em Cidades Inteligentes, [Amer et al. 2016] propôs o uso da função de custo "multi-objetivo" e planejamento de rota dinâmica. para obter uma redução no tempo de viagem, consumo de combustível e emissão de CO_2 . O método proposto leva a uma redução no tempo de viagem, consumo de combustível e emissão de CO_2 . O algoritmo denominado de ISATOPSIS, fornece a cada veículo o caminho menos congestionado de acordo com a condição da estrada, podendo melhorar o fluxo de tráfego em uma média geral, trazendo importantes benefícios para uma Cidade Inteligente.

Em [Zhang et al. 2016], foi desenvolvido um algoritmo de planejamento de rotas, tendo

como base o algoritmo de *Dijkstra*, a proposta é melhorar as rotas de veículos em áreas urbanas congestionadas. O algoritmo considerou o peso do consumo de óleo denominado OCW e métodos de planejamento de rotas para desenvolver um algoritmo denominado *OCW-Dijkstra*. No algoritmo, os parâmetros relativos ao veículo e ao ambiente de condução, como distância, velocidade, tempo de condução, tempo de marcha lenta, fluxo de viagem, consumo de óleo e consumo de óleo ocioso, são utilizados para produzir o OCW com cálculo ponderado em cada seção da viagem. O algoritmo gera uma matriz de adjacência carregando a descrição do segmento, roteamento regional e a informação do ponto em um mapa urbano. Após o ponto inicial e o ponto de destino serem selecionados, a rota ideal é gerada automaticamente. Além disso, o algoritmo possui métodos de autoaprendizagem, que podem atualizar os parâmetros e o OCW em tempo real. A partir dos resultados da simulação de experimentos e da comparação com outros algoritmos, chegaram a conclusão que o algoritmo *OCW-Dijkstra* funciona de forma mais efetiva e robusta, o que conseqüentemente economiza tempo de condução.

Alguns algoritmos de roteamento que abordam o caminho mais curto, visam reduzir o consumo de energia em uma RSSF. Diversos trabalhos relatam que o algoritmo de *Dijkstra* é usado para o encaminhamento através do caminho mais curto em um RSSF [Khan et al. 2014]. De fato, este algoritmo aplicado a Cidades Inteligentes consegue resolver o problema de caminho com menor custo, podendo ser usado em diversas aplicações para as RSSF, otimizando problemas complexos como definir a rota ideal para o trafego de veículos em ambientes urbanos de Cidades Inteligentes.

2.4 Relevância e Priorização

Diversos trabalhos abordam o conceito da utilização de relevância em nós sensores de uma RSSF tendo como principal objetivo a eficiência energética dos nós e a otimização da transmissão de dados entre os nós sensores. Para [Costa 2013], o uso de relevância como índices de prioridade do sensor, indicam a importância de cada nó como fonte visual para toda a função de monitorização de um aplicativo, podendo este nó ser explorado por diferentes mecanismos de otimização alcançando desse modo, um melhor desempenho na transmissão de dados. Assim, nós com alta relevância têm prioridade no tráfego de dados [Costa 2013, Peixoto e Costa 2017a]. Entretanto, sensores que transmitem pacotes de dados indicados como de baixa relevância podem ser descartados, com o propósito de economizar energia [Peixoto e Costa 2017a]. Alguns sensores podem evitar a transmissão de pacotes de baixa relevância reduzindo o consumo de energia local e indiretamente preservar a energia de sensores intermediários. Ao priorizar nós na rede, pacotes de dados menos relevantes podem ser descartados, isso proporcionará uma diminuição do congestionamento da rede, reduzindo assim o impacto no percentual de dados recebido pelo *sink* [Costa e Guedes 2011].

Para [Costa et al. 2015], os parâmetros usados na priorização podem ter um âmbito de aplicação local ou global. No contexto local, a transmissão está relacionada apenas com

o fluxo de transmissão de um nó origem. Já no contexto global, as abordagens de priorização são pertinentes para toda a rede.

Desta forma, definir o nível de relevância e o parâmetro de priorização dos nós sensores em uma RSSF, é considerado pelos autores pesquisados, como um princípio importante para otimizar a comunicação na rede.

2.4.1 QoS (Quality of Service)

Qualidade de Serviço (QoS) é um conjunto de requisitos que estão relacionados com a qualidade de serviços a serem priorizados ao transportar dados da origem ao destino pelas redes. Neste cenário, QoS refere-se a uma garantia da rede de fornecer um conjunto de requisitos mensuráveis de serviço para os usuários em termos de atraso, *jitter*, largura de banda disponível e perda de pacotes [Chen e Varshney 2004]. De acordo com [Chen e Varshney 2004], por terem características específicas, diversos tipos de redes de comunicação podem ter comportamentos diferentes em relação ao QoS, podendo desta forma, estabelecer restrições inerentes ao suporte do QoS específicos para cada tipo de rede. Por exemplo, a restrição de largura de banda e topologia dinâmica das redes móveis *Ad Hoc* fazem o suporte de QoS muito mais desafiador do que em outros tipos de rede. Portanto, requisitos de QoS gerados pelas aplicações de RSSF podem ser muito diferentes, parâmetros de QoS fim a fim tradicionais podem não ser suficientes para este tipo de rede. O desenvolvimento de novos parâmetros de QoS são desejados para a medição da entrega dos dados do sensor de forma eficiente e eficaz [Chen e Varshney 2004]. Com isso, projetistas de rede serão capazes de investigar qual arquitetura de QoS ou mecanismo pode ser explorado para fornecer um suporte de QoS para as aplicações de uma RSSF. Assim, o conceito de qualidade de serviço mensura a qualidade dos serviços ofertados por uma rede de comunicação, ou seja, mostra o quanto ela é capaz de atender às expectativas de seus usuários através dos serviços que a mesma oferece.

2.4.2 QoE (Quality of Experience)

Uma outra forma de otimizar a rede usando o conceito de relevância é usando a técnica de QoE, que está relacionado a um nível de aceitação geral de um aplicativo ou serviço e como este nível é percebido pelo usuário final. Para [Baraković e Skorin-Kapov 2013], os usuários móveis estão cada dia mais exigentes em relação às aplicações de rede e multimídia não havendo limitações por restrições de tempo, localização, tecnologia, dispositivo e restrições de mobilidade. Assim, a comunicação confiável passa a ser a qualquer hora e em qualquer lugar, por qualquer meio e tecnologia de rede. Desta forma, o autor afirma que o objetivo geral do QoE na rede esta relacionado com a perspectiva do usuário final.

O usuário pode mensurar a qualidade de um transmissão de dados e esse resultado pode ser transformado em valores numéricos. Desse modo, QoE pode ser usada como uma métrica subjetiva que pode variar de acordo com a expectativa do usuário em relação a

um contexto. Além disso, o QoE é usado como um indicador confiável para os fornecedores de serviços e operadores de Telecomunicações [Pokhrel et al. 2014]. Assim, a estimativa de qualidade no tráfego de vídeo pode ser dividida em subjetiva ou objetiva. Os métodos subjetivos são compostos de uma análise de vários usuários que visualizam as amostras multimídia e classificam sua qualidade de acordo com uma escala de qualidade pré-definida. Alguns dos métodos subjetivos de teste disponíveis para estimar a qualidade de vídeo são a pontuação média da opinião dos usuários (MOS) ou a escala dupla de estímulo contínua da qualidade (DSCQS), entre outros. Uma estimativa de QoE é uma métrica subjetiva, no entanto, métodos objetivos tentam medir a qualidade de vídeo com base em parâmetros do sistema ou mídia. É feita uma tentativa de modelar QoE com base em métricas avaliadas objetivamente relacionadas à satisfação dos usuários, sem a participação do usuário. Isso reduz o custo e o tempo de avaliação [Pokhrel et al. 2014].

Capítulo 3

Metodologia de Pesquisa

Neste capítulo, apresenta-se a metodologia utilizada para o desenvolvimento de um algoritmo de otimização com o propósito de determinar a melhor rota e posicionamento de múltiplos *sink* móveis que fazem parte de uma RSSF reativa em uma Cidade Inteligente. Relativamente à metodologia utilizada na presente dissertação, a sua sustentação teórica passa por uma criteriosa e exaustiva busca de elementos que permitem o estabelecimento do seu estado da arte.

Diversas abordagens e algoritmos que tratam sobre o assunto aqui pesquisado, foram estudados para descobrir e compreender soluções existentes. Após a fase teórica, torna-se necessário proceder à operacionalização dos resultados experimentais, de modo a avaliar sua viabilidade e possível efetividade.

3.1 Escopo da Solução

Para este trabalho, foi considerado o ambiente urbano como o escopo conceitual do problema. Como ambiente urbano entendemos como a zona urbana dos municípios, chamada aqui, por simplicidade, como "cidade". Além disso, o problema está inserido no contexto das Cidades Inteligentes, que, grosso modo, são definidas como cidades que possuem um ou mais sistemas automatizados que operam para prover algum tipo de serviço que melhore a vida dos seus habitantes [Yin et al. 2015]. Em outra definição comum, este conceito parte da caracterização de uma Cidade Digital cujo os procedimentos de comunicação, informação e monitoramento são informatizados. Em [Yin et al. 2015], o autor relata que somados a uma camada de inteligência, processos referente ao ambiente urbano podem tomar decisões de alto nível com base em técnicas relacionadas à área de inteligência artificial, mineração de dados, *machine learning*, entre outras, onde as aplicações são automatizadas e focada no uso prático e experiência do usuário [Batty et al. 2012, Su et al. 2011, Yin et al. 2015].

O ambiente urbano definido neste trabalho considera que em uma determinada cidade genérica irá ser implantado uma ou mais redes de sensores sem fio, compos-

tas por um ou mais nós escalares e/ou multimídias que irão desempenhar algum tipo de monitoramento (fora do escopo deste trabalho). Além disso, tais redes irão implementar algum mecanismo de priorização "global", tipicamente baseado em eventos (*event-based*), usando qualquer tipo de abordagem para priorização de sensores [Costa et al. 2015, Costa e Guedes 2013]. Para cada uma das redes implantadas, um ou mais *sinks* móveis estarão disponíveis, sendo que tais *sinks* podem mover-se, exclusivamente, dentro da área de interesse da cidade definida por cada aplicação. Por fim, cidades serão modeladas como um conjunto de vias de trânsito, que serão por onde os *sinks* podem se movimentar, como apresentado na Figura 3.1.

Portanto, para o desenvolvimento desta pesquisa, o tráfego urbano é um tema considerado de extrema importância devido ao seu alto grau de influência na dinâmica de uma Cidade Inteligente. Um exemplo genérico do escopo de uma cidade é apresentado na Figura 3.1, apresentando vias, sensores e *sinks* móveis. Dessa forma, neste trabalho foi considerado a movimentação e posicionamento dos nós *sinks* móveis e variáveis como: comportamento e configuração das vias, posicionamento corrente dos *sinks* móveis, prioridade dos sensores e configurações de comunicação, com o objetivo de otimizar o posicionamento de *sinks* móveis em RSSF, orientada a eventos em um ambiente urbano.

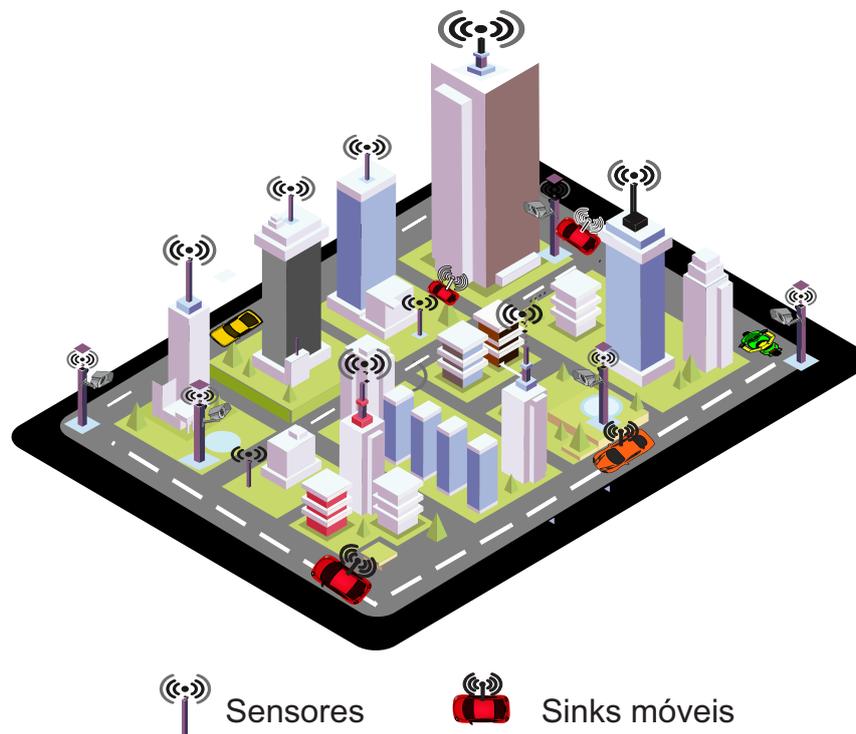


Figura 3.1: Exemplo de uma Cidade Inteligente com sensores para monitoramento.

