

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS ARARANGUÁ**

Victor Aguiar Marcon

**MECANISMOS DE CONTROLE DAS TRANSMISSÕES  
EM REDES DE SENSORES SEM FIO: UMA AVALIAÇÃO  
EXPERIMENTAL**

Araranguá

2018



Victor Aguiar Marcon

**MECANISMOS DE CONTROLE DAS TRANSMISSÕES  
EM REDES DE SENSORES SEM FIO: UMA AVALIAÇÃO  
EXPERIMENTAL**

Trabalho de Conclusão de Curso submetido ao Programa de graduação em Engenharia de Computação para a obtenção do Grau de Bacharel em Engenharia de Computação.

Orientador: Ricardo Moraes, Dr.

Coorientadora: Suelen Laurindo, Msc.

Araranguá

2018

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

Marcon, Victor

Mecanismos de controle das transmissões em redes  
de sensores sem fio: Uma avaliação experimental /  
Victor Marcon ; orientador, Ricardo Moraes,  
coorientadora, Suelen Laurindo, 2018.

70 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Campus  
Araranguá, Graduação em Engenharia de Computação,  
Araranguá, 2018.

Inclui referências.

1. Engenharia de Computação. 2. Redes de sensores  
sem fio. 3. Experimento. 4. Redes Autônomas. 5.  
Teoria de Jogos. I. Moraes, Ricardo. II. Laurindo,  
Suelen. III. Universidade Federal de Santa  
Catarina. Graduação em Engenharia de Computação. IV.  
Título.

Victor Aguiar Marcon

**MECANISMOS DE CONTROLE DAS TRANSMISSÕES  
EM REDES DE SENSOES SEM FIO: UMA AVALIAÇÃO  
EXPERIMENTAL**

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de "Bacharel em Engenharia de Computação", e aprovado em sua forma final pelo Programa de graduação em Engenharia de Computação.

Araranguá, 30 de Novembro 2018.

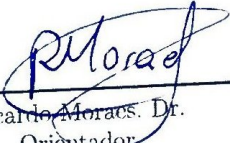


---

Eliane P. Zitzler, Dra.  
Coordenadora  
Universidade Federal de Santa Catarina


Banca Examinadora:

---



Ricardo Moraes, Dr.  
Orientador

---

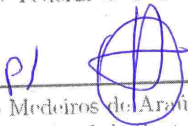


Suelen Laurindo, Msc.  
Coorientadora



---

Analucia Schiaffino Morales. Dra.  
Universidade Federal de Santa Catarina



---

Gustavo Medeiros de Araújo. Dr.  
Universidade Federal de Santa Catarina

Prof. Dr. Eliane Pozzebon  
Professor Adjunto  
SIAPE: 1680681  
UFSC Campus Araranguá







## AGRADECIMENTOS

Gostaria de agradecer a todos que estiveram sempre ao meu lado nestes anos de faculdade, as pessoas especiais pra mim.

Aos meus amigos, alguns presentes e outros nem tanto, eles Gustavo, Raul, Norberto, Diego, Jhuan, Renan, Conrado, que através de fortes laços da amizade me deram forças e minha namorada Bianca, que me apoia e me torna um homem melhor dia após dia.

Especialmente meu orientador e minha coorientadora, que estiveram sempre presentes no desenvolvimento deste documento e foram essenciais para a conclusão do mesmo.

E também ao corpo docente da UFSC - Araranguá, que mesmo diante dos mais variados problemas, lutou para que o campus consolidasse.



We are all connected on a deeper plane  
of existence.

(Enigma, 2013)



## RESUMO

Em algumas redes de sensores sem fio (RSSF), nodos sensores estão dispostos de maneira redundante, isto faz com que um grande número de nodos transmitam a mesma informação desnecessariamente. Diante deste problema várias soluções foram propostas visando escalonar o número de nodos que necessitam estar ativos ao mesmo tempo para garantir a qualidade de serviço. Este trabalho explora os algoritmos de escalonamento de rede que se baseiam na Teoria de Jogos para executar um controle não cooperativo entre os nodos, apresentando uma avaliação experimental que visará analisar a factibilidade de implementação de um conjunto de abordagens disponíveis na literatura.

**Palavras-chave:** Redes de sensores sem fio. Teoria de Jogos. Experimentação.



## ABSTRACT

In some wireless sensor networks (WSNs), sensor nodes are arranged redundantly, this causes a large number of nodes to transmit the same information unnecessarily. Based on this problem, several solutions were proposed aiming to schedule the number of nodes that need to be active at the same time to ensure the quality of service. This work explores network scheduling algorithms based on Game Theory to perform a non-cooperative control between the nodes, presenting an experimental assessment that will analyze the feasibility of implementing a set of approaches available in the literature.

**Keywords:** Wireless Sensor Networks. Game Theory. Experimentation.





## LISTA DE FIGURAS

Figura 1	Exemplo de uso de RSSF em conjunto com outros meios.	30
Figura 2	Topologia de rede do padrão IEEE 802.15.4.	31
Figura 3	Nodos FFD e RFD em uma rede.	32
Figura 4	Um módulo XBee série 1 (DIGI International®).	33
Figura 5	Integração de RSSF com Teoria de Jogos.	36
Figura 6	Função da probabilidade.	40
Figura 7	Autômato de tamanho S.	41
Figura 8	Função da probabilidade Gureen Game.	43
Figura 9	Funcionamento da maquina de estados do Gureen Game.	44
Figura 10	Máquina de estados do Skip Game.	48
Figura 11	QoS vs número de estados.	51
Figura 12	Curva de carga de uma bateria de Lipo comum.	52
Figura 13	Nodo conectado à bateria.	56
Figura 14	Disposição radial dos nodos na experimentação.	57
Figura 15	Modelo de comunicação do experimento.	58
Figura 16	Resultados dos experimentos.	58
Figura 17	Resultados Simulação.	59
Figura 18	Resultados Simulação.	60
Figura 19	Consumo energético da rede em cada configuração.	62



## LISTA DE TABELAS

Tabela 1	Configurações do padrão IEEE 802.15.4.....	32
Tabela 2	Modelos disponíveis da série 1 (DIGI International®).	33
Tabela 3	Estados designados do skip game. ....	50
Tabela 4	Algoritmos e suas configurações.....	50
Tabela 5	Valores Exemplo de imparcialidade.....	53
Tabela 6	Parâmetros da simulação no OMNeT++.....	55
Tabela 7	Parâmetros da experimentação.....	57
Tabela 8	Desvio padrão da QoS entre os testes experimentais....	60
Tabela 9	Desvio padrão da QoS entre as simulações. ....	61
Tabela 10	Imparcialidade média dos testes.....	61
Tabela 11	Consumo de energia para objetivo em 30%.....	62
Tabela 12	Número de nodos mortos experimento. ....	63
Tabela 13	Número de nodos mortos nas simulações.....	63



## LISTA DE ABREVIATURAS E SIGLAS

RSSF	Rede de Sensores Sem Fio
MAC	Medium Access Control
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
DS-CDMA	<i>Direct Sequence Code Division Multiple Access</i>
P2P	<i>peer-to-peer</i>
SSM	<i>Soft System Methodology</i>
AT	Modo Transparente
DOS	<i>Denial of Service</i>
API	<i>Application Program Interface</i>
LiPo	<i>Lithium Ion Polymer</i>



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	23
1.1	MOTIVAÇÃO .....	24
1.2	OBJETIVOS .....	25
1.2.1	Objetivo Geral .....	25
1.2.2	Objetivos Específicos .....	25
1.3	METODOLOGIA .....	26
1.3.1	Classificação da pesquisa em relação a natureza ...	26
1.3.2	Classificação da pesquisa em relação a objetivos ...	27
1.3.3	Classificação da pesquisa em relação a abordagem .	27
1.3.4	Classificação da pesquisa em relação aos procedi- mentos técnicos .....	27
1.4	ORGANIZAÇÃO DO DOCUMENTO .....	28
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	29
2.1	REDES DE SENSORES SEM FIO .....	29
2.1.1	Características das RSSF .....	29
2.2	PADRÃO IEEE 802.15.4 .....	31
2.3	TECNOLOGIAS QUE IMPLEMENTAM O PADRÃO IEEE 802.15.4 .....	33
2.3.1	Modos de comando XBee .....	33
2.4	TEORIA DE JOGOS .....	34
2.4.1	Teoria de jogos em RSSF .....	35
2.4.2	Jogos para aprimorar Comunicação sem fio .....	37
2.4.3	Jogos para aprimorar Qualidade de Serviço .....	37
<b>3</b>	<b>ALGORITMOS DE CONTROLE</b> .....	39
3.1	ALGORITMOS AUTÔNOMOS APLICADOS À RSSF ..	39
3.2	ALGORITMO GUR GAME .....	39
3.3	ALGORITMO GUREEN GAME .....	42
3.4	ALGORITMO $(M, K)$ GUR GAME .....	45
3.5	ALGORITMO SKIP GAME .....	48
3.6	CONFIGURAÇÃO DOS ALGORITMOS .....	50
3.7	MÉTRICAS .....	50
3.7.1	QoS - Quality of Service .....	51
3.7.2	Imparcialidade .....	51
3.7.3	Consumo de energia .....	52
3.7.4	Nodos Mortos .....	52
<b>4</b>	<b>EXPERIMENTOS</b> .....	55
4.1	DESCRIÇÃO DO CENÁRIO .....	55

4.1.1	Modelo de comunicação .....	57
4.2	RESULTADOS .....	58
4.2.1	QoS .....	58
4.2.2	Imparcialidade .....	61
4.2.3	Consumo energético .....	61
4.3	DIFICULDADES ENCONTRADAS .....	63
5	<b>CONCLUSÃO</b> .....	65
5.1	TRABALHOS FUTUROS .....	66
	<b>REFERÊNCIAS</b> .....	67



# 1 INTRODUÇÃO

Redes de sensores sem fio (RSSF) são formadas por um conjunto de nodos de baixo custo, limitado poder computacional (processador e memória), com capacidade sensorial (temperatura, umidade, pressão atmosférica etc) e, muitas vezes, operando de maneira autônoma (DEIF; GADALLAH, 2014). As RSSF estão sendo cada vez mais utilizadas em processos de monitoramento e detecção de eventos, operando em múltiplos ambientes. Os níveis de qualidade de serviço (QoS) variam de acordo com as aplicações e, muitas vezes há a necessidade que essas redes funcionem sem manutenção por um longo período de tempo (WERNER-ALLEN et al., 2006).

Um dos principais desafios a serem superados em RSSF se trata do consumo energético, pois, em ambientes hostis pode não ser prático, ou até mesmo impossível, realizar a troca da bateria dos nodos da rede, fazendo com que o desempenho caia drasticamente no momento em que algumas das baterias esgotarem. Isto implica em manter o tempo de vida de rede o maior possível, fazendo com que a rede fique operante por mais tempo com a QoS desejada. A qualidade de serviço dependerá da aplicação. Uma técnica comum que Semprebom et al. (2015) citam para melhorar o desempenho das RSSF é a implementação de nodos redundantes, aumentando a resistência da rede a falhas. Além disso, para aumentar o tempo de vida das RSSF, técnicas de escalonamento de sono são, geralmente, implementadas. Neste caso, apenas um certo número de nodos precisam estar acordados, executando as tarefas de sensoriamento e comunicação, enquanto os demais nodos entrarão em modo *sleep*, ou seja, estarão funcionando em modo de economia de energia. Desta forma, além de economizar energia também se diminui o tráfego da rede, portanto, consumindo energia e banda de maneira mais eficiente (AYERS; LIANG, 2011).

No processo de escalonamento dos nodos da rede é necessário levar em conta a área de cobertura, fazendo com que exista um conjunto de nodos enviando dados das regiões monitoradas e que ocorra a maximização das mensagens recebidas com sucesso no nodo coordenador. Portanto, a eficiência energética e o tempo de vida da rede são métricas essenciais para medir o desempenho de qualquer abordagem de monitoramento utilizando RSSF, a definição da mesma pode variar de aplicação para aplicação (DIETRICH; DRESSLER, 2009). Assumindo uma implementação com nodos redundantes, uma definição apropriada para o tempo de vida da rede é a que considera que uma rede deve con-

tinuar operacional até que não existam nodos suficientes para efetuar o serviço de monitoramento. Esta definição independe do número de nodos da rede, e o número de nodos mínimos é definido pela aplicação (SEMPREBOM et al., 2015). Ou seja, aplicações relacionadas a RSSF, geralmente utilizam nodos redundantes e não há necessidade de manter todos os nodos transmitindo dados em todos os ciclos de operação. Porém ao reduzir o número de nodos ativos é necessário manter um nível de qualidade de serviço (CURIAC et al., 2009). Alguns algoritmos de controle com base na teoria de jogos tem sido propostos para realizar a escolha dos nós que devem transmitir em cada ciclo de operação da RSSF. Neste contexto, Iyer e Kleinrock (2003) propuseram o chamado Gur Game, que foi aprimorado pelo trabalho de Ayers e Liang (2011), definindo-o como Gureen Game e, mais recentemente, Semprebom et al. (2015) publicaram o algoritmo conhecido por Skip Game, que busca rápida convergência e melhor consumo energético.

A questão fundamental ao se tratar QoS em uma RSSF é como controlar a quantidade ótima de nodos que transmitem mensagens para o coordenador em cada intervalo de tempo. Pois, as condições da rede são dinâmicas e os nodos, geralmente, não tem informações globais da rede, como por exemplo o número de nodos disponíveis (ativos e *stand by*), que varia devido a falhas, mobilidade etc. Por fim, o sistema de controle de nodos transmissores deve ser distribuído e autônomo, onde os nodos e o coordenador trabalham de maneira cooperativa (IYER; KLEINROCK, 2003).

Na literatura encontra-se diversas abordagens autônomas que utilizam RSSF, para selecionar os nodos transmissores em cada instante de tempo, porém, a grande maioria é avaliada somente através de simulações. Este trabalho apresentará uma avaliação experimental analisando a factibilidade de implementação das principais abordagens disponíveis na literatura.

## 1.1 MOTIVAÇÃO

RSSF impulsionada pela evolução tecnológica se torna cada vez mais comum para a aquisição de dados para funcionamento de agentes distribuídos ou para fins que ainda estarão por vir, e por mais que se esteja consolidada, a aplicação de uma RSSF é repleta de desafios. (SHARMA; BANSAL; BANSAL, 2013).

Este trabalho investiga a viabilidade de implementação dos algoritmos, que propõem melhorias de QoS, eficiência energética e im-

parcialidade.