3.2 Formalização do Problema

O problema que motivou o desenvolvimento deste trabalho está relacionado com a movimentação e otimização do posicionamento de nós *sink* em um ambiente urbano de uma Cidade Inteligente. Com o crescimento das cidades surgiu a necessidade da automatização de processos inerente à administração dos recursos urbanos e um dos principais meios para esta automatização é o desenvolvimento de redes de comunicação que possam responder com um desempenho aceitável a eventuais mudanças em um ambiente urbano [Yin et al. 2015]. Entretanto, o processo de transformação de uma “cidade genérica” em Cidade Inteligente motiva vários desafios relacionados à infraestrutura de comunicação. Considerando que sensores frequentemente possuem restrições de recursos, quanto mais otimizada for a localização dos nós *sinks*, potencialmente maior será o tempo de vida útil da rede [Peixoto e Costa 2017a]. Portanto, um problema a ser discutido neste trabalho está associado a encontrar as posições “ideais” para os *sinks* móveis da rede em um ambiente urbano. De fato, a otimização do posicionamento destes nós *sinks* torna-se uma tarefa desejada para a melhoria do desempenho da rede em ambientes urbanos. Em [Peixoto e Costa 2017a] foi discutida uma abordagem de RSSF para posicionar múltiplos *sinks* móveis, construindo uma valiosa solução para cenários de Cidades Inteligentes. A proposta considera a relevância de cada nó sensor na rede e algumas restrições de movimentação em uma cidade, dando uma importante contribuição para resolução do problema de posicionamento.

Em relação a movimentação dos nós *sinks*, o problema a ser elucidado está em selecionar um algoritmo de caminho mais curto que possa encontrar o caminho com menor custo entre dois pontos em um grafo que representará pontos de origem e destino em vias urbanas de uma Cidade Inteligente.

A mobilidade em uma RSSF passa por uma evolução natural. As redes tradicionais não suportam requisitos de mobilidade, porque os elementos da rede ficam próximos aos concentradores de rede [Achour et al. 2015]. Já em uma RSSF, a mobilidade é essencial para o seu desempenho. Porém, quando a conectividade é bidirecional, torna-se necessário um modelo de mobilidade mais elaborado, justificando a necessidade de propor melhorias em um algoritmo de posicionamento existente com o propósito de reduzir a latência e otimizar o tempo de reposicionamento dos dispositivos na rede urbana. Um dos problemas é o fato de grandes cidades enfrentarem um aumento no congestionamento do trânsito, ocasionando um obstáculo para qualquer cidade que queira modernizar-se para se transformar em uma Cidade Inteligente. O crescimento sem precedentes das cidades de hoje, juntamente com o aumento da mobilidade e da população estão trazendo crescimento desordenado no número de veículos nas ruas e avenidas das cidades e este desenvolvimento levou a novos desafios para implantação de redes de comunicação em uma cidade [Masek et al. 2016], como mostra a Figura 3.1.

O tráfego urbano é uma questão importante em Cidades Inteligentes. Sistemas de tráfego urbano são muitas vezes o mais importante dos sistemas existentes de uma cidade. Dispositivos implantados em transporte público, veículos particulares, sistemas de GPS,

câmeras de trânsito e outros sensores pode gravar uma grande quantidade de comportamento do tráfego e fornecer uma quantidade significativa de dados [Yin et al. 2015]. Portanto, sistemas e tecnologias implantadas nas cidades tornam-se incapazes de adquirir parâmetros de tráfego precisos e pode perder dados de tempo real, como acidentes que ocorrem ao longo das vias.

Para [Masek et al. 2016], os sistemas de gestão de tráfego oferecem recursos que podem ser utilizados para reduzir o congestionamento rodoviário e melhorar os tempos de resposta em caso de acidentes de carro. No entanto, a implantação de redes de sensores em ambientes rodoviários enfrenta diversos desafios a serem investigados. Desta forma, o ambiente urbano definido neste trabalho considera que uma determinada cidade genérica irá implantar uma ou mais redes de sensores sem fio. De fato, o tráfego urbano é um tema considerado de extrema importância devido ao seu alto grau de influência na dinâmica de uma Cidade Inteligente, o problema agravou-se nas últimas décadas graças à concentração de pessoas nas cidades, e à falta de planejamento urbano.

Um outro desafio referente as redes de sensores urbanas é a forma com que os sensores trocam informações. Neste contexto, as redes podem ser classificadas como "proativa" ou "reativa" [Martincic e Schwiebert 2006]. Nas redes proativas, os sensores trocam informações periodicamente. Nas redes reativas, as trocas de informações só ocorrem quando determinados eventos acontecem. Algumas aplicações para as RSSF podem ser caracterizadas como reativa, ou seja, sensores executam uma ação quando é detectado uma mudança nas leituras do dispositivo, atribuído a um evento específico. Os nós podem ser programados para enviar uma mensagem de notificação sempre que suas leituras de sensores ultrapassem um limite predefinido. A mensagem é encaminhada para o *sink* para posterior processamento. O *sink* pode emitir uma consulta com uma condição que é distribuído e armazenado localmente em cada nó sensor. Quando a condição é atendida, o nó transmite uma mensagem que é encaminhada para o *sink* indicando que o evento foi detectado [Martincic e Schwiebert 2006]. A detecção de evento é um dos principais recursos das redes de sensores sem fio e podem monitorar fenômenos relacionados a diversas aplicações em uma cidade. Um exemplo é a área da saúde, implantada para detectar comportamentos anormais dos pacientes, outro exemplo é a detecção de fogo, implantada para configurar um alarme se um incêndio começar em algum lugar da área monitorada [Martincic e Schwiebert 2006]. Independentemente da aplicação específica, a rede deve ser capaz de detectar a ocorrência de determinados eventos. No entanto, diversos fenômenos não têm um significado real para um nó sensor. Assim, é necessário desenvolver técnicas adequadas que permitam descrever eventos de forma que os nós sensores possam "entender" o real significado do evento [Kapitanova et al. 2012]. De fato, o problema está relacionado com algumas aplicações que transmitem continuamente informações e as alterações nas leituras dos sensores de um nó não são necessariamente a indicação de que ocorreu um evento, ocasionando um falso positivo. Além disso, se a notificação de envio de nó sensor estiver localizada a uma longa distância do nó *sink* a sobrecarga referente a propagação das mensagens adicionais em toda a RSSF reduziria a vida útil da rede.

Portanto, o tráfego urbano e a forma como os eventos são detectados é um tema conside-

rado de extrema importância devido ao seu alto grau de influência na dinâmica de uma Cidade Inteligente. O problema agravou-se nas últimas décadas graças à concentração de pessoas nas cidades, e à falta de planejamento urbano.

3.3 Descrição das Simulações

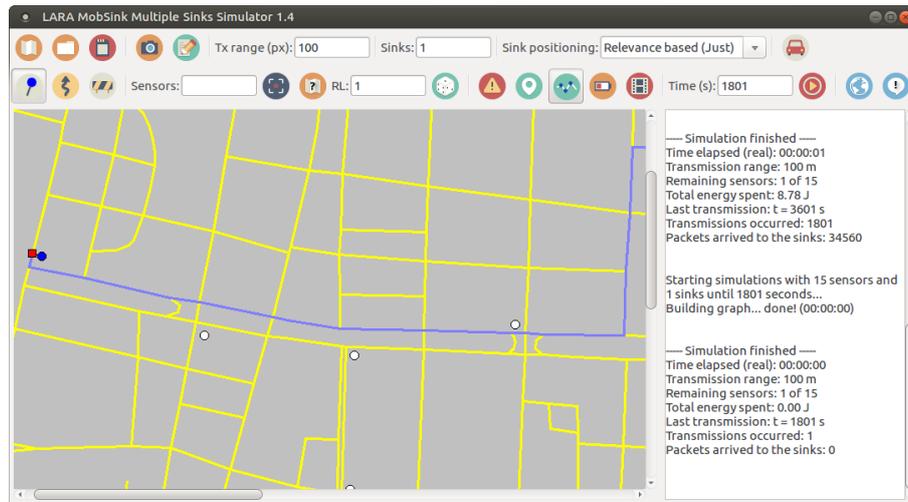
Para simular a movimentação dos *sinks* em vias urbanas foi usado neste trabalho uma extensão do simulador MobSink [Peixoto e Costa 2015a], que é um simulador de RSSF desenvolvido pelo grupo de pesquisa do LARA – Laboratório de Aplicações e Redes Avançadas, como mostra a Figura 3.2. O MobSink foi elaborado para implementar e otimizar algoritmos de mobilidade para múltiplos *sinks* móveis, simulando RSSF e novos algoritmos de posicionamento para múltiplos *sinks* móveis em ambientes urbanos, onde cada nó do sensor possuía um nível de relevância *RL* que é considerado ao calcular a posição ideal dos *sinks* móveis na RSSF. O objetivo foi reduzir o uso de energia priorizando nós mais importantes. Na nova versão do simulador MobSink desenvolvido para esta pesquisa foram feitas as seguintes modificações:

- controle e programação dos sensor para criar sensores dinâmicos que alteram os níveis de relevância ao longo do tempo;
- função para importação de mapas do OpenStreetMaps para criar simulações em cidades reais;
- controle de tráfego, para a simulação de engarrafamentos em um ambiente urbano e movimentação dos *sinks* móveis pela melhor rota;

Na abordagem proposta, foi simulado a mobilidade e o posicionamento dos nós *sinks* em um ambiente urbano, para isso foram usados mapas extraídos do *OpenStreetMap*, considerado por [Haklay e Weber 2008] como um projeto de mapeamento colaborativo para criar mapa livre e editável [Haklay e Weber 2008]. A abordagem ainda leva em conta dados de trânsito, relevância dos sensores e posicionamento otimizado dos nós *sink* em relação a um ambiente urbano de uma Cidade Inteligente, como mostra a Figura 3.2.

Para validar o trabalho, foi simulado, no MobSink, ambientes urbanos particularmente considerando ruas e avenidas da cidade de Feira de Santana, Ba, Brasil, onde os sensores foram espalhados de forma aleatória e inicialmente desligados. Para simular um evento na cidade, os sensores da RSSF ficaram ativos por um determinado tempo monitorando um evento na cidade. Para isso, foi alterado o nível de relevância dos sensores, simulando assim, o monitoramento de um evento. Os nós *sinks* tiveram que se mover pelas vias urbanas representadas por coordenadas geográficas, definidas no mapa, para a posição otimizada de monitoramento e coleta de dados dos sensores.

Os *sinks* móveis usaram as coordenadas geográficas extraídas do OpenStreetMap para se moverem pelas vias da cidade e os níveis de relevância dos *sinks* para otimizar o posicionamento na rede.

Figura 3.2: MobSink - Simulador para múltiplos *sinks*.

3.4 Etapas da Metodologia de Pesquisa

Para alcançar os objetivos propostos neste trabalho, foram executados, inicialmente, algumas atividades de pesquisa exploratória qualitativa. Ou seja, foi realizado um estudo inicial dos objetivos da pesquisa, buscando desta forma, conhecimento sobre o fenômeno a ser investigado. Diante disso, foi necessário revisar artigos científicos para realizar uma análise da literatura e construir o referencial teórico, mostrando o surgimento de novas aplicabilidades referentes a estas redes. Além disso, investigou-se novas metodologias de posicionamento de múltiplos *sinks* móveis com ênfase na otimização geral da rede de sensores em um ambiente urbano, considerando as especificações de trânsito nas cidades modernas. Por fim, foram investigadas as soluções para construção de *clusters* de comunicação e como o posicionamento dos *sinks* móveis pode ser otimizado considerando essa particularidade.

Uma RSSF possui diversas restrições de posicionamento e recursos, ocasionando vários problemas em relação ao consumo de energia, alcance na comunicação, baixa largura de banda, processamento e armazenamento em cada nó. Além disso, o ambiente em que a rede foi instalada exerceu um papel fundamental na determinação do tamanho da rede, no modelo de implantação e na sua topologia [Yick et al. 2008]. Diante disso, foi necessário analisar as melhores metodologias de posicionamento do *sinks* móvel em uma topologia, onde foi levado em conta o método de priorização usado para otimizar a transferência de dados em uma RSSF. Por fim, foi analisado de forma experimental a solução proposta para o posicionamento de *sinks*, usando simuladores de rede, com o intuito de validar o objeto de estudo.

Capítulo 4

Solução Proposta

Este capítulo apresenta a abordagem proposta descrevendo seu funcionamento, objetos de otimização, modelos matemáticos usados e algoritmo proposto. Serão abordados conceitos referente ao tráfego de Cidades Inteligentes. ’

4.1 Conceitos Fundamentais

Para [Yin et al. 2015] a revolução das TICs ofereceu aos cidadãos a oportunidade de minimizar ou até resolver problemas relacionados à urbanização. A cada ano, as cidades se tornam mais digitais e as pessoas tem mais acesso à informação, à economia, à cultura, aos transportes, ao entretenimento e a diversos aspectos urbanos. É neste cenário que este trabalho propõe otimizar a arquitetura de posicionamento e a movimentação de múltiplos *sinks* em redes de sensores sem fio, considerando as principais restrições de comunicação detectada em um ambiente urbano de uma Cidade Inteligente. A Figura 4.1 ilustra o posicionamentos de sensores e *sinks* móveis trasportados por automóveis para o monitoramento de grupos de sensores em uma pequena área de uma cidade.

De um modo geral, identificar os requisitos fundamentais relacionados às melhorias em algoritmos e modelos matemáticos que reduzam o tempo, a distância e identifiquem com precisão os principais obstáculos de um ambiente urbano são de grande valor no que diz respeito à otimização dos *sinks* móveis em uma RSSF. Entretanto, é de grande importância que os algoritmos possam identificar outras restrições relacionadas ao ambiente urbano das cidades, tais como o nível de fluidez do tráfego de veículos nas principais vias das cidades, obstáculos urbanos que podem atrapalhar a movimentação e posicionamento do *sinks*, sentido permitido das vias, horários de pico, etc.



Figura 4.1: Modelo proposto.

É neste contexto que a solução proposta de otimização tem o objetivo de definir o posicionamento de múltiplos *sinks* e traçar a melhor rota para que um ou mais *sinks* se movimentem do ponto de origem até a posição de destino, levando em consideração restrições de movimentação no ambiente, como mostra a Figura 4.1.

O modelo proposto é subdividido em três etapas, conforme apresentado na Figura 4.2. Os parâmetros da simulação, como posição atual dos *sinks*, configuração das vias, tráfego corrente e configuração dos sensores entram no modelo. A partir desses parâmetros, são definidas as novas posições que os *sinks* devem assumir, às quais é feito um ajuste fino, reposicionando os *sinks* em situações onde a nova posição calculada é uma zona proibida (local de estacionamento proibido, por exemplo). Por último, com as novas posições já definidas, o modelo proposto neste trabalho efetua o cálculo das rotas a serem seguidas pelos *sinks* considerando os parâmetros de entrada relacionados ao tráfego nas vias e suas configurações.

Cada etapa deste processo está bem definida. As próximas seções apresentam as definições básicas consideradas.

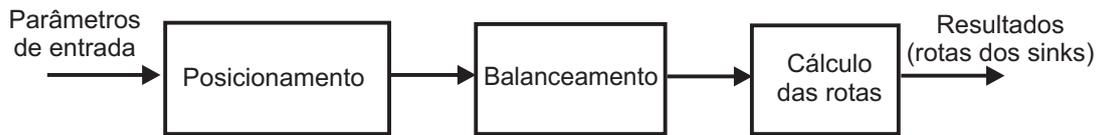


Figura 4.2: Fluxograma do processo de obtenção de rotas.

4.1.1 Definições Básicas

O objetivo fundamental de uma rede de sensores é produzir, durante certo período de tempo, informação significativa a partir de dados locais, obtidos por sensores que transmitem pacotes pela rede. De forma simplificada, será considerado um nó sensor como um componente de *hardware* capaz de monitorar ambientes urbanos e organizados em grupos de sensores, com o objetivo de transmitir informações através de *sinks*, para um sistema computacional de monitoramento. Essas e outras definições básicas necessárias para a realização deste trabalho são apresentadas as seguir:

- **Nó sensor:** um nó sensor é uma unidade básica da rede de sensores. É normalmente composto de unidade de sensoriamento, processador, memória, interface de comunicação sem fio e fonte de alimentação. No cenário proposto, sensores estarão distribuídos em setores estratégicos (de acordo com os requisitos da aplicação), para o monitoramento de fenômenos no ambiente urbano. Um sensor pode ser representado por um ponto s do plano cartesiano onde fica associado a um par (x, y) , onde $s = (x, y)$ em uma modelagem 2D, representado por coordenadas geográficas (latitude e longitude) do local da implantação. Um sensor estará posicionado em um ponto fixo da cidade, junto a um *cluster* de sensores.
- **Nó *sink*:** no cenário considerado neste trabalho, um nó *sink* tem o objetivo de coletar todas as informações transmitidas pelos grupos de sensores. Para isso, os *sinks* se deslocarão através das vias da cidade, quando algum reposicionamento for necessário, se posicionando o mais próximo possível de grupos de sensores com maior demanda de transmissão. A solução proposta considera múltiplos *sinks* que podem ser reposicionados a qualquer momento ao longo do tempo de vida da rede. Assim, espera-se que com um número maior de coletores, a probabilidade de isolamento de um grupos de nós sensores seja reduzida, aumentando o desempenho da rede.
- **Posicionamento do *sink*:** o posicionamento define os locais exatos de parada dos nós *sinks* em um ambiente urbano, dentro da área de cobertura de uma RSSF. Para o cenário proposto, encontrar a melhor estratégia para o posicionamento final destes coletores contribui para o desempenho nos projetos de comunicação com múltiplos *sinks*. Desta forma, um reposicionamento torna-se necessário toda vez que um fenômeno modifique a demanda de transmissão da RSSF, otimizando desta forma a comunicação com os grupos de sensores. Portanto, o nó *sink*, quando reposicionado, será fixado próximo aos *cluster* de sensores aumentando o tempo de vida útil dos mesmos. Desta forma, o local de posicionamento de cada nó *sink* será definido

por um ponto de localização geográfica representado por p , associado às coordenadas geográficas (x, y) , onde será identificada por $p = (x, y)$. E estarão localizados ao longo da extensão da via. Assim, a posição de um nó *sink* representado pela variável p estará posicionado em um ponto fixo da cidade, junto a um *cluster* de sensores.

- **Via:** no âmbito de uma cidade, uma via urbana será representada por ruas e avenidas integrantes do sistema viário de uma cidade. Sendo, desta forma, representado por arestas em um grafo ponderado direcionado. Portanto, uma via possui peso e direção, atribuindo-se desta forma à via um sentido e um comprimento. Cada via possui ainda em tempo de deslocamento, que varia de acordo com o tráfego ponderado da via.
- **Tráfego urbano de veículos:** é representado pela quantidade de veículos passando por uma via em uma cidade. O nível do tráfego é medido pelo fluxo de veículos na via, representado pela taxa de passagem de veículos em um período de tempo. Essa informação define o "peso" de cada via.
- **Deslocamento:** no cenário proposto o deslocamento está descrito como a variação na posição (caminho) de um nó *sink* em uma RSSF implantada em um Cidade Inteligente. Neste contexto, o deslocamento envolve um vetor ordenado de vias. Portanto, o deslocamento em um vetor significa que ele tem uma direção estabelecida. Levando em consideração: comprimento, fluxo de veículos e tempo de deslocamento, que varia de acordo com o tráfego da via.

Em uma cidade, vários fatores urbanos podem interferir no deslocamento e posicionamento do nó *sink*. Desta forma, definir o posicionamento preciso é de fundamental importância para a comunicação dos sensores no contexto de uma cidade. Portanto, este trabalho pretende calcular a melhor rota para deslocar e posicionar os *sinks* móveis toda vez que ocorrer um fenômeno capaz de alterar o nível de prioridade nos *clusters* de sensores, definindo desta forma, a melhor posição final do *sink*. Uma maneira de resolver este problema é usar modelos matemáticos na produção de algoritmos capazes de identificar a melhor configuração para a rede em um determinado momento. Para o desenvolvimento do algoritmo, os dispositivos de rede serão modelados usando pontos em um plano cartesiano, associados às coordenadas dos *sinks*. Com estes dados, será possível traçar o ponto entre a origem e o destino do *sink*, fornecendo informações para o posicionamento do *sink*. Com isso, pretende-se definir o melhor caminho para o deslocamento entre um local de origem e o destino que se quer chegar, possibilitando uma economia de recursos relevantes para a rede.

Um outro fator importante para o posicionamento de cada *sink* móvel é o nível de relevância. Para isso, [Costa et al. 2014] atribui aos grupos de sensores um nível de relevância denominado *RL Relevance Level*. Este nível poderá variar de acordo com o fenômeno urbano que o grupo de sensores estiver monitorando no momento. Ao iniciar uma rede, os *sinks* são posicionados em pontos centrais da área de cobertura dos sensores e a ocorrência de um evento pode desencadear um aumento do tráfego de dados e, conseqüentemente, a necessidade de comunicação em tempo real. Portanto, o *sink*, de forma autônoma, é

direcionado para uma nova posição de acordo com a estado da rede. Para cada sensor ou grupo de sensores será atribuído um nível de relevância RL e a rede iniciará com todos os sensores no nível de relevância $RL = 0$. Durante o tempo de vida útil da rede os níveis de relevância podem variar dependendo do tipo de ocorrência que o sensor ou grupo de sensores estiverem monitorando na cidade inteligente, podendo chegar até o nível $RL = 15$, para os setores do ambiente urbano com maior necessidade desempenho e tempo de vida útil da rede [Peixoto e Costa 2017a, Collotta et al. 2018]. Neste trabalho, os cálculos e definições dos valores de RL estão fora do escopo da pesquisa realizada, e espera-se que esses valores sejam atribuídos por algum sistema externo.

Com o objetivo de determinar uma rota entre um endereço de origem e um endereço de destino (latitude e longitude de uma posição do *sink* no ambiente urbano) em uma cidade, a solução aqui apresentada utiliza o algoritmo Dijkstra para o cálculo com menor custo em um grafo [Zhang et al. 2016]. Em uma RSSF, os sensores agrupados irão enviar dados para os *sink* que se deslocarão por ruas e avenidas buscando o melhor posicionamento para a transmissão de dados com os grupos de sensores. Para isto, os caminhos vão ser representados pelas arestas de um grafo e os pontos de posicionamento, em relação aos *clusters* de sensores, estarão distribuído ao longo da via, em um ponto de posicionamento na aresta, como mostra o exemplo apresentado na Figura 4.3.

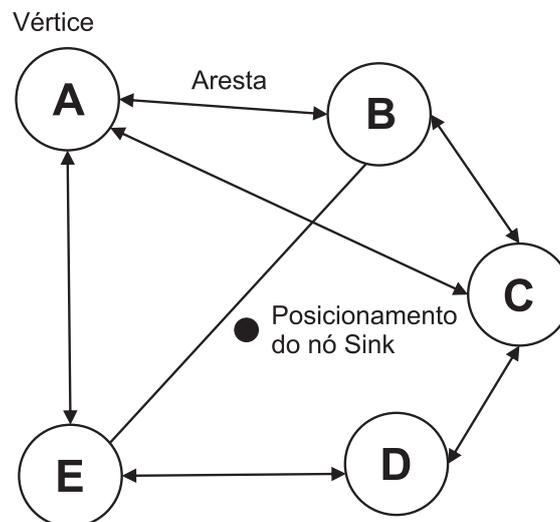


Figura 4.3: Modelo de um grafo com posicionamento do nó *sink*.

A seguir são descritos os elementos que compõem o formalismo utilizado para o cálculo do deslocamentos dos *sinks*, que é um dos objetivos finais do trabalho proposto:

- Vértice: os vértices são representados pelos pontos das extremidades das vias em uma cidade. Um *sink* se deslocará entre os *clusters* de sensores usando coordenadas

(x, y) passando pelos vértices do grafo. Assim, um caminho C de comprimento n de um vértice v a um vértice v_n em um grafo $G = (V, A)$, pode ser definido como uma sequência de vértices, que neste contexto, representará um caminho a ser percorrido pelo nó *sink*.

$$C = (v_0, v_1, v_2, \dots, v_n) \quad (4.1)$$

- **Aresta:** as arestas representam as vias entre dois vértices v_0 e v_n , ou seja, representa uma via no contexto de uma cidade. Cada aresta está associada a um par de vértices. Em um grafo ponderado, um peso é associado à cada via e o peso de cada via representa o tempo para percorrê-la no contexto deste trabalho representado por $P(a_0, a_n)$ (peso da aresta que conecta o vértice v_0 ao vértice v_n). Um caminho entre a_0 e a_n é a lista $(a_0, a_1, a_2, \dots, a_n)$. Um caminho representa o deslocamento do *sink* entre as posições dos grupos de sensores por vias urbanas de uma cidade. Desta forma um conjunto de arestas é representado na Equação 4.2.

$$A = (a_0, a_1, a_2, \dots, a_n) \quad (4.2)$$

- **Grafo:** um grafo $G = (V, A)$ é um conjunto de vértices (V) e arestas (A). No contexto deste trabalho, as arestas representam vias em uma cidade enquanto que os vértices representam os pontos de conexão entre as vias. O algoritmo *Dijkstra* percorre o grafo para traçar a rota de movimentação do *sink* na ocorrência de um evento na RSSF. Em caso de vias de mão dupla, arestas direcionadas são utilizadas no grafo. A Figura 4.3 ilustra um grafo com indicação de uma possível localização de um *sink*.

Para definir o peso do deslocamento, o algoritmo *Dijkstra* calcula a distância entre dois pontos x e y sobre uma via real que é dada pelo valor da diferença entre as coordenadas dos pontos. Considerando que x tem uma coordenada e y tem outra coordenada, então a distância entre x e y , será definida como $P(x, y)$. Portanto, neste caso foi utilizado para definir o peso da via, a distância euclidiana entre os dois pontos de deslocamento do nó *sink*. Considerando que os pontos $x = (x_1, y_1)$ e $b = (x_2, y_2)$. A distância Euclidiana entre x e y é o comprimento entre os pontos x e y . usando as coordenadas geográficas dos sensores, a distância $P(x, y)$ é representada pela Equação 4.3.

$$P(x, y) = xy = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (4.3)$$

Em um grafo ponderado, o comprimento do caminho C , que também pode ser chamado de peso do caminho, $P(C)$, é representado pela soma dos pesos das vias mostrada na Equação 4.5. O caminho mais curto é representado pelo somatório dos pesos do caminho, como mostra a Equação 4.4. Assim, o caminho formado pelo menor peso no grafo é o caminho mais curto.

$$P(C) = \sum_{i=1}^n P(a_{i-1}, a_i) \quad (4.4)$$

Para determinar o caminho mais rápido, com o propósito de otimizar o deslocamento do nó *sink* na cidade considerada, a solução proposta usará o algoritmo *Dijkstra* [Zhang et al. 2016]. Os caminhos criados pelo algoritmo *Dijkstra* geralmente mostram um menor número de conexões, o que diminui o atraso médio da disseminação de mensagens na rede [Musznicki et al. 2012]. Para isso, o algoritmo identifica, a partir de um vértice do grafo, qual é o custo mínimo entre esse vértice e todos os outros do grafo. Portanto, esta solução usará como base o algoritmo *Dijkstra*, para estabelecer o caminho mais curto entre dois pontos a e b de um grafo com pesos nas vias.

$$s(v, v) = \begin{cases} \min\{P(a) : v_a \sim^c v_n\} \\ \infty \end{cases} \quad (4.5)$$

Os pesos serão representados por um vetor de adjacências onde os índices do vetor correspondem aos vértices do grafo e o armazenamento no vetor será os pesos das vias representado pelas arestas do grafo. Desta forma, partimos do vértice v , e vamos incorporando passo a passo outros vértices até chegar em v_n , em cada passo, teremos um subgrafo do grafo, constituído por vértices e arestas. Este subgrafo é constituído de forma que ele forneça um caminho mais curto de v até v_n , em cada um dos vértices do subgrafo. Desta forma, para calcular o caminho mais curto entre dois vértices será usado a Equação 4.5, que representa matematicamente o algoritmo utilizado nesta abordagem. Desta forma, em um grafo não direcionado o nó *sink* poderá se deslocar pela cidade usando a rota que estiver com menor custo atribuído às vias da cidade. Entretanto, o caminho mais curto não significa o caminho mais rápido entre um percurso no ambiente urbano por conta dos níveis de congestionamento das vias e é neste ponto que esta abordagem pretende alterar a forma com que os pesos são definidos no algoritmo *Dijkstra*.