Continuamente busca-se produzir RSSF mais eficientes, soluções com base nas mais variadas técnicas computacionais foram propostas para atacar um ou mais problemas pertinentes a RSSF, como lógica Fuzzy (ZINONOS; CHRYSOSTOMOU; VASSILIOU, 2014), técnicas de algoritmos genéticos (JIA et al., 2009) e Teoria de Jogos (IYER; KLEINROCK, 2003), entre tantas outras.

Apesar de uma gama enorme de soluções propostas, a maioria se concentra em simulações de alto nível em plataformas de simulação geral como OMNeT++ ou específicas para RSSF como Avrrora ou PowerTOSSIM, e uma quantidade menor de propostas testadas de maneira experimental (HORNEBER; HERGENRODER, 2014).

Mesmo com bom desempenho dos ambientes de simulação, ainda existe uma disparidade significativa, diante disto este trabalho é fundamentado sobre a seguinte pergunta de pesquisa:

“É viável a implementação de algoritmos autônomos baseados na teoria de jogos, que, selecionem quais nodos devem estar ativos e quais devem estar inativos, realizando a tarefa requerida com sucesso, utilizando redes de sensores sem fio com nodos redundantes?”.

## 1.2 OBJETIVOS

Os objetivos gerais e específicos deste trabalho são descritos abaixo.

### 1.2.1 Objetivo Geral

Este trabalho possui como objetivo geral avaliar experimentalmente um conjunto de algoritmos de autônomos para o controle das transmissões em RSSF, esses algoritmos estão disponíveis na literatura e foram estudados em ambientes de simulação. Nesse trabalho o seu desempenho e sua factibilidade de implementação em nodos comerciais são analisados.

### 1.2.2 Objetivos Específicos

Para alcançar este objetivo principal, os seguintes objetivos específicos são requeridos:

1. Realizar um levantamento bibliográfico sobre as áreas de pesquisa do trabalho, como RSSF, redes autônomas e Teoria de jogos;
2. Selecionar um conjunto de implementações de RSSF autônomas em que se utiliza a Teoria de Jogos para aprimorar certas características de rede;
3. Propor uma plataforma de experimentação compatível com os algoritmos selecionados;
4. Validar os códigos desenvolvidos na plataforma;
5. Definir métricas de desempenho que serão utilizadas para avaliar os algoritmos;
6. Desenvolver um cenário de avaliação;
7. Modelar cenários experimentais baseado no cenário desenvolvido;
8. Executar os experimentos e analisar resultados.

### 1.3 METODOLOGIA

Esta seção descreve o trabalho no que diz respeito a metodologia adotada, onde foram descritas as características metodológicas do trabalho e os procedimentos realizados.

#### 1.3.1 Classificação da pesquisa em relação a natureza

A pesquisa quanto à sua natureza classifica-se como básica e como aplicada, onde a primeira busca por novos conhecimentos que contribuam para o avanço da ciência em estudo, no entanto a sua aplicabilidade prática ainda inexistente e a de natureza aplicada busca por conhecimento que resultarão em uma aplicação prática, com o intuito de solucionar problemas específicos. Com base nisto, o presente trabalho se caracteriza como uma pesquisa aplicada, pois tem o objetivo de realizar um teste experimental que verifica a factibilidade de pesquisas já realizadas, identificando resultados e analisando-os.

### 1.3.2 Classificação da pesquisa em relação a objetivos

Segundo Severino (2007), a pesquisa pode ser classificada em três tipos, de acordo com os seus objetivos, sendo estes: pesquisa exploratória, pesquisa descritiva e pesquisa explicativa. Gil (2010) aponta que uma pesquisa é classificada como exploratória quando busca-se um maior esclarecimento de um tema ou problema ainda pouco explorado, tornando-o mais compreensivo. Já a pesquisa descritiva busca verificar um fenômeno e suas características, através de um estudo padronizado de coleta de dados, onde se realiza observações, análises e registros. Uma pesquisa é considerada explicativa quando investiga por meio de métodos empíricos, ou seja, tem o objetivo de testar explicar o porquê das coisas (MARCONII; LAKATOS, 2003).

Com os tipos de pesquisa abordados e suas características, este trabalho se assemelha com uma pesquisa descritiva, uma vez que busca realizar uma análise de algoritmos da literatura e por meio de uma plataforma compatível realizar uma coleta de dados experimental.

### 1.3.3 Classificação da pesquisa em relação a abordagem

No que se refere à abordagem da pesquisa, a classificação se divide em duas, quantitativa e qualitativa, uma pesquisa quantitativa é quando resultados e informações podem ser revertidas em valores numéricos, podendo ser quantificados, já a pesquisa qualitativa, quando as informações não podem ser expressas em números, são subjetivas, podendo também ser combinada, quando há uma integração entre ambas abordagens mencionadas (PROVDANOV; FREITAS, 2013) (MELLO et al., 2012). Dessa forma, analisando o pressuposto dos autores mencionados, é possível afirmar que o trabalho atual possui características quantitativas, já que será necessário analisar um conjunto de resultados numéricos e colocá-lo em tabelas e gráficos, realizando comparações.

### 1.3.4 Classificação da pesquisa em relação aos procedimentos técnicos

De acordo com Mello et al. (2012), no que tange os procedimentos técnicos, a pesquisa pode ser classificada em várias categorias com por exemplo: levantamento ou *survey*, pesquisa experimental, modelagem e simulação que se enquadram na categoria de quantitativa; estudo

de caso, pesquisa-ação e *Soft System Methodology* - SSM na perspectiva qualitativa. Desta maneira contata-se que o presente trabalho se caracteriza como uma pesquisa experimental.

## 1.4 ORGANIZAÇÃO DO DOCUMENTO

Este Trabalho de Conclusão de Curso está organizado em mais três capítulos sendo estes:

O **Capítulo 2** apresenta uma fundamentação teórica para as principais áreas de pesquisa relacionadas a este trabalho, mencionando o que são RSSF, suas principais características, abrange sobre o padrão IEEE 802.15.4, discorre sobre a teoria de jogos e sua vertente algorítmica com sua implementação em RSSF.

O **Capítulo 3** apresenta os principais algoritmos de teoria de jogos selecionados na literatura, de que maneira eles foram configurados nos experimentos em relação a trabalhos prévios, apresentando os resultados dos experimentos, comparando com os dados contidos na literatura, além de adicionar métricas que não foram comparadas previamente.

O **Capítulo 4** apresenta os resultados dos experimentos.

O **Capítulo 5** apresenta a conclusão do trabalho, e discute propostas futuras.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar conceitos básicos a respeito das RSSF, do padrão IEEE 802.15.4, além da teoria dos jogos, focando principalmente na sua aplicabilidade em RSSF.

Sendo assim, são descritos de forma sucinta os aspectos mais relevantes para a compreensão deste trabalho.

### 2.1 REDES DE SENSORES SEM FIO

RSSF são constituídas de dispositivos autônomos distribuídos em uma área geográfica que se comunicam empregando tecnologias sem fio, coletando informações e detectando eventos do ambiente. Cada um destes dispositivos é capaz de comunicar, processar e executar sensoriamento. Devido a estas características as RSSF podem ser empregadas em um vasto número de aplicações (YETGIN et al., 2017).