4.1.2 Abordagem Proposta

A abordagem proposta pretende otimizar o caminho para um *sink* móvel em uma Cidade Inteligente, a fim de fazer que o *sink* mova-se para sua posição final o mais rápido possível. Embora *Dijkstra* possa calcular a rota mais curta da cidade [Musznicki et al. 2012], não é suficiente por causa da diversidade de mudanças que podem haver no trânsito de uma cidade. Assim, esta abordagem otimiza o algoritmo usando informações de tráfego de modo que o *sink* possa se mover rapidamente. Dito isto, a distância euclidiana não é suficiente para mensurar, quando se trata de vias urbanas. Portanto, apenas aplicar *Dijkstra* não resolverá o problema.

Uma das preocupações sobre como encontrar uma rota em uma cidade é conhecer o estado do trânsito no momento da condução. É muito comum obter um caminho mais longo em

tempo de corrida se quisermos chegar mais rápido. Mesmo que haja rotas mais curtas, elas podem estar congestionadas, fazendo com que o motorista demore mais tempo para chegar ao seu destino final. Então, usar apenas o comprimento das estradas não nos dá uma solução para a melhor rota em um engarrafamento.

O trabalho em [Narayanan et al. 2015] apresenta um método para o cálculo da estimativa de tempo de viagem em uma cidade em um projeto de RSSF para Cidades Inteligentes no qual é desejado fazer a movimentação do *sinks* através das vias urbanas. O método apresentado em [Narayanan et al. 2015] define uma matriz $A_{m \times n}$ contendo a velocidade média dos veículos nas vias em determinados intervalos de tempo. Cada célula $a_{i,j}$ indica a velocidade média que os veículos levam para percorrer a via s_j no tempo $(t_i - \Delta t, t_i)$.

A partir dessa informação, uma segunda matriz $B_{m \times n}$, tal que $b_{i,j} = \frac{d_j}{a_{i,j}}$, sendo d_j o comprimento da via s_j . O método histórico apresentado em [Narayanan et al. 2015], define como o tempo médio de viagem em uma rota l a partir do tempo i a soma de todos os tempos médios dos trechos que compõem a rota, como mostrado na Equação 4.6, onde k_j é o número de instantes de tempo que é decorrido até que o condutor entre no j -ésimo trecho da rota (o valor de k_j pode ser obtido a partir do tempo decorrido nos trechos anteriores de cada iteração).

$$T_l^i = \sum_{j=1}^l b_{i+k_j, j} \quad (4.6)$$

A abordagem proposta, combina o comprimento da via e a velocidade máxima da viagem nas vias, para determinar o tempo total do percurso do *sink*. Para definir a velocidade de viagem em uma via, foi usado o limite de velocidade como valor padrão em uma boa situação de trânsito. Enquanto o tráfego for mais pesado, a velocidade dessa estrada diminui, sendo 0 Km/h a velocidade da estrada quando o tráfego é completamente interrompido. Para obter um valor de velocidade para uma via, consideramos seu limite de velocidade $V_{max}(i)$ e um multiplicador de tráfego $m(i)$ que vai de 0 (tráfego interrompido) para 1 (estrada livre). O tempo total da viagem t em uma via i foi definido usando a Equação 4.7.

$$T(i) = \lim_{x \rightarrow m(i)} \frac{S(i)}{V_{max}(i) \cdot x}, 0 \leq m(i) \leq 1 \quad (4.7)$$

- $T(i)$: Tempo para percorrer a via i
- $S(i)$: Comprimento da via i
- $m(i)$: Função trânsito, indicando a situação do congestionamento na via i (0 = trânsito parado, 1 = trânsito livre)
- $V_{max}(i)$: Velocidade máxima da via i
- x : Variável que define o nível de congestionamento

Para calcular a rota mais rápida, o algoritmo modela cada via como uma aresta e cada cruzamento como um vértice em um grafo direcionado, o peso de cada ponta é o seu tempo de viagem, calculado usando a Equação 4.7. Tendo cada tempo de viagem da via como o peso das arestas, assim, podemos usar o algoritmo *Dijkstra* para determinar o caminho otimizado para uma Cidade Inteligente. Desta forma, o tempo total da viagem do *sink* pode ser calculado a partir da soma dos tempos de viagem de cada via que o *sink* percorre, como mostrado na Equação 4.8. A abordagem, considera um conjunto $C = (a_0, a_1, a_2, \dots, a_{n-1})$ contendo todas as arestas do grafo que representam o caminho que o *sink* usará para se mover para o seu destino final. O peso de cada aresta em P pode ser obtida lendo o atributo de cada vértice depois de executar *Dijkstra*, desde o vértice de destino até a origem. O tempo total de viagem T do *sink* é então calculado usando a Equação 4.8.

$$T = \sum_{i=0}^{n-1} T(a_i) \quad (4.8)$$

Portanto em um grafo ponderado, o algoritmo *Dijkstra* pode ser usado para encontrar o caminho mais rápido entre dois vértices [Zhang et al. 2016]. Usando para isso o valor numérico do peso das arestas para executar esse cálculo. Assim, o tempo total da viagem percorrido nas vias passa a ser usado como peso nas arestas em um grafo. Desta forma, é possível usar o algoritmo *Dijkstra* para calcular o caminho em que a soma do tempo de viagem de cada via é o mínimo, ou seja, a rota mais rápida no grafo.

No modelo proposto, foi usado variáveis de otimização para a movimentação e posicionamento dos *sinks* móveis. Todos os parâmetros estão relacionados com a infraestrutura de uma Cidade Inteligente. Portanto, o novo modelo pretende estabelecer uma nova metodologia de comunicação para uma RSSF, aplicado a um ambiente urbano. Assim, será usado no desenvolvimento da pesquisa os seguintes parâmetros:

- Fluxo de veículos em uma via urbana: representa neste contexto o número de veículos que cruzam uma determinada via, considerada dentro de um dado intervalo de tempo. Com essa informação pode-se definir o nível de congestionamento das vias, representado neste trabalho como uma condição em que os automóveis guiam a variadas velocidades ou paradas frequentes durante o percurso percorrido em, ruas ou avenidas;
- Configuração da via: caracterizado como mão e contramão das vias, obstáculos e interrupções periódicas do fluxo, causadas por acidentes, desvios ou obstrução das vias; característicos de um ambiente urbano;

Desta forma, a fim de obter o tempo instantâneo do trânsito, foi considerando o fluxo do tráfego representado pela número de veículos que cruzam uma determinada via dentro de um dado intervalo de tempo. Portanto, para definir a velocidade de deslocamento em uma via, foi considerar o fluxo e a densidade do tráfego [van Wee et al. 2013]. O tempo t é então obtida pela relação de fluxo (q) e densidade (k) usando Equação 4.9.

$$t = \frac{q}{k} \quad (4.9)$$

4.1.3 Algoritmo Proposto

No modelo proposto neste trabalho, o algoritmo deverá calcular, quando solicitado, a melhor rota para o deslocamento e posicionamento otimizado do nó *sink*, usando para isso dados do tráfego de veículos, localizações geográficas, dados de configurações das vias extraídos do ambiente urbano, como mostra a Figura 4.3. Desta forma, a comunicação dos grupos de sensores com o *sink* móvel será otimizada para a aplicação em uma Cidade Inteligente. Como base para o desenvolvimento do algoritmo, será usado *Dijkstra*, para identificar o caminho com menor custo da origem até o posicionamento final do *sink* na cidade. Entretanto, para o otimização em um ambiente urbano, será inserido parâmetros referentes às configurações das vias e tráfego das cidades, para que o algoritmo afira estes parâmetros e possa modificar o caminho, determinando o deslocamento com menor custo para a rede, como mostra o Algoritmo 1.

Algorithm 1 Algoritmo para Movimentação de Múltiplos *Sinks* (MSMA).

Require: V : Conjunto de vértice

A : Conjunto de aresta

v : Vértice de origem

t : Vértice de destino

Ensure: P : Pilha de aresta

{Calcular o tempo para atravessar cada aresta}

1: **for** $a \in A$ **do**

2: $a.weight \leftarrow \lim_{x \rightarrow m(a)} \frac{a.lenght}{a.vmax \cdot x}$

3: **end for**

{mover o *sink* de acordo com [Peixoto e Costa 2017b]}

4: MOVESINK()

{Executa *Dijkstra* usando tempo como pesos}

5: DIJKSTRA(V, A, v)

{Criar uma pilha de arestas}

6: $P.push(t)$

7: $cur \leftarrow t.prev$

8: **while** $cur \neq NULL$ **do**

9: $P.push(cur)$

10: $cur \leftarrow cur.prev$

11: **end while**

12: **return** P

Em um grafo ponderado comum, o algoritmo *Dijkstra* pode ser usado para calcular o caminho otimizado entre dois vértices [Zhang et al. 2016]. Deste modo, o abordagem proposta usa o valor numérico do peso das bordas para executar esse cálculo, usando para

isso o tempo de viagem nas vias, como peso nas bordas de um grafo. Assim é possível usar o (MSMA) Algoritmo para Movimentação de Múltiplos *Sinks*, representado pelo Algoritmo 1, para calcular o caminho em que a soma dos tempos de viagem de cada borda é o mínimo, ou seja, a rota mais rápida no grafo. Para calcular a rota mais rápida em uma Cidade Inteligente, o algoritmo MSMA, modela cada estrada como uma via em um grafo direcionado. O peso de cada ponta é o seu tempo de viagem calculado usando Equação 4.7. Tendo cada tempo de viagem na estrada como o peso da via e fazendo o somatório dos pesos, como mostra a Equação 4.8, o algoritmo MSMA é usado para informar o caminho mais rápido entre os pontos (x, y) . Desta forma, para cada uma das redes implantadas um ou mais *sinks* móveis estarão disponíveis dentro da área de interesse na cidade.

O Algoritmo 1, percorre todas as arestas do grafo e permite que estas possuam pesos diferentes. Entretanto, na abordagem proposta, inicialmente cada aresta recebe um peso baseado nos valores obtidos na Equação 4.8, onde é realizado um somatório para definir o tempo total a ser percorrido pelo *sink*, podendo ser calculado a partir da soma dos tempos de viagem percorridos em cada via, consideramos um conjunto $A = (a_0, a_1, a_2, \dots, a_{n-1})$ contendo todas as arestas do grafo que representam o caminho que o *sink* usará para se mover para o seu destino final. Desta forma, o tempo total da viagem pode ser obtido usando a Equação 4.8, que representa o valor do tempo total percorrido, o que possibilita o Algoritmo 1 encontrar o caminho mais rápido entre a posição anterior do *sink* e sua nova posição como mostra a Figura 4.4.

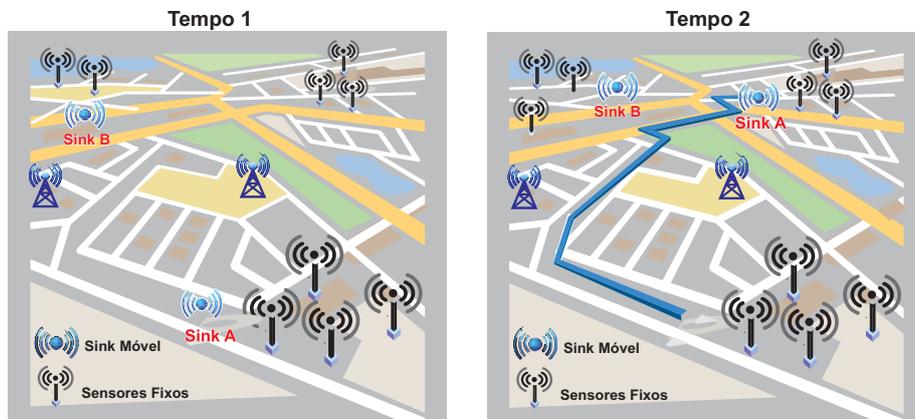


Figura 4.4: Exemplo do resultado do algoritmo (MSMA).

O fluxograma da Figura 4.5, mostra de forma genérica os passos do algoritmo para determinar o deslocamento e posicionamento do nó *sink* usando dados como: comportamento e configuração das vias, posicionamento corrente dos *sinks* móveis, prioridade dos sensores e configurações de comunicação do ambiente urbano, com o objetivo de otimizar o posicionamento dos *sinks* móveis em uma RSSF orientada a eventos. Inicialmente é carregado as variáveis e dados das configurações de trânsito, logo após é definido os conjuntos de rota viáveis e posicionamento inicial dos *sink* móveis, se ocorrer um evento

na cidade a relevância dos sensores passa para o nível de $RL = 15$, então o algoritmo MSMA determina qual o caminho mais rápido para o deslocamento e posicionamento do *sink*.