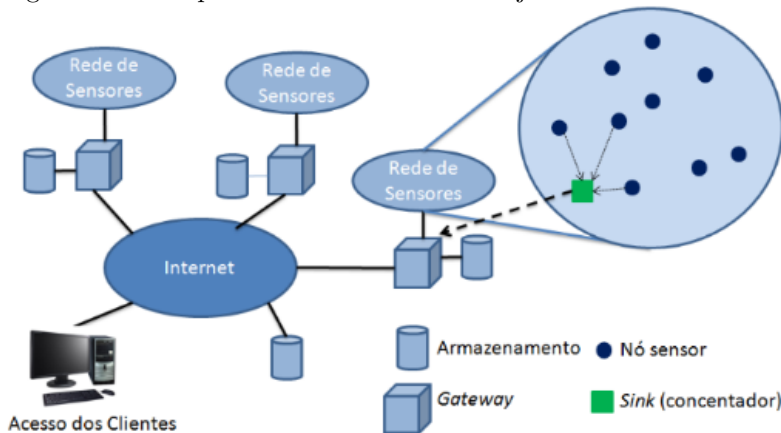
RSSF tem se tornado uma solução solução muito utilizada desde sistemas distribuídos como *smart grid*, até aplicações exigentes como vídeo vigilância (MARTINEZ; LI; ZHOU, 2014). As informações que as RSSF adquirem podem ser transmitidas para um nodo central no contexto de uma rede com topologia estrela, ou para os nodos vizinhos no caso de uma rede com topologia em malha. Além disso, uma RSSF pode estar conectada a um computador central, que organiza e analisa as informações e esse computador, pode estar conectado a internet ou satélite, como ilustra a Figura 1 (BURATTI et al., 2009).

Também é possível observar na Figura 1 a presença do nodo coordenador. Nele a presença de sensoriamento não é obrigatória, mas tem como obrigação transmitir informações coletadas para o *gateway*, este é responsável por transmitir os dados coletados para um outro meio.

#### 2.1.1 Características das RSSF

Os problemas de uma RSSF são parecidos com os de uma rede *ad hoc*, sendo eles, necessidade de mecanismos de controle de acesso ao meio, criação de gerenciamento de rotas, mobilidade e segurança. No entanto há características que são exclusivas de RSSF tais como topologia múltipla de rede, ou estrela, ou P2P - *peer-to-peer*, ou uma

Figura 1 – Exemplo de uso de RSSF em conjunto com outros meios.



Fonte: (BURATTI et al., 2009).

combinação destas. Podem ser implementadas em diversas tecnologias como ZigBee, RF4CE, MiWi, entre outras. As RSSF diferem em suas características de tráfego de dados bem definidas e sofrem de limitações severas de recursos pois possuem baixo poder computacional e limitada carga de bateria (AKYILDIZ et al., 2002).

De acordo com Lazarescu (2013), algumas características são requeridas para o bom funcionamento de uma RSSF, alguma delas são citadas a seguir:

**Tolerância a falha:** Como qualquer rede é passível de falhas, os sistemas de uma RSSF devem estar aptos a arcar com estas falhas, que podem ser tanto de software, como de hardware, já que os mesmos podem estar inseridos em um ambiente que não seja favorável realizar a manutenção.

**Custo de Produção:** Como aplicações podem requerer uma quantidade significativa de nodos, o custo de cada unidade tem uma grande importância, tendo em vista de que caso a rede de sensores possua um valor mais elevado do que uma maneira tradicional de sensoriamento, pode não haver justificativa em sua implementação.

**Alcance:** Na transmissão sem fio entre dois nodos, a altura que o nodo fica em relação ao chão é importante, pois o alcance máximo de uma antena não depende puramente de possuir a visão direta de outra, estando relacionado com um volume que circunda o caminho direto do sinal. Devido a antenas de tamanho reduzido utilizadas em RSSF, esta



altura do chão não precisa ser tão grande como ocorre com as antenas utilizadas em telecomunicações. E este volume tem seu raio com base na equação 2.1. Onde é possível perceber a escala em quilômetros para a distância entre duas antenas, portanto no ambiente de RSSF deste trabalho este volume terá uma escala muito reduzida, pois terá uma distância de antenas em uma escala de poucos metros.

$$r_{metros} = 17.32 \sqrt{\frac{d_{km}}{4f_{GHz}}} \quad (2.1)$$

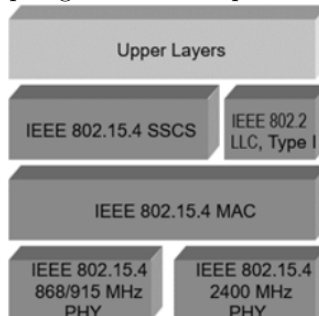
**Escalabilidade:** Suportar uma grande quantidade de sensores, permitindo em tempo de execução a adição de mais nodos, ou como de costume por falhas na rede, escalar para uma quantidade menor de nodos ativos, sem que interfira no funcionamento da rede.

**Consumo de energia:** O maior requisito de algumas RSSF é manter ao máximo o tempo de vida da rede, evitando o desperdício com recebimento e envio de mensagens desnecessárias, e mantendo um ciclo correto de *sleep*, aproveitando o máximo de suas baterias.

## 2.2 PADRÃO IEEE 802.15.4

O padrão IEEE 802.15.4 define o funcionamento de redes sem fio de baixa potência (IEEE, 2016). Ele especifica as camadas física e de enlace (controle de acesso ao meio), esta última subdividida em SSCS (*Specific convergence sub-layer*) e LLC (*local link control*), como pode ser visto na Figura 2.

Figura 2 – Topologia de rede do padrão IEEE 802.15.4.



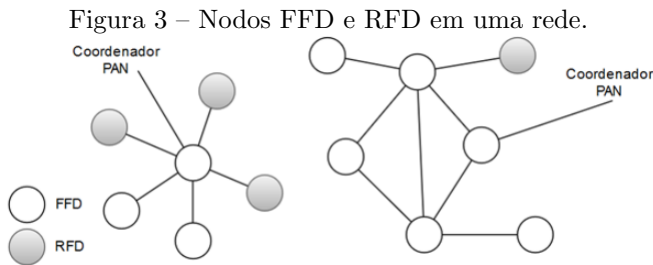
Fonte: (IEEE, 2016).

A camada física fornece o serviço de transmissão de dados, e controla o transmissor de rádio frequência (RF), junto da escolha de canais, manuseio de energia, e manuseio de sinais, operando nas frequências e canais demonstrados pela Tabela 1.

Tabela 1 – Configurações do padrão IEEE 802.15.4.

Frequência	Local	Largura	Canais
2400-2483.5	Mundo	250 Kbps	16
902-928	EUA	40 Kbps	10
868-868,6	Europa	20 Kbps	1

A camada MAC especifica duas classes de dispositivos, FFD (*Full Function Device*) e RFD (*Reduced Function Device*) como demonstrado na Figura 3.



Fonte: Do autor.

**Full function Device:** É um nodo capaz de implementar qualquer topologia, podendo também ser um coordenador PAN ou não, com capacidade de comunicação irrestrita com qualquer nodo da rede e que implementa todos os protocolos (IEEE, 2016).

**Reduced Function Device:** Um nodo de capacidade limitada, que apenas implementa a topologia estrela, ou ponto a ponto, caso seja uma rede com apenas dois nodos. Não é capaz de se tornar o coordenador PAN da rede e implementa uma quantidade reduzida de protocolos (IEEE, 2016).

Uma RSSF que utiliza o padrão IEEE 802.15.4 é formada por, pelo menos, alguns dispositivos de rede (RFD ou FFD); um ou mais coordenadores (FFD) e um Coordenador PAN, que obrigatoriamente deve ser um FFD.

As topologias de rede ainda podem ser combinadas, criando uma rede *p2p* de estrelas interconectadas, ou estrelas de redes, ou uma mis-

tura de ambas, possibilitando uma flexibilização nos casos de uso, e de mais maneiras de implementação.

A camada MAC implementa um modo de operação de baixa potência, em que o dispositivo entra em economia de energia, entre as suas transmissões.

## 2.3 TECNOLOGIAS QUE IMPLEMENTAM O PADRÃO IEEE 802.15.4

Com o padrão IEEE 802.15.4 consolidado, várias empresas desenvolveram seus produtos. A DIGI possui seus módulos XBee Séries 1 como o da Figura 4 e com as configurações descritas na Tabela 2, estes serão os módulos utilizados nos experimentos.

Figura 4 – Um módulo XBee série 1 (DIGI International®).



Fonte: digi.com.

Tabela 2 – Modelos disponíveis da série 1 (DIGI International®).

Protocolo	Distancia	Ganho	Frequência	Hardware
802.15.4	90m	0db dBm	2.4GHz	S1
802.15.4	1.6 km	+18 dBm	2.4GHz	S1
802.15.4	1 km	+10 dBm	2.4GHz	S1

Fonte: digi.com.

### 2.3.1 Modos de comando XBee

Os módulos XBee podem se comunicar de maneira assíncrona com outros dispositivos através de sua porta serial, que utiliza UART - *Universal Asynchronous Receiver/Transmitter*, ou através de uma outra conexão que implementa o mesmo, como RS232 ou interface USB.

Para que possa modificar e ler seus parâmetros, o módulo deve estar primeiro em modo de comando, em que o conjunto de *strings* de

entrada são interpretados de duas maneiras, AT - Modo Transparente e API - Modo de Programa de Aplicação.

**Modo AT:** Modo de operação simplificado, no qual ele apenas repassará o que foi colocado na UART para o módulo remoto.

**Modo API:** Modo em que o interfaceamento através da UART se dá por comunicação através de pacotes que definem as operações e comandos que o módulo XBee executará. O modo API especifica comandos, respostas de comandos e mensagens de status do módulo.

## 2.4 TEORIA DE JOGOS

A teoria de jogos é uma área da matemática usada para modelar fenômenos onde dois ou mais agentes interagem entre si. Um jogo é um padrão matemático que modela a tomada de decisões buscando maximizar seu ganho, ou minimizar seu custo. Estas tomadas de decisões também dependem das decisões tomadas por outros indivíduos. Seu uso é ainda constante na área de economia, onde foi primeiramente aplicada por meio de trabalhos de von Neumann (SCHOUERY et al., 2015).

A Teoria de Jogos algorítmica é a variação computacional dos modelos utilizados na economia, política e modelagem. Nesta área, se investiga, analisa e projeta algoritmos e regras associados ao ambiente, onde as decisões de cada agente neste ambiente, podem ou não interferir no resultado final. Em jogos não cooperativos será necessário encontrar o *Nash equilibrium* que é o estado de equilíbrio que pode ser exemplificado como o dilema do prisioneiro (SCHOUERY et al., 2015). Neste exemplo, dois suspeitos são presos por cometerem crimes menores, porém o detetive não tem provas suficientes para condená-los de seus crimes principais. Ambos interrogados de maneira individual tentando convencê-los a confessar seus crimes principais. Neste caso o detetive fez uma proposta para cada um deles.

Esta situação possui três resultados, primeiro, a confissão de ambos ficando quatro anos presos; a confissão de um deles que diminuirá a sua pena ou se ninguém confessar, ficarão ambos presos por menos tempo. Sem eles poderem se comunicar entre si, devem escolher entre confessar ou não, como cada um dos suspeitos deseja realizar a melhor escolha individual, ambos confessarão seus crimes.

O *Nash equilibrium* é uma situação onde nenhum jogador pode aumentar sua utilidade trocando de estratégia individualmente. Este conceito é essencial para prever qual será o comportamento dos agentes

em uma situação real, mesmo não sendo claro quando os jogadores atingem o equilíbrio, o conceito foi fundamental ao longo dos anos para o desenvolvimento da Teoria de Jogos não cooperativa (SCHOUERY et al., 2015).

Para os jogos cooperativos, não existe uma solução intrínseca como o *Nash equilibrium* e suas vertentes, mas sim um extenso conjunto de soluções que variam do ponto de vista; De maneira geral os jogos se separam então em duas grandes categorias:

**Jogos não cooperativos:** Esta classe de jogos tem foco na utilidade individual de cada jogador, ao invés da utilidade de toda a rede. Nestes jogos não cooperativos, os jogadores agem de maneira individualista, evitando coordenar com os outros participantes, os agentes irão ter um comportamento “egoísta”, porém racional.

**Jogos cooperativos:** Já em aplicações com jogos cooperativos, a rede considera a utilidade de todos os jogadores com o objetivo de maximizar a eficiência de toda a rede, enquanto mantém a imparcialidade. O objetivo da teoria de jogos cooperativos é estudar caminhos para sustentar a cooperação entre os agentes, uma vez que há um custo para manter essa cooperação e ele tem que ser diluído em toda a rede, para de maneira geral a cooperação ser vantajosa.

### 2.4.1 Teoria de jogos em RSSF

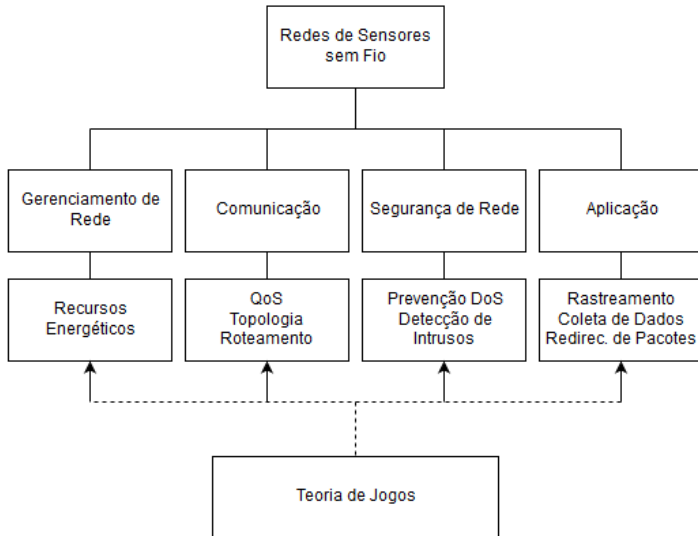
Teoria de jogos é um método matemático que descreve um fenômeno de conflito e cooperação entre entidades. Neste caso, as entidades são nodos de uma RSSF com limitações de recursos e comunicação e através de ações cooperativas ou não, buscar tirar o melhor proveito deste ambiente de recursos limitados. Estas premissas fazem de sua implementação em RSSF uma aplicação interessante e de ótimos resultados (SHI et al., 2012).

A teoria de jogos em RSSF vem sendo aplicada intensivamente na última década, como pode ser visto no livro “*Game Theory for Wireless Engineers*” em que ele apresenta modelos de teoria de jogos e suas aplicações em RSSF (MACKENZIE; DASILVA, 2006). O livro mostra que é uma técnica muito apropriada para RSSF, primeiro porque soluções que utilizam este tipo de rede necessitam ser parcialmente ou completamente distribuídas, outro motivo de ser apropriada é pela escassez de recursos dos nodos (MACKENZIE; DASILVA, 2006).

O desenvolvimento de uma rede sem fio e a otimização de sua performance é um processo complicado e não trivial. Tendo que cum-

prir os requisitos de rede, a teoria de jogos possui o papel de favorecer o desenvolvimento e operação de uma RSSF (AKYILDIZ et al., 2002).