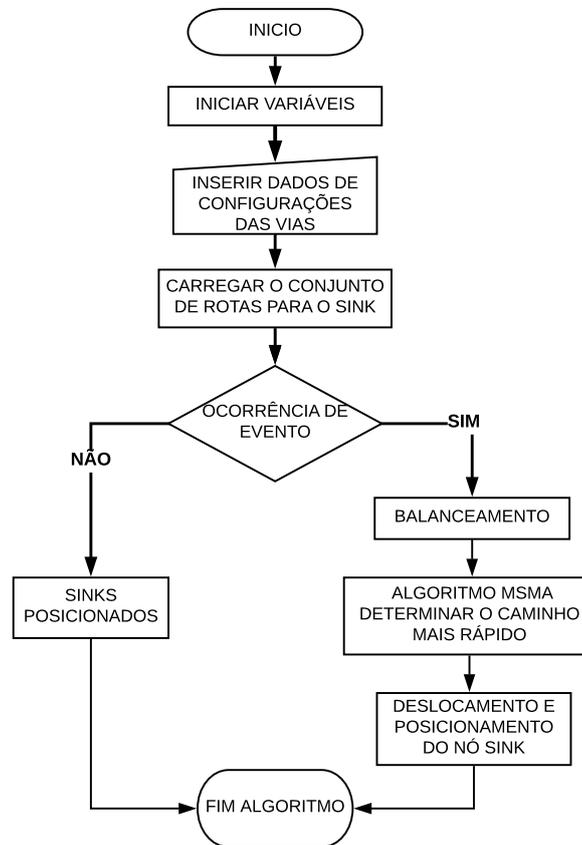


Figura 4.5: Fluxograma do algoritmo proposto.

4.1.4 Posicionamento dos *sinks*

O posicionamento de múltiplos *sink* móveis é feito com o propósito de capturar dados dos sensores de forma otimizada na RSSF posicionando o *sink* no ponto central do grupo de sensores que irá monitorar um evento na Cidade Inteligente. A abordagem proposta utiliza o algoritmo *k-means* [Hartigan e Wong 1979], usado para efetuar o posicionamento otimizado de múltiplos *sinks* em uma RSSF de uma Cidade Inteligente por [Peixoto e Costa 2015b]. Esta abordagem considera os níveis de relevância (RL) dos sensores para posicionar de forma otimizada múltiplos *sinks*, calculando a posição ideal de cada *sink* móvel da rede. Em uma RSSF cada *sink* fica responsável pela coleta de dados de um grupo de sensores na rede [Peixoto e Costa 2015b].

Para [Peixoto e Costa 2015b] o método de setorização *k-means* com relevância, define a posição ótima dos *sinks*, considerando para isso os níveis de relevância dos sensores em

relação aos grupos de sensores a ser monitorado em uma Cidade Inteligente. O algoritmo é executado em cada setor, definindo a posição otimizada do *sink*, após o posicionamento, o *sink* passa a receber dados apenas dos sensores pertencente ao setor de monitoramento que ocorreu o evento e elevou a prioridade dos sensores para $RL = 15$. Os *sinks* permanecem posicionados até ocorrer uma nova mudança no evento que altere o nível de relevância dos sensores e atualize para novas posições na rede. A atualização consiste em recompor a rede, mudando o *sink* para outro grupo de sensores se necessário, como mostra a Figura 4.6.

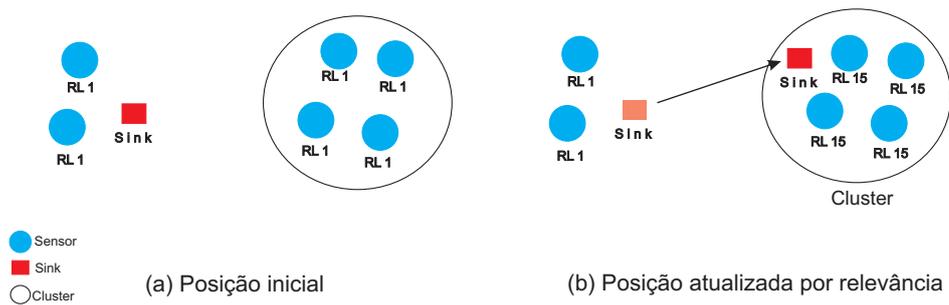


Figura 4.6: Posicionamento do *sink*.

O algoritmo *k-means* considera o nível de relevância dos sensores que fazem parte no mesmo grupo de sensores do *sink*, para manter-se próximo aos nós mais relevantes. Para isso o algoritmo usa as coordenadas de cada sensor multiplicado pelo seu nível de relevância. Assim é possível mover o *sink* para a posição mais adequada.

Capítulo 5

Resultados Experimentais

Após as simulações do algoritmo na extensão do simulador MobSink, espera-se que seja encontrado o caminho e o posicionamento otimizado para os *sinks* móveis em uma RSSF de uma Cidade Inteligente, a fim de fazer com que os *sinks* movam-se para sua posição final o mais rápido possível. Para isso, foram realizados testes no simulador, com o propósito de comparar com abordagens de *sinks* móveis que usam apenas a rota mais curta para o deslocamento e posicionamento. Com isso, espera-se uma otimização do tempo percorrido do *sink* e uma melhora na taxa de transferência de dados em uma RSSF.

5.1 Preparação dos cenários de testes

A fim de validar a abordagem proposta neste trabalho, foi preparado um cenário de um ambiente urbano, para simular uma rede de sensores sem fio adaptáveis, concebida para a detecção e monitoramento de eventos críticos ocorridos durante um determinado tempo. Para isso, foi modelado um mapa da região central da cidade de Feira de Santana, Bahia, como mostra a Figura 5.1, a fim de fazer com que os *sinks* selecionados movam-se pelas ruas e avenidas da cidade considerada toda vez que ocorrer mudança de relevância nos sensores, simulando assim a ocorrência de um ou mais eventos no ambiente urbano.

Para a realização das simulações, foram espalhados sensores por várias regiões do mapa para o monitoramento de algumas regiões críticas na área central da cidade. Algumas condições foram executadas no simulador MobSink, para a realização da coleta de dados com múltiplos *sinks* móveis espalhados no ambiente. Foi monitorado o tempo de deslocamento, a partir da origem até a posição final, para cada *sink* considerado, com o propósito de verificar o caminho com menor custo possível entre os grupos de sensores, comparando com abordagens de *sinks* móveis que usam apenas a rota mais curta para o deslocamento e posicionamento. Durante as simulações foram analisadas as taxas de pacotes recebidos e descartados pelos nós *sinks* móveis instalados na rede, assim como o tempo de deslocamento e a distância percorrida da origem até o destino, nos cenários de monitoramento da região simulada.



Figura 5.1: Mapa da região simulada: cidade de Feira de Santana.

5.2 Cenário 1: Comportamento dos *sinks* em vias congestionadas

Para simular o comportamento dos nós *sinks* móveis, foram distribuídos de forma aleatória no centro de Feira de Santana um total de 15 sensores, posicionados em pontos com bastante movimentação de pedestres e numerados de $S1$ a $S15$. Esse posicionamento foi realizado de acordo com o esquema apresentado na Figura 5.2.

Cada sensor segue uma programação de atividades para simular eventos na cidade considerada. Nesse esquema, foi atribuído a cada sensor um nível de relevância. Para cada período de tempo, um valor de RL é simulado, como definido na Tabela 5.2, ou seja, naquele tempo específico, o sensor é ativado com o nível de relevância indicado, fazendo com que os *sinks* se movam. Um valor de relevância irá indicar se o sensor está ativo ou inativo, portanto, "transmitindo" ou "não transmitindo" dados.

Se a relevância do sensor estiver entre os níveis $RL = 1$ e $RL = 15$, significa que os sensores estarão transmitindo pacotes de dados. O objetivo da agenda de programação de atividades dos sensores é simular o comportamento dinâmico da rede, com ocorrência aleatória de eventos em uma cidade, como mostra a Tabela 5.2. Sensores foram programados para ligar e desligar de acordo com o que acontece no ambiente e os *sinks* precisam reagir a isto, mudando suas posições.

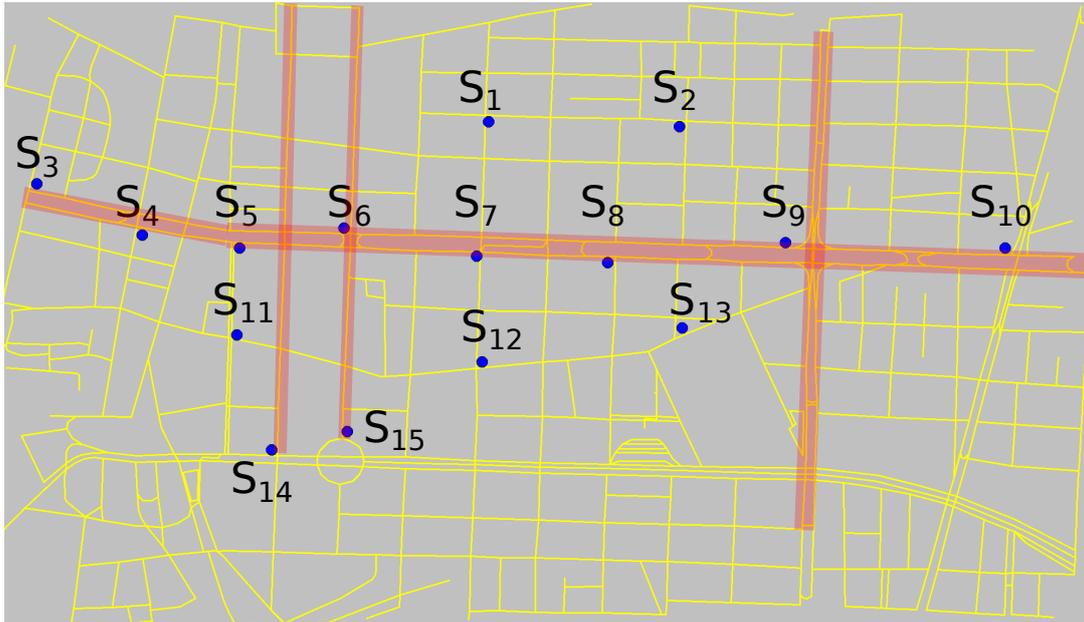


Figura 5.2: Sensores posicionados no mapa.

Nesse cenário de testes, as ruas destacadas em vermelho na Figura 5.2 ficam congestionadas das 11 às 19 horas, simulando engarrafamentos. Neste caso, os *sinks* terão que encontrar a rota mais rápida para alcançar os sensores. Os valores de intensidade de tráfego para o multiplicador $m(i)$, usado na Equação 4.7, foram definidos de acordo com a Tabela 5.1, simulando situações de trânsito intenso ($m(i) = 0.1$) e trânsito extremamente engarrafado ($m(i) = 0.01$). O objetivo neste cenário é validar o algoritmo em situações extremas de engarrafamento.

Tabela 5.1: Situação do congestionamento.

Horário	Valor	Situação
11 às 13h	0.01	Extremamente engarrafado
13 às 17h	0.1	Trânsito intenso
17 às 19h	0.01	Extremamente engarrafado

Para calcular as rotas mais rápidas, o MobSink usa a Equação 4.7 para determinar o tempo necessário que um *sink* irá se mover por cada via. As vias são modeladas como arestas em um grafo ponderado e as interseções das vias são modelados como vértices no grafo [Peixoto e Costa 2017b]. Tendo o tempo para se mover por cada via, o algoritmo proposto é usado para obter a rota mais rápida no mapa. Durante a viagem aos pontos de destino um *sink* não recebe nenhum PDU (*Packet Data Unit*), ou pacote dos nós sensores. Isso ocorre porque o evento desencadeia uma mudança de relevância nos sensores: o sensor próximo ao evento obtém grande relevância e os outros sensores reduzem para

Tabela 5.2: Programação de atividades dos sensores.

Time	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
05:00	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
05:30	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
06:00	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
07:00	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
08:00	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
09:00	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
10:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10
11:00	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
12:00	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
13:00	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
14:00	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
15:00	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16:00	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0
17:00	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
18:00	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
19:00	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0
20:00	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
21:00	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
21:30	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
22:00	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0

$RL = 0$ e hibernam. A tabela 5.2, demonstra a programação de atividades dos nós sensores, toda vez que que um nível de relevância muda para $RL = 5$ ou $RL = 10$ simula um evento na cidade.

Os sensores utilizados na simulação tem um tamanho de *buffer* (Memória física) de dados de 128 KB. Quando seus *buffers* estão cheios, novos pacotes provenientes de sensores próximos e novos pacotes gerados pelo próprio sensor são descartados. Assim, quanto mais tempo o *sink* leva para alcançar o sensor desejado, mais pacotes são descartados. Cada pacote na simulação contém 127 B de dados e cada sensor gera 20 pacotes por segundo.

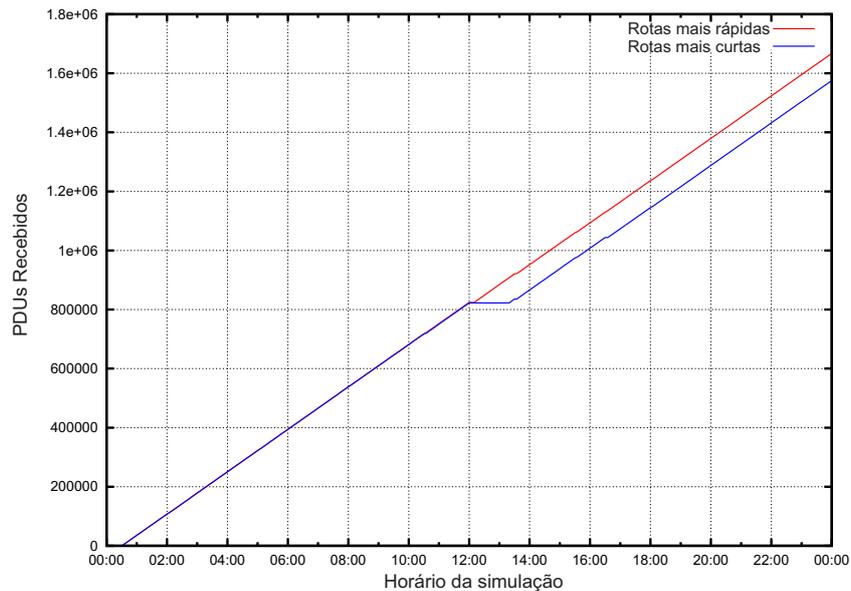
Para comparar os níveis de recebimento e perda de pacotes na RSSF, duas simulações foram executadas: a primeira foi usando uma abordagem que faz com que o *sink* considerado se movimente sempre pelas rotas mais rápidas, e no segundo caso foi usando uma abordagem onde o *sink* prefere as rotas mais curtas, mesmo que isso implique em passar por uma via congestionada. Na primeira simulação, o *sink* encontra as rotas mais rápidas com base em dados do tráfego gerados pelo algoritmo proposto. Para comparar as duas abordagens, a quantidade de unidades de dados de pacotes (PDU) que atingem o *sink* foi contabilizada como mostrado na Figura 5.3. Além disso, a Tabela 5.3 mostra uma redução no tempo médio de viagem mesmo com uma distância percorrida maior, para que os *sinks* se posicionem próximos ao grupo de sensores a serem monitorados.