Figura 5 – Integração de RSSF com Teoria de Jogos.



Fonte : (YICK; MUKHERJEE; GHOSAL, 2008).

Os requisitos de uma RSSF, em que geralmente se aplica a teoria de jogos são os seguintes: gerenciamento de redes, comunicação, segurança de rede e aplicação, como mostra a Figura 5.

Na área de gerenciamento de redes é comum utilizar teoria de jogos para controlar os custos de energia de transmissão e recepção de rádio, e o custo energético do sensoriamento, estendendo o tempo de vida da rede e/ou utilizando a energia de maneira igualmente distribuída.

Um dos pontos mais críticos de RSSF, sua comunicação, é alvo de trabalhos que buscam aprimorar questões como a qualidade de serviço, que é aprimorar a tarefa desempenhada pela rede. Também manusear a topologia de rede de maneira mais eficiente, e como será realizado o roteamento entre nodos, uma vez que há um consumo energético para cada transmissão e para cada recebimento de informações, esta é a principal área de teoria de jogos aplicada neste trabalho.

Com uma rede centralizada, dados são coletados e analisados com objetivo de suprir a QoS desejada, seja por meio de otimização de topologia ou por meio de uma implementação própria dos algoritmos

de protocolo de rede.

Enquanto a RSSF está operando, ela pode estar sendo alvo de *hackers*, que tentam interceptar a comunicação, ou executar algum ataque de de DoS - *Denial of Service*, pode ser criada uma técnica de proteção com base na teoria de jogos, que diminuem esse tipo de ataque ou detecta que a rede possui intrusos. E o ultimo elemento demonstrado na Figura 5, aplicação de Teoria de Jogos diretamente na aplicação da rede, otimizando o sensoriamento, coleta de dados ou redirecionamento de pacotes.

#### 2.4.2 Jogos para aprimorar Comunicação sem fio

O trabalho de Dehnie e Memon (2008) analisa os efeitos de comportamentos errôneos, com base em teoria de jogos algorítmica, que faz com que nodos escolham parceiros de confiança. O mecanismo de reputação introduz um ótimo modelo bayesiano, pois tende ao equilíbrio perfeito. Tsuo et al. (2011) usaram uma abordagem com teoria de jogos para modelar um controle de transmissão em uma RSSF que coletava energia. Neste problema, o estado energético de um nodo que coleta carga elétrica do meio varia com maior intensidade, diferentemente de sensores alimentados por bateria. Cada sensor decide a estratégia de transmissão com base em qual estado de energia ele acredita que seus oponentes estejam, sendo este um problema de *nash equilibrium* bayesiano, e os autores Tsuo et al. (2011) demonstraram que cada sensor determina seu limiar para maximizar função de utilidade, e por simulação seu modelo é quase tão eficiente quanto um que possuía todas as informações da rede.

Kondi e Bentley (2010) propuseram um esquema de otimização de múltiplas camadas para RSSF que utiliza DS-CDMA - *Direct Sequence Code Division Multiple Access* aplicando em redes de sensores visuais, um esquema de *nash bargaining solution*, em que os nodos negociam com uma entidade central a alocação de recursos.

#### 2.4.3 Jogos para aprimorar Qualidade de Serviço

QoS é um termo muito abrangente que tem vários significados, e muitas perspectivas, porém a *QoS* da RSSF é de fundamental importância. A rede precisa realizar a tarefa de sensoriamento, maximizando a eficiência de utilização da rede. Desta forma a rede tem

que analisar os requisitos da aplicação, e implementar múltiplos mecanismos de QoS. Ayers e Liang (2011) desenvolveram um algoritmo de controle de QoS chamado Gureen game, que melhora alguns dos problemas do seu antecessor como: péssima imparcialidade e baixa taxa de convergência, eliminando os seus pontos fracos de eficiência energética. Zhou e Mu (2007) propuseram técnicas de aprimoramento de QoS que se ajustam de acordo com a densidade da rede e o formato de implementação, seja retangular, circular ou áreas elípticas em que as redes se adaptavam dinamicamente utilizando o gur game. E por fim, Cui et al. (2011) propuseram uma abordagem para RSSF multimídia que estuda o efeito da potência de transmissão na QoS e na eficiência energética.



### 3 ALGORITMOS DE CONTROLE DE TRANSMISSÃO

Neste capítulo são apresentados os algoritmos de controle de transmissão selecionados na literatura, a configuração do cenário proposto para o experimento, os dispositivos utilizados para o desenvolvimento da RSSF e as métricas utilizadas para avaliar o desempenho dos algoritmos.

#### 3.1 ALGORITMOS AUTÔNOMOS APLICADOS À RSSF

Conforme apresentado nos capítulos anteriores, há diversas aplicações que utilizam nodos redundantes no processo de monitoramento de uma grandeza física. Conseqüentemente, não é necessário que todos os nodos realizem o monitoramento e a transmissão dos dados coletados em todos os ciclos de comunicação da rede. Porém, devido ao funcionamento descentralizado, a ocorrência de falhas nos nodos, os erros de transmissão e outros fatores que afetam uma RSSF, diversos algoritmos autônomicos foram propostos na literatura para selecionar quais nodos devem transmitir em cada ciclo de operação da rede. Neste capítulo, os algoritmos Gur Game, Gureen Game, (m,k)Gur Game e Skip Game são descritos. Além disso, os cenários e métricas que serão utilizados para a avaliação experimental destes algoritmos são discutidos. Por fim, a implementação dos algoritmos em nodos Arduinos® UNO R3 e módulos XBee é apresentada.

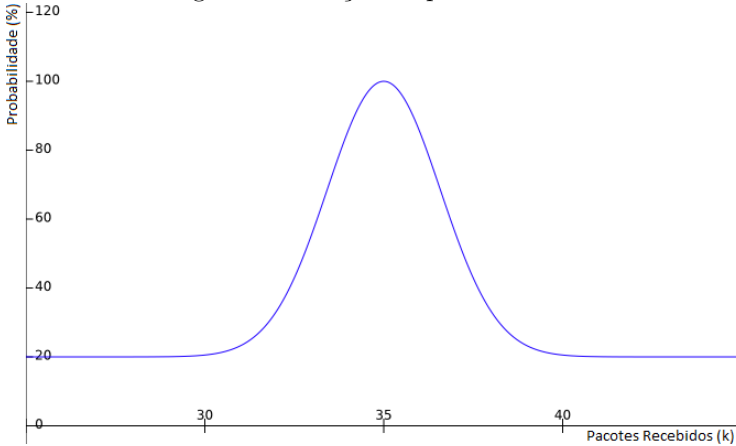
O coordenador funcionará de maneira semelhante em todos os algoritmos, ele se encarregará de contabilizar mensagens e devolver pra rede o resultado da função de recompensa.

#### 3.2 ALGORITMO GUR GAME

O algoritmo Gur Game foi apresentado inicialmente por Tung e Kleinrock (1996). Esse algoritmo foi construído com base na teoria de jogos e considera que há diversos jogadores, nenhum deles ciente dos outros, e um árbitro. A cada segundo, o árbitro solicita que cada jogador vote “sim” ou “não”, calculando na sequência a quantidade de votos dados para cada opção, então, uma probabilidade de recompensa  $p = p(k)$  é determinada como uma função do número de jogadores ( $k$ ) que votaram “sim”. Assume-se que  $0 \leq p(k) \leq 1$ , uma função típica

é ilustrada na Figura 6.