Tabela 5.3: Comparação entre rotas.

Região do centro da cidade		
Descrição	Rotas mais rápidas	Rotas mais curtas
Tempo da simulação	1 dia	1 dia
Tempo médio das viagens	0:01:26	0:04:23
Tempo total em viagem	0:37:30	1:54:01
Distância percorrida	22,088 Km	20,195 Km

Durante o período simulado, o tempo médio da viagem dos *sinks* móveis pelas vias da região simulada obteve uma economia de tempo de viagem de 69,6 minutos, usando a abordagem proposta, mesmo com um aumento médio de 10 % da distância total percorrida, como mostra a Tabela 5.3.

Às 12 horas, a diferença entre ambas as abordagens fica clara. Neste momento da simulação os congestionamentos são muito intensos na cidade fazendo com que o *sink* gaste mais tempo parado. Ao usar a abordagem proposta, o *sink* pode alcançar os sensores prioritários mais rapidamente, escolhendo rotas menos congestionadas.

Figura 5.3: PDUs recebidos pelo *sink* por hora do dia.

As simulações também compararam o total de pacotes descartados devido ao estouro de *buffer* nos sensores. A Figura 5.4, apresenta o número de pacotes descartados por hora do dia. Assim como na contabilização dos PDUs recebidos, o número de pacotes descartados aumenta após às 12 horas no cenário de rotas mais curtas porque o *sink* fica preso em um engarrafamento, enquanto o sensor está gerando e descartando pacotes devido a um *buffer* cheio.

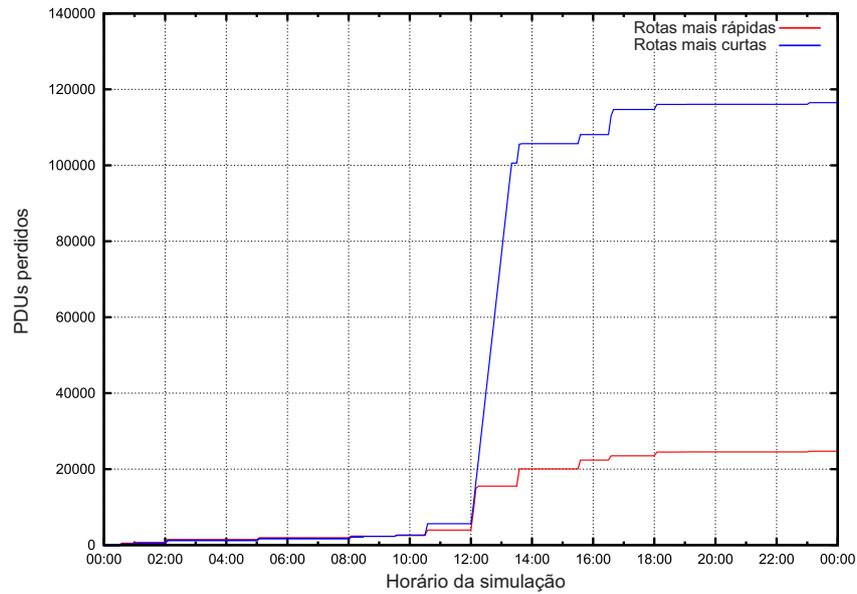


Figura 5.4: Perda de PDUs por hora do dia.

Por fim, o tempo de viagem total durante a simulação foi comparado em ambas as abordagens. Ao escolher as rotas mais curtas o *sink* gasta 114 minutos movendo-se pelas rotas mais curtas enquanto gasta apenas 37,5 minutos se escolher as rotas mais rápidas. Isso dá uma economia de 67,1 % de tempo na abordagem proposta.

Uma segunda verificação foi realizada com três *sinks* móveis usando o mesmo cenário com o propósito de comparar a performance da rede nos dois casos. Nesta simulação foi usado outro cronograma de eventos com três sensores monitorando eventos distintos e transmitindo dados de uma só vez, simulando, desta forma, múltiplos eventos simultâneos acontecendo na cidade, exigindo assim mais *sinks*. Com três *sinks* sempre circundando entre os sensores de transmissão, a diferença da quantidade de pacotes recebidos não foi muito alta em comparação com a abordagem com um *sink*, como mostra a Figura 5.5. Isso é possível porque quanto mais *sinks* estão sendo usados na rede, mais curtos são os caminhos que eles têm que percorrer e menos tempo é gasto nos movimentos dos *sinks*.

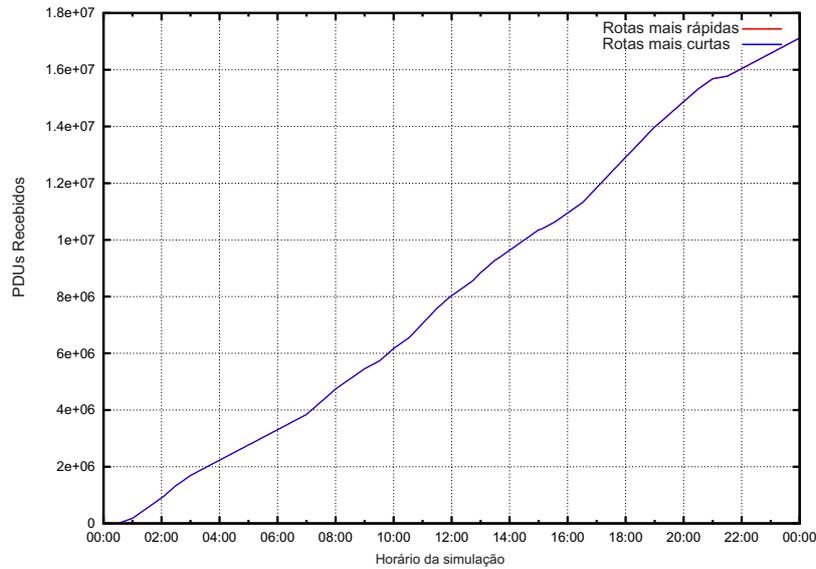


Figura 5.5: PDUs recebidos por três *sinks*.

O mesmo pode ser notado quando se comparam os pacotes descartados, como mostrado na Figura 5.6. Se menos tempo for necessário para mover os *sinks*, menos pacotes serão descartados nas duas simulações. Além disso, nenhuma diferença perceptível estava presente no tempo médio de viagem ou no tempo total de viagem ao usar três *sinks*.

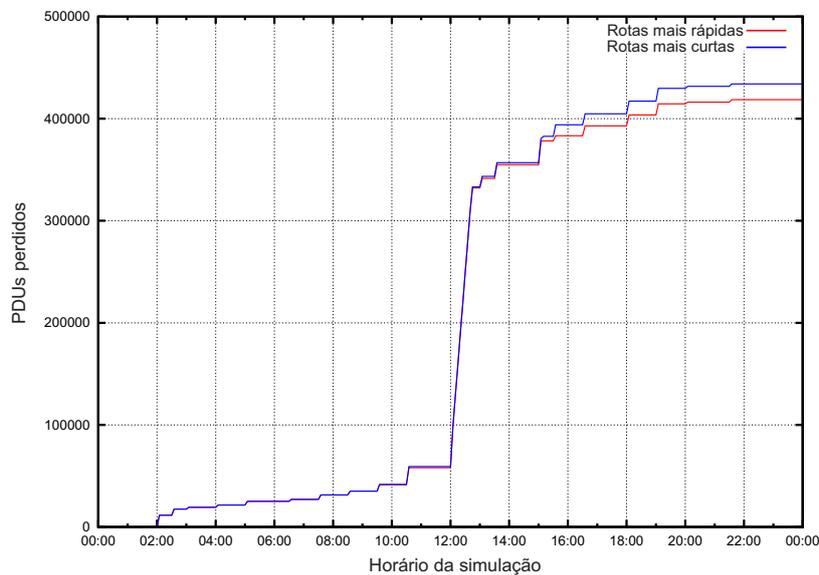


Figura 5.6: Perda de PDUs por hora do dia com três *sinks*.

Portanto, fica claro nesta simulação que estabelecer o número de *sinks* para o monitoramento de sensores espalhados na RSSF de uma determinada região, aumentará a eficiên-

cia da rede, caso aconteça eventos simultâneos na cidade e os *sinks* precisem se posicionar próximo aos grupos de sensores.

5.3 Cenário 2: Análise com dados da Av. Getúlio Vargas

Em outra simulação para verificar a eficiência da abordagem proposta, foi feito um monitoramento da velocidade média dos veículos durante duas semanas em diversos horários na Avenida Getúlio Vargas, em Feira de Santana, Bahia. Os dados obtidos empiricamente foram inseridos no MobSink para comparar com a abordagem proposta, como mostra a Tabela 5.4.

Tabela 5.4: Tabela da velocidade média na Av. Getúlio Vargas.

Velocidade média na Av. Getúlio Vargas (km/h)										
Dia/Hora	06:00	08:00	10:00	12:00	14:00	16:00	18:00	20:00	22:00	00:00
Segunda	25	13	23	10	18	24	12	25	24	42
Terça	24	12	22	8	16	28	13	26	30	40
Quarta	27	10	22	12	17	25	12	24	26	40
Quinta	25	10	22	13	18	24	14	28	26	42
Sexta	28	16	26	18	20	28	12	26	28	40

Foi feita uma simulação por cinco dias, de segunda à sexta, e coletado dados da velocidade média medida em km/h, em horários com alto índice de congestionamento, como mostra a Tabela 5.4. Não foram considerados na simulação os dias de sábado e domingo, por terem níveis menores de congestionamento na região monitorada. Nesse cenário de testes, foram considerados os valores de intensidade do tráfego da Av. Getúlio Vargas. Neste caso, os *sinks* terão que encontrar a rota mais rápida para alcançar os sensores. Os valores de intensidade de tráfego para o multiplicador $m(i)$, usado na Equação 4.7, foram definidos de acordo com a Tabela 5.5, e obtidos a partir da Tabela 5.4. O objetivo neste cenário é validar o algoritmo em situações reais de engarrafamento.

Tabela 5.5: Situação do congestionamento na Getúlio Vargas.

Valores de intensidade de tráfego para o multiplicador $m(i)$										
Dia/Hora	06:00	08:00	10:00	12:00	14:00	16:00	18:00	20:00	22:00	00:00
Segunda	0,41	0,21	0,38	0,16	0,3	0,4	0,2	0,41	0,4	0,7
Terça	0,4	0,2	0,36	0,13	0,26	0,46	0,21	0,43	0,5	0,66
Quarta	0,45	0,16	0,36	0,2	0,28	0,41	0,2	0,4	0,43	0,66
Quinta	0,41	0,16	0,36	0,21	0,3	0,4	0,23	0,46	0,43	0,7
Sexta	0,46	0,26	0,43	0,3	0,33	0,46	0,2	0,43	0,46	0,66

Para a simulação, foram usados três *sinks* móveis para o monitoramento da rede. Sensores foram espalhadas por regiões próximas à avenida monitorada e também utilizaram

um *buffer* de dados de 128 KB. Desta forma, quando os *buffers* estão cheios, pacotes recebidos e gerados pelo próprio sensor são descartados.

O simulador MobSink utiliza a Equação 4.7 para determinar o tempo que os *sinks* móveis irão se mover pelas vias durante um período de cinco dias. De posse do tempo, o algoritmo proposto é usado para obter as rotas mais rápidas toda vez que ocorrer um evento e os *sinks* precisarem se mover pelas vias da cidade, calculando o tempo de viagem entre os *sinks* da rede, como apresentado na Tabela 5.6.

Tabela 5.6: Comparação entre a simulação de rotas.

Região da Av. Getúlio Vargas		
Descrição	Rotas mais rápidas	Rotas mais curtas
Tempo da simulação	5 dias	5 dias
Tempo médio das viagens	00:01:48	00:02:28
Tempo total em viagem	02:08:08	02:54:33
Distância percorrida	83,22 Km	71,69 Km

Durante os cinco dias da simulação, o tempo médio da viagem dos *sinks* móveis pelas vias da região simulada obteve uma redução de 64 % usando a abordagem proposta e redução da distância total percorrida em aproximadamente 11 Km como mostra a Tabela 5.6.

Foi comparado o total de pacotes descartados durante os cinco dias da simulação, como apresenta a Figura 5.7. Ao escolher o cenário de rotas mais rápidas, houve uma redução da perda de pacotes na rede ao longo dos dias monitorados, já que os *sinks* não ficaram presos em engarrafamentos nas vias.

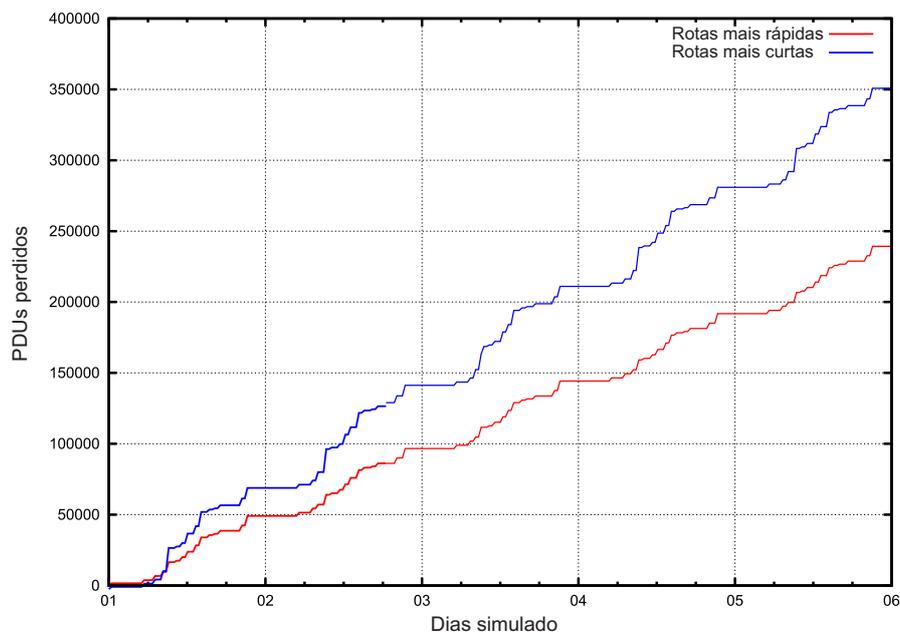


Figura 5.7: PDUs perdidos por dia.

5.4 Cenário 3: Perda de PDUs com muitos sensores

Para esta simulação foi usado um cenário com 1000 sensores espalhados em uma área do ambiente urbano a ser monitorado, com o propósito de monitorar eventos durante o período de um dia. Cada sensor possui um padrão de geração de pacotes de acordo com seu nível de relevância. Cada sensor será configurado com um *buffer* de 128 KB, quando este *buffer* estiver cheio os pacotes recebidos de outros sensores e gerados pelo próprio sensor são descartados. Os testes foram realizados com o objetivo de verificar a abordagem com uma grande quantidade de sensores na RSSF.

Para esta simulação foram utilizados dados de um dia do cenário anterior, com o monitoramento da velocidade média dos veículos na Avenida Getúlio Vargas. Assim, foi inserido na rede um total de 1000 sensores espalhados em *Grid* no ambiente urbano monitorado, como mostra a Figura 5.8. Ao simular a abordagem proposta, os nós sensores foram priorizados de forma aleatória, para simular eventos simultâneos durante vários períodos.

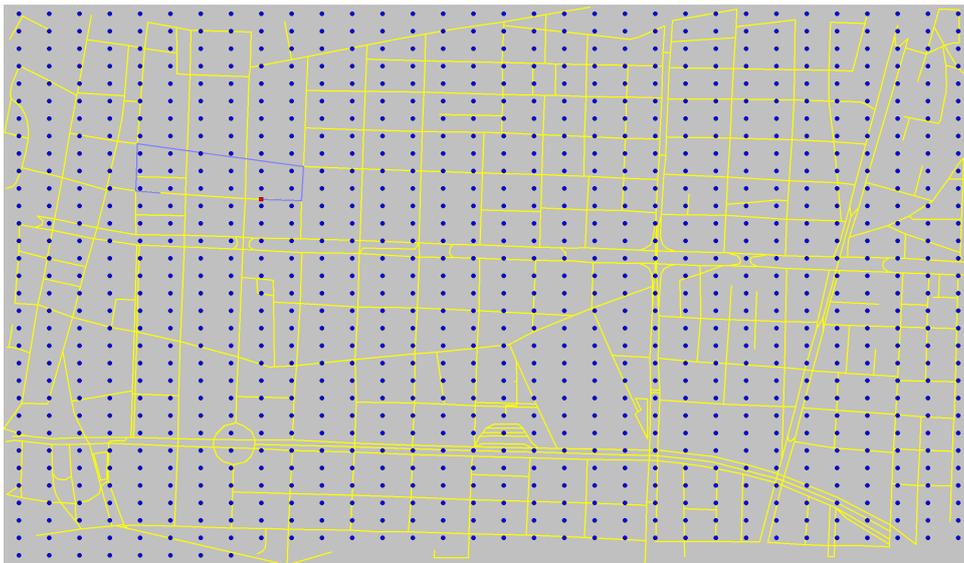


Figura 5.8: Mapa com 1000 sensores.

Para coletar os dados transmitidos, foram realizados dois testes iniciais: o primeiro teste foi usando um *sink* móvel, já no segundo teste foram usados três *sinks* móveis para o monitoramento dos eventos distribuídos na rede, foi simulado durante um dia uma programação com diversos eventos simultâneos e duração de 45 minutos cada evento. A Figura 5.9 demonstra que mesmo com apenas um *sink* móvel usado no monitoramento da rede e usando a abordagem proposta de rota mais rápida, houve uma redução de 23 % na perda de PDUs em comparação com a rota mais curta, mesmo usando grandes quantidades de sensores na RSSF.

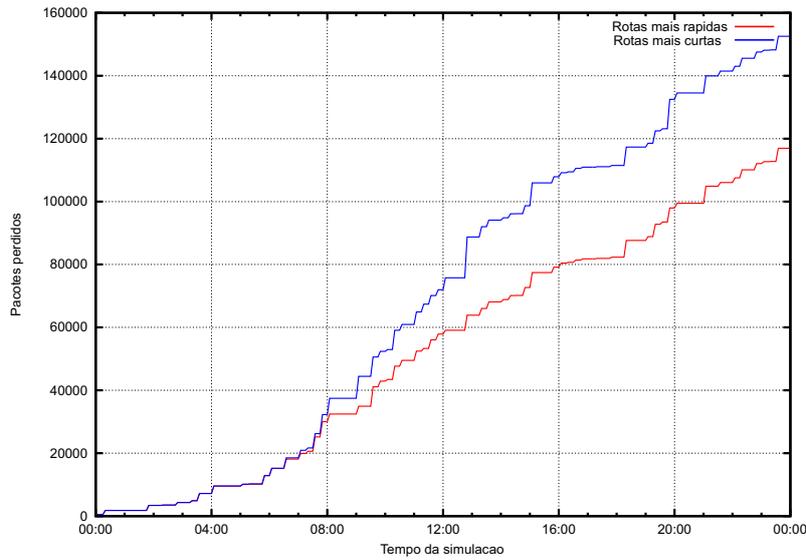


Figura 5.9: Perda de PDUs usando um *sink*.

Um segundo teste foi realizado com três *sinks* móveis usando o mesmo cenário com 1000 sensores, com o propósito de confrontar os níveis de perda de pacote, nos testes com um ou com múltiplos *sinks* móveis. Na simulação com três *sinks* móveis houve uma redução de 20 % na perda de pacotes em comparação com a rota mais curta, mesmo usando uma grande quantidade de sensores na rede, como mostra a Figura 5.10. Portanto, nas duas simulações houve uma redução nos níveis de perda de PDUs na RSSF com 1000 sensores espalhados em *Grid*, e como esperado a abordagem com múltiplos *sinks* móveis obteve um nível menor de perda de pacotes por poder monitorar de forma simultânea uma quantidade maior de eventos durante o dia.

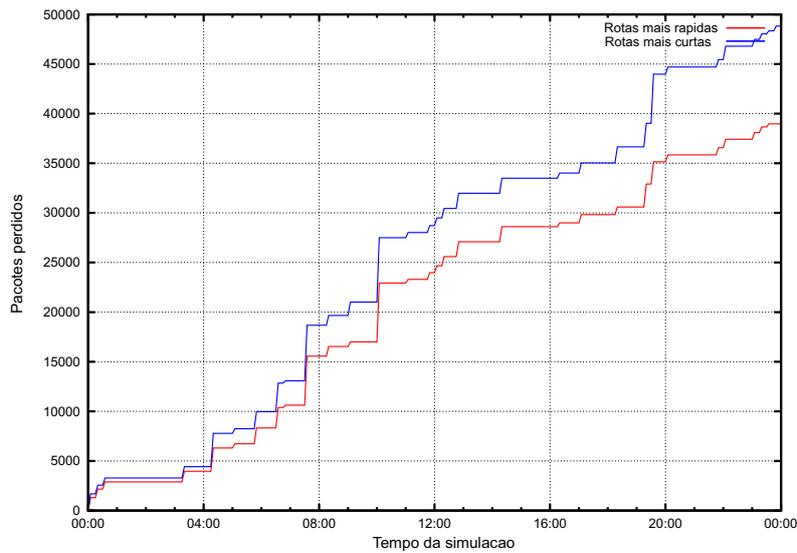


Figura 5.10: Perda de PDUs usando três *sinks*.

Deste modo, o desenvolvimento do algoritmo proposto abriu caminho para o aprimoramento das rotas dos *sinks* móveis em vias com tráfego de veículos em um ambiente urbano, resultando em uma otimização da resposta a eventos e redução nos níveis de perdas de pacotes durante as transmissões de dados nas RSSF, instaladas em um ambiente urbano, abrindo o campo da pesquisa para o aprimoramento de algoritmos que usem dados de trânsito para o roteamento dinâmico dos *sinks* móveis.

5.5 Análise dos Resultados

Neste trabalho, foi proposta uma abordagem de mobilidade de *sinks* móveis em Cidades Inteligentes com o objetivo de melhorar a mobilidade dos *sinks* e o desempenho da rede em ambientes com alto índice de congestionamento urbano. A configuração do tráfego urbano foi usada para simular o congestionamento de veículos e comparar com o modelo proposto. A abordagem proposta de cálculo de rotas mais rápidas para um *sink* móvel em um cenário urbano foi realizada considerando regras de engarrafamentos e limites de velocidade da via, e foi simulado comparando com um cenário de rotas mais curtas. Como esperado, ao usar o modelo com as rotas mais rápidas para os *sinks* móveis em cenários urbanos, as simulações demonstraram que o uso de dados de tráfego para calcular as rotas mais rápidas pode otimizar drasticamente a taxa de transferência de dados em um RSSF.

Dados obtidos durante as simulações mostraram um aumento considerável em pacotes recebidos por um *sink* ao escolher rotas mais rápidas e um aumento nos pacotes descartados ao escolher as rotas mais curtas, ignorando o *status* do tráfego. Portanto, além da redução na perda de pacotes, as otimizações realizadas nas RSSF apresentam inúmeros benefícios relacionados com a performance e economia de recursos da rede, elevando o tempo de vida útil dos sensores na rede.

A simulação com dados reais obtidos na Avenida Getúlio Vargas, Feira de Santana, demonstra uma redução significativa no tempo total de viagem, e distância percorrida, toda vez que ocorrer um evento e o *sink* precisar se mover pela cidade para monitorar um grupo de sensores instalados em uma determinada área da cidade. Reduz-se, desta forma, a perda de pacotes na rede, resultando na economia de recursos de hardware essenciais para RSSF.

Por fim, foi realizada uma simulação com 1000 sensores a fim de verificar a eficácia da abordagem proposta em um ambiente urbano com eventos simultâneos acontecendo em um período de um dia. Os resultados desta simulação mostraram que mesmo ao usar uma grande quantidade de sensores na rede, a abordagem proposta demonstra ser eficiente em relação à redução do número de PDUs perdidos da RSSF.

As simulações mostraram que quando os *sinks* deslocam-se de um grupo de sensores para outro em um ambiente urbano, com o objetivo de monitorar um evento ocorrido na cidade, o congestionamento de veículos pode levar a um atraso no deslocamento e, como consequência, um aumento na perda de pacotes transmitidos na rede, provocando, desta

forma, várias questões referentes a diminuição do tempo de vida útil da rede. Evidentemente, a economia de recursos como o consumo de energia dos nós sensores terá um ganho considerável e o desempenho da rede aumentará com a movimentação e posicionamento otimizado dos *sinks* em redes de sensores sem fio. Portanto, diferentes resultados podem ser obtidos quando levado em consideração as diversas características peculiares do trânsito de uma Cidade Inteligente.

Capítulo 6

Considerações Finais

As Redes de Sensores Sem Fio são um tipo especial de rede que pode ser aplicado para o monitoramento de áreas críticas em um ambiente urbano nas cidades modernas. Este tipo de rede possui diversas restrições de *hardware*, o que requer o desenvolvimento de tecnologias que reduzam o consumo de energia, potencialmente aumentando o tempo de vida útil da rede. Diante disso, a pesquisa aqui proposta comprova que melhorias no deslocamento e posicionamento dos *sinks* móveis em uma rede de sensores sem fio trazem benefícios no desempenho da comunicação e redução de perdas de pacotes na rede.

Para alcançar os objetivos traçados neste trabalho, foi desenvolvido um algoritmo usando como base a teoria dos grafos e modelos matemáticos, tendo como referência o algoritmo Dijkstra para a otimização do posicionamento dos *sinks* móveis em um ambiente urbano. Para isso, foram inseridas no algoritmo informações de tráfego e configurações das vias de uma cidade real. Desta forma, foi possível verificar ganhos no nível de desempenho da comunicação quando são utilizados *sinks* móveis em redes de sensores sem fio.

Um outro aspecto observado durante a pesquisa foi a técnica de priorização global dos nós sensores. O algoritmo proposto calcula o deslocamento dos *sinks* móveis para o grupo de sensores que indica uma mudança de relevância na rede, quando houver um evento urbano, monitorado pelos sensores. Portanto, verificou-se que com este deslocamento e posicionamento baseado em eventos, houve redução da perda de pacotes transmitidos entre um sensor e os *sinks* na rede, e, portanto, espera-se que a solução proposta seja uma importante contribuição para essa área de pesquisa.