Figura 6 – Função da probabilidade.



Fonte: adaptado de Iyer e Kleinrock (2003).

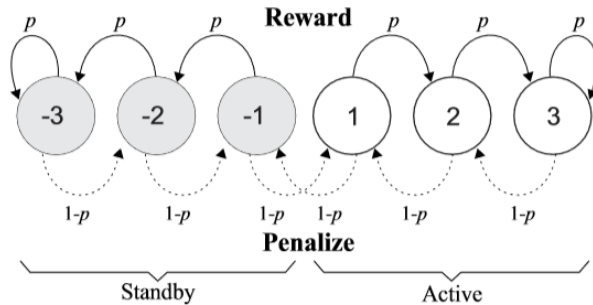
Cada jogador, de acordo com o seu voto, é independentemente recompensado (com probabilidade  $p$ ) ou penalizado (com probabilidade  $1 - p$ ). Por exemplo, assumindo que em um determinado instante de tempo o número de jogadores votando “sim” foi  $k_1$ , a probabilidade de recompensa seria  $p(k_1)$ . Cada jogador é então recompensado com probabilidade  $p(k_1)$ . Note que o número máximo no exemplo representado na Figura 6 ocorre em  $k^*=35$ . Os autores demonstraram que independente da quantidade de jogadores é possível construir uma função em que aproximadamente  $k^*$  deles (neste exemplo, 35) votem “sim” depois de uma quantidade de tentativas.

Neste algoritmo os jogadores não tentam prever o comportamento dos outros jogadores, pelo contrário, cada jogador realiza por tentativa e erro a mesma ação que produz os melhores resultados para aquele jogador.

Para implementar esse esquema, cada jogador precisa ter uma memória de suas tentativas prévias. Mais precisamente, a cada jogador é associado um autômato de estados finitos  $M_j$ , o que representa a memória do jogador. O tamanho total da memória é  $2N$ , para um valor de  $N$  qualquer, iniciando no estado mais à esquerda de  $-N$  até  $-1$ , e então de  $1$  até  $N$  (Figura 7). O jogador pode estar em um estado em cada instante de tempo.

A transição de estados ocorre entre os estados  $j$ ,  $j + 1$  e  $j - 1$

Figura 7 – Autômato de tamanho S.



Fonte: (IYER; KLEINROCK, 2003).

(ou seja, o jogador pode transitar somente para estados adjacentes). Se  $j$  vai para o estado  $N$ , então a única transição permitida é para o estado  $N - 1$ . Além disso, o jogador sempre vota “sim” quando ele está nos estados numerados como positivos e “não” nos estados numerados como negativos. Quando está em um estado negativo, a transição à esquerda acontece quando ele é recompensado pelo árbitro e para à direita quando é penalizado, ou seja, transições para o centro é um comportamento de penalização e para as extremidades de recompensa.

No contexto de uma RSSF, Iyer e Kleinrock (2003) estabelecem um mecanismo onde cada jogador pode ser representado como um Sensor  $S_i$  que tem um autômato  $M_i$  associado. O sensor  $S_i$  transmitirá dados para o coordenador ( $C$ ) quando estiver nos estados numerados como positivo e entrará em modo de *sleep* sempre que estiver nos estados numerados como negativo. Neste caso o coordenador é quem fará o papel de árbitro.

O objetivo do coordenador será sempre receber a quantidade de mensagens considerada como ótima em cada instante de tempo, ou seja, pode-se definir *a priori* esse objetivo. Assume-se que ao receber  $k$  pacotes no instante  $t$  em uma rede com  $S_i$  sensores, com um objetivo de  $o_b$  pacotes, o arbitro consegue calcular o quão próximo está da quantidade considerada ótima utilizando a Equação 3.1.

$$p(k) = 20 + 80e^{(-0.2 \times (\frac{10}{N}) \times (o_b - k)^2)} \quad (3.1)$$

A função de recompensa  $p(k)$  está associada ao coordenador. Em cada instante de tempo  $t$ , o coordenador conta o número de pacotes  $k_t$  que recebeu dos sensores. Então, calcula a probabilidade de recom-

pensa  $p(k_t)$  e envia essa informação para os sensores e cada sensor gerará um número aleatório uniformemente distribuído, gerando um resultado  $r(k_t)$  que é sua probabilidade de recompensar-se ou punir-se com probabilidade  $1 - r(k_t)$ , o Algoritmo 1 representa o funcionamento da máquina de estados dos nodos sensores  $S_i$ .

---

**Algoritmo 1:** Algoritmo do Gur Game

---

```

1 input:  $p(k_t)$ ;
2  $r(k_t) = \text{probabilidade}(p(k_t))$  ;
3 if  $r(k_t) == 0$  then
4   | acao = penalizar;
5 else
6   | acao = recompensar;
7 switch acao do
8   | case penalizar do
9     | muda_p_estado_central();
10  end
11  case recompensar do
12    | muda_p_estado_extremo();
13  end
14 end

```

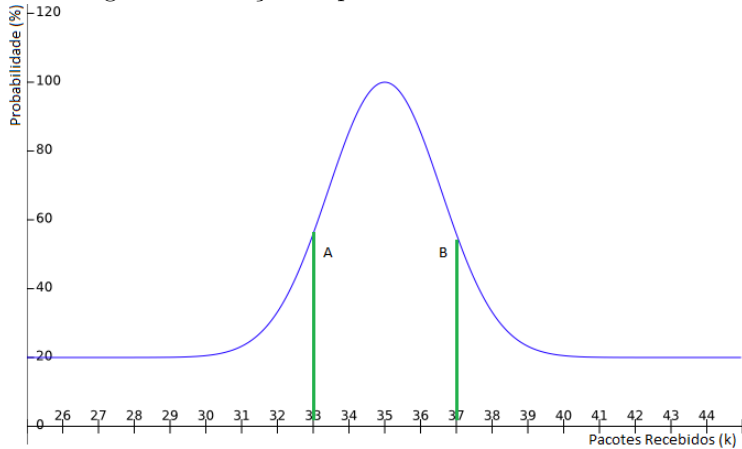
---

### 3.3 ALGORITMO GUREEN GAME

O algoritmo do Gureen Game proposto por Ayers e Liang (2011) utiliza como base o algoritmo Gur Game já citado previamente, aplicando melhorias no seu funcionamento para aprimorar a convergência do jogo e para melhorar sua eficiência energética. O nome Gureen Game foi dado com base na união de duas palavras “*green*” e Gur Game, com intuito de titular o algoritmo como uma forma mais “ecológica” do Gur Game.

A primeira dinâmica implementada é na função de probabilidade  $p$  que o coordenador (C) envia para os jogadores que agora possui um segundo valor além da probabilidade  $p$ , chamado de “bit delta”. Este novo parâmetro traz mais informação para os jogadores, permite que saibam se o número de pacotes está acima ou abaixo do objetivo. Por exemplo: Uma rede que recebeu  $35 \pm 2$  pacotes, o valor de  $p(k)$  será

Figura 8 – Função da probabilidade Gureen Game.



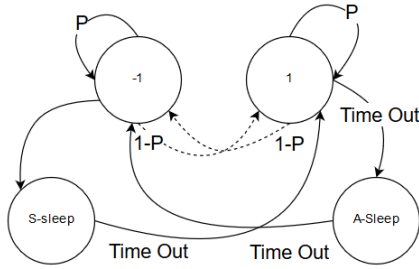
Fonte: (AYERS; LIANG, 2011) adaptado pelo autor.