Diante disso, pode-se verificar que este trabalho propõe uma nova abordagem para a comunicação entre *sinks* móveis e os grupos de sensores instalados em um ambiente urbano. Foram encontradas na literatura algumas abordagens sobre deslocamento e posicionamento de nós *sinks*, porém nenhuma delas usam uma abordagem específica para cidades inteligentes, com configurações urbanas na otimização da comunicação de uma RSSF. Por este motivo, esta pesquisa despertou uma vasta gama de novos temas para estudo e desenvolvimento de tecnologias futuras com ênfase em cidades inteligentes.

Trabalhos futuros estão focados no desenvolvimento de mecanismos para obter dados do

tráfego em tempo real em uma cidade, com propósito de alimentar diversos algoritmos de roteamento para encontrar a melhor solução para a rota dos *sinks* móveis em um ambiente urbano. Desta forma, dados obtidos em tempo real podem melhorar o desempenho de um modo geral das RSSF nos cenários considerados.

Adicionalmente, espera-se que este trabalho seja estendido de forma a ser aplicado a outros meios de transporte, como o desenvolvimento de um sistema tridimensional para abordagem com drones, que poderá reduzir ainda mais o tempo gasto na mobilidade dos *sinks*, aumentando o desempenho geral da rede. Porém, este tipo de transporte possui diversas limitações, impondo restrições à duração do voo, aos sensores e a carga útil que pode ser levada a bordo, bem como ao alcance e largura de banda das comunicações. Essas e outras questões precisam ser investigadas em pesquisas futuras.

Referências Bibliográficas

- [Achour et al. 2015] Achour, A., Deru, L., e Deprez, J. C. (2015). Mobility management for wireless sensor networks a state-of-the-art. *Procedia Computer Science*, 52:1101–1107.
- [Ahmed et al. 2016] Ahmed, E., Yaqoob, I., Gani, A., Imran, M., e Guizani, M. (2016). Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges. *IEEE Wireless Communications*, 23:10–16.
- [Ahuja et al. 1990] Ahuja, R. K., Mehlhorn, K., Orlin, J., e Tarjan, R. E. (1990). Faster algorithms for the shortest path problem. *Journal of the ACM (JACM)*, 37(2):213–223.
- [Aini e Salehipour 2012] Aini, A. e Salehipour, A. (2012). Speeding up the floyd-warshall algorithm for the cycled shortest path problem. *Applied Mathematics Letters*, 25(1):1–5.
- [Akyildiz et al. 2007] Akyildiz, I. F., Melodia, T., e Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. *Computer networks*, 51(4):921–960.
- [Amer et al. 2016] Amer, H., Salman, N., Hawes, M., Chaqfeh, M., Mihaylova, L., e Mayfield, M. (2016). An improved simulated annealing technique for enhanced mobility in smart cities. *Sensors*, 16(7):1013.
- [Anastasi et al. 2009] Anastasi, G., Conti, M., Di Francesco, M., e Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad hoc networks*, 7(3):537–568.
- [Angelidou 2015] Angelidou, M. (2015). Smart cities: A conjuncture of four forces. *Cities*, 47:95–106.
- [Baraković e Skorin-Kapov 2013] Baraković, S. e Skorin-Kapov, L. (2013). Survey and challenges of qoe management issues in wireless networks. *Journal of Computer Networks and Communications*, 2013.
- [Baronti et al. 2007] Baronti, P., Pillai, P., Chook, V. W., Chessa, S., Gotta, A., e Hu, Y. F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards. *Computer communications*, 30(7):1655–1695.
- [Batty et al. 2012] Batty, M., Axhausen, K., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G., e Portugali, Y. (2012). Smart cities of the future. *The European Physical Journal Special Topics*, 214:481–518.

- [Botta et al. 2016] Botta, A., de Donato, W., Persico, V., e Pescapé, A. (2016). Integration of cloud computing and internet of things: A survey. *Future Generation Computer Systems*, 56(Supplement C):684 – 700.
- [Camp et al. 2002] Camp, T., Boleng, J., e Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502.
- [Chen e Varshney 2004] Chen, D. e Varshney, P. K. (2004). Qos support in wireless sensor networks: A survey. In *International conference on wireless networks*, volume 233, pp. 1–7.
- [Collotta et al. 2018] Collotta, M., Pau, G., e Costa, D. G. (2018). A fuzzy-based approach for energy-efficient wi-fi communications in dense wireless multimedia sensor networks. *Computer Networks*, 134:127–139.
- [Costa 2013] Costa, D. G. (2013). Otimizações da transmissão de imagens em redes de sensores visuais sem fio explorando a relevância de monitoramento dos nós fontes e codificação dwt. *Sensors*.
- [Costa e Guedes 2011] Costa, D. G. e Guedes, L. A. (2011). A survey on multimedia-based cross-layer optimization in visual sensor networks. *Sensors*, 11(5):5439–5468.
- [Costa e Guedes 2013] Costa, D. G. e Guedes, L. A. (2013). Exploiting the sensing relevancies of source nodes for optimizations in visual sensor networks. *Multimedia tools and applications*, 64(3):549–579.
- [Costa et al. 2014] Costa, D. G., Guedes, L. A., Vasques, F., e Portugal, P. (2014). Relevance-based balanced sink mobility in wireless visual sensor networks. In *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*, pp. 124–129. IEEE.
- [Costa et al. 2015] Costa, D. G., Guedes, L. A., Vasques, F., e Portugal, P. (2015). Research trends in wireless visual sensor networks when exploiting prioritization. *Sensors*, 15(1):1760–1784.
- [Divoky e Hung 1990] Divoky, J. J. e Hung, M. S. (1990). Performance of shortest path algorithms in network flow problems. *Management Science*, 36(6):661–673.
- [Dreyfus 1969] Dreyfus, S. E. (1969). An appraisal of some shortest-path algorithms. *Operations research*, 17(3):395–412.
- [García Villalba et al. 2009] García Villalba, L. J., Sandoval Orozco, A. L., Triviño Cabrera, A., e Barenco Abbas, C. J. (2009). Routing protocols in wireless sensor networks. *Sensors*, 9(11):8399–8421.
- [Gupta et al. 2015] Gupta, H. P., Rao, S., Yadav, A. K., e Dutta, T. (2015). Geographic routing in clustered wireless sensor networks among obstacles. *IEEE Sensors Journal*, 15(5):2984–2992.

- [Haklay e Weber 2008] Haklay, M. e Weber, P. (2008). Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18.
- [Hartigan e Wong 1979] Hartigan, J. A. e Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.
- [Hashem et al. 2016] Hashem, I. A. T., Chang, V., Anuar, N. B., Adewole, K., Yaqoob, I., Gani, A., Ahmed, E., e Chiroma, H. (2016). The role of big data in smart city. *International Journal of Information Management*, 36(5):748 – 758.
- [Kapitanova et al. 2012] Kapitanova, K., Son, S. H., e Kang, K.-D. (2012). Using fuzzy logic for robust event detection in wireless sensor networks. *Ad Hoc Networks*, 10(4):709–722.
- [Khan et al. 2014] Khan, P., Konar, G., e Chakraborty, N. (2014). Modification of floyd-warshall’s algorithm for shortest path routing in wireless sensor networks. In *India Conference (INDICON), 2014 Annual IEEE*, pp. 1–6. IEEE.
- [Lassabe et al. 2007] Lassabe, N., Berro, A., e Duthen, Y. (2007). Improvement of a shortest routes algorithm. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pp. 613–617. IEEE.
- [Letaifa 2015] Letaifa, S. B. (2015). How to strategize smart cities: Revealing the smart model. *Journal of Business Research*, 68(7):1414–1419.
- [Liu e Hong 2006] Liu, J. e Hong, X. (2006). A traffic-aware energy efficient routing protocol for wireless sensor networks. In *Computer Architecture for Machine Perception and Sensing, 2006. CAMP 2006. International Workshop on*, pp. 142–147. IEEE.
- [Lu et al. 2014] Lu, H., Li, J., e Guizani, M. (2014). Secure and efficient data transmission for cluster-based wireless sensor networks. *IEEE transactions on parallel and distributed systems*, 25(3):750–761.
- [Magzhan e Jani 2013] Magzhan, K. e Jani, H. M. (2013). A review and evaluations of shortest path algorithms. *International journal of scientific & technology research*, 2(6):99–104.
- [Martincic e Schwiebert 2006] Martincic, F. e Schwiebert, L. (2006). Distributed event detection in sensor networks. In *Systems and Networks Communications, 2006. ICSNC’06. International Conference on*, pp. 43–43. IEEE.
- [Masek et al. 2016] Masek, P., Masek, J., Frantik, P., Fujdiak, R., Ometov, A., Hosek, J., Andreev, S., Mlynek, P., e Misurec, J. (2016). A harmonized perspective on transportation management in smart cities: the novel iot-driven environment for road traffic modeling. *Sensors*, 16(11):1872.
- [Mattoni et al. 2015] Mattoni, B., Gugliermetti, F., e Bisegna, F. (2015). A multilevel method to assess and design the renovation and integration of smart cities. *Sustainable Cities and Society*, 15:105–119.

- [Mottola e Picco 2011] Mottola, L. e Picco, G. P. (2011). Programming wireless sensor networks: Fundamental concepts and state of the art. *ACM Computing Surveys (CSUR)*, 43(3):19.
- [Musznicki et al. 2012] Musznicki, B., Tomczak, M., e Zwierzykowski, P. (2012). Dijkstra-based localized multicast routing in wireless sensor networks. In *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on*, pp. 1–6. IEEE.
- [Narayanan et al. 2015] Narayanan, A., Mitrovic, N., Asif, M. T., Dauwels, J., e Jaillet, P. (2015). Travel time estimation using speed predictions. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 2256–2261.
- [Oh et al. 2010] Oh, S., Lee, E., Park, S., Jung, J., e Kim, S.-H. (2010). Communication scheme to support sink mobility in multi-hop clustered wireless sensor networks. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pp. 866–872. IEEE.
- [Oyman e Ersoy 2004] Oyman, E. I. e Ersoy, C. (2004). Multiple sink network design problem in large scale wireless sensor networks. In *Communications, 2004 IEEE International Conference on*, volume 6, pp. 3663–3667. IEEE.
- [Peixoto 2015] Peixoto, J. P. J. (2015). Posicionamento de múltiplos sinks em redes de sensores sem fio com priorização. Master's thesis, Mestrado em Computação Aplicada. DEPARTAMENTO DE TECNOLOGIA.
- [Peixoto e Costa 2015a] Peixoto, J. P. J. e Costa, D. G. (2015a). Lara mobsink multiple sinks simulator. Internet.
- [Peixoto e Costa 2015b] Peixoto, J. P. J. e Costa, D. G. (2015b). QoS-aware multiple sinks mobility in wireless sensor networks. In *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–4. IEEE.
- [Peixoto e Costa 2017a] Peixoto, J. P. J. e Costa, D. G. (2017a). Wireless visual sensor networks for smart city applications: A relevance-based approach for multiple sinks mobility. *Future Generation Computer Systems*.
- [Peixoto e Costa 2017b] Peixoto, J. P. J. e Costa, D. G. (2017b). Wireless visual sensor networks for smart city applications: A relevance-based approach for multiple sinks mobility. *Future Generation Computer Systems*, 76(Supplement C):51–62.
- [Petrioli et al. 2014] Petrioli, C., Nati, M., Casari, P., Zorzi, M., e Basagni, S. (2014). Load-balancing geographic routing around connectivity holes in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(3):529–539.
- [Pokhrel et al. 2014] Pokhrel, J., Paudel, I., Wehbi, B., Cavalli, A., e Jouaber, B. (2014). Performance evaluation of video transmission over 802.11 n wireless network: A mac layer perspective. In *Smart Communications in Network Technologies (SaCoNeT), 2014 International Conference on*, pp. 1–6. IEEE.

- [Qiu et al. 2016] Qiu, J., Wu, Q., Ding, G., Xu, Y., e Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1):67.
- [Rawat et al. 2014] Rawat, P., Singh, K. D., Chaouchi, H., e Bonnin, J. M. (2014). Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of Supercomputing*, 68(1):1–48.
- [Sarkar e Murugan 2016] Sarkar, A. e Murugan, T. S. (2016). Routing protocols for wireless sensor networks: What the literature says? *Alexandria Engineering Journal*, 55(4):3173–3183.
- [Silva et al. 2014] Silva, R., Silva, J. S., e Boavida, F. (2014). Mobility in wireless sensor networks—survey and proposal. *Computer Communications*, 52:1–20.
- [Souza et al. 2016] Souza, A. M. d. et al. (2016). Controle de congestionamentos de veículos utilizando sistemas de transporte inteligentes. Master’s thesis, UNICAMP.
- [Su et al. 2011] Su, K., Li, J., e Fu, H. (2011). Smart city and the applications. In *International Conference on Electronics, Communications and Control (ICECC)*, pp. 1028–1031. IEEE.
- [van Wee et al. 2013] van Wee, B., Annema, J., e Banister, D. (2013). *The Transport System and Transport Policy: An Introduction*. Edward Elgar.
- [Yick et al. 2008] Yick, J., Mukherjee, B., e Ghosal, D. (2008). Wireless sensor network survey. *Computer networks*, 52(12):2292–2330.
- [Yin et al. 2015] Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D., e David, B. (2015). A literature survey on smart cities. *Science China Information Sciences*, 58(10):1–18.
- [Zhang et al. 2016] Zhang, J.-d., Feng, Y.-j., Shi, F.-f., Wang, G., Ma, B., Li, R.-s., e Jia, X.-y. (2016). Vehicle routing in urban areas based on the oil consumption weight-dijkstra algorithm. *IET Intelligent Transport Systems*, 10(7):495–502.