60 em ambos os casos (A e B da Figura 8), apesar deles estarem em pontos diferentes. Portanto, além da probabilidade  $p(k)$  o coordenador (C) enviará um dado que sinalizará se está acima ou abaixo do objetivo. A Figura 8 ilustra esta nova funcionalidade, em que uma quantidade A e B de pacote resulta em uma mesma probabilidade, mas retornam valores de bit delta diferentes.

Devido ao Gur Game convergir para um estado final, onde estará cumprindo o objetivo corretamente, o coordenador (C) devolverá a seus jogadores, a probabilidade  $p$  com valor 100, os jogadores continuarão ativos até que suas baterias esgotem, ou sofram algum problema. Os agentes inativos continuarão inativos sem executar nenhuma função. Com objetivo de distribuir a tarefa com todos jogadores e evitar que uma pequena quantidade participe e outra fique ociosa, a dinâmica de rotação de jogadores foi implementada. Nesta nova dinâmica os jogadores irão entrar em *sleep* ao ficarem muito tempo participando ou muito tempo parado sem participar. O Autômato  $M_i$  presente em todo agente  $S_i$  agora possui uma forma diferente, apenas com estados 1 e  $-1$  e dois estados de *sleep*, um deles sendo por ficar muito tempo ativo, e o outro muito tempo inativo como visto na Figura 9.

A transição entre os estados  $-1$  e 1 será dada puramente pelo resultado da função de recompensa  $p$ , ao recompensar-se com  $p$  manterá seu estado, e a punição  $1 - p$  transitará entre estados ativo e inativo, porém dependendo do bit delta. A transição para os estados de *sleep* se

Figura 9 – Funcionamento da maquina de estados do Guren Game.



Fonte: (AYERS; LIANG, 2011) adaptado pelo autor.

dará por meio de estouro de um *timer*, assim forçará um tempo máximo em atividade do sensor, e forçará que outros jogadores também participem da atividade, esta dinâmica é chamada de “rotação de jogadores”.

Com a inclusão desta dinâmica de rotação, o coordenador (C) terá que possuir mais algumas informações dos jogadores, como os que estão em *sleep* e os que entrarão em *sleep*, agora os jogadores comunicarão com o coordenador avisando que eles estão fora por algumas rodadas. Com isso, mesmo que o jogo obtenha QoS ótima ele não estará estável, pois a introdução de *sleeps* cíclicos ocasionará uma instabilidade no jogo. Toda vez que um ou mais jogadores fiquem de fora. Para sanar este comportamento turbulento o coordenador será avisado das rotações, e ele atualizará sua função de recompensa  $p(k, n)$  sob demanda em que  $n$  é o número de jogadores ativos, diminuindo um possível *delay* ao buscar a QoS de novo.

Sua implementação em RSSF será semelhante a citada na Seção 3.2, com o diferencial do coordenador (C) além de receber mensagens  $k$ , receberá uma sinalização  $z_i$  de um nodo  $S_i$  que entrará em estado de *sleep* no próximo *round*. E desta forma, ao atualizar a função de recompensa  $p(k, N)$ , além de contabilizar o número de pacotes, o coordenador (C) precisa saber o número  $N$  de sensores  $S_i$  que estarão ativos no próximo *round*, esta modificação pode ser vista na equação 3.2.

$$p(k, N) = 20 + 80e^{(-0.2 \times (\frac{10}{N}) \times (ob - k)^2)} \quad (3.2)$$

Ao receber o resultado da função  $p(k, N)$  do coordenador, assim como o algoritmo Gur Game, o nodo gerará um número aleatório normalmente distribuído e se recompensará com probabilidade  $r(k_t)$  e receberá uma punição com probabilidade  $1 - r(k_t)$ , ao punir-se também

leva em consideração o valor do bit delta recebido junto da probabilidade, demonstrado no Algoritmo 2.

---

**Algoritmo 2:** Algoritmo do Guren Game

---

```

1 input:  $p(k, n)$ ;
2  $r(k)$  = probabilidade  $p(k, N)$ ;
3 if  $r(k) == 0$  then
4 |   acao = penalizar;
5 else
6 |   acao = recompensar;
7 switch acao do
8 |   case penalizar do
9 |     | if estado == ativo and delta == acima then
10 |     |   muda_estado();
11 |     | if estado == ativo and delta == abaixo then
12 |     |   manter_estado();
13 |     | if estado == inativo and delta == acima then
14 |     |   manter_estado();
15 |     | if estado == inativo and delta == abaixo then
16 |     |   muda_estado();
17 |   end
18 |   case recompensar do
19 |     |   manter_estado();
20 |   end
21 end
22 if ativo_por_x_rounds() and inativo_por_x_rounds() then
23 |   sleep()

```

---

### 3.4 ALGORITMO $(M, K)$ GUR GAME

O trabalho de Ramanathan (1999) introduz o escalonador  $(m, k)$ -firme para escalonar tarefas periódicas. De acordo com este modelo, uma tarefa periódica é necessário ter pelo menos  $m$  execuções, em uma janela de  $k$  ativações. Esta representação  $(m, k)$  pode ser tratada como uma proporção de ativações  $(k - m)/k$ . Por exemplo um escalonador  $(1, 2)$ -firme especifica um número mínimo de 50% ativações, no caso em uma janela de tempo  $k = 2$  terá que ter pelo menos 1 ativação. Um *deadline* representado pelo modelo  $(m, k)$ -firme representa o quão

restrita é a tarefa.

Semprebom et al. (2009) apresentou um classificador de tarefas, onde as instâncias são obrigatórias (1) ou opcionais (0). Instâncias obrigatórias terão seus *deadlines* garantidos e instâncias opcionais irão receber prioridades menores sem garantias de serem executadas. As execuções classificadas como obrigatória e opcionais são baseadas nos valores  $m_i$  e  $k_i$  de cada tarefa, então uma instância de uma tarefa  $\tau_i$  ativada no instante  $w.T_i$  será classificada como obrigatória se satisfizer a equação 3.3, senão será opcional,  $t$  representa o tempo.

$$w = \left\lfloor \left\lceil \frac{wm_i}{k_i} \right\rceil \times \frac{k_i}{m_i} \right\rfloor, \text{ para } \rightarrow w = \left\lfloor \frac{t}{T_i} \right\rfloor \quad (3.3)$$

Por exemplo, uma tarefa (2,3)-firm possui suas ativações  $w = 0, 1, 2$  classificadas como obrigatória, obrigatória e opcional, respectivamente. Este padrão se repetirá a cada três ativações ( $k = 3$ ).

Em uma abordagem de RSSF, o modelo constituirá de um coordenador (C) recebendo mensagens periódicas enviadas pelos nodos escravos, e a cada período de tempo chamado de *round*, o coordenador devolverá uma probabilidade  $p$  como o Gur Game e Gureen.

A definição  $(m, k)$ -firm é usada como padrão de transmissão em cada nodo, por exemplo um nodo configurado com o padrão (2, 5)-firm enviará 2 mensagens a cada 5 rounds, os nodos usarão a Equação 3.3, classificando seus envios de mensagens como opcional ou obrigatória.

O autômato será o mesmo da Figura 7, em que os estados  $-1$  e  $1$  representam “enviar mensagens se for uma ativação obrigatória”, porém os estados centrais irão implementar uma nova ação ao penalizar-se, nesse caso os parâmetros  $(m, k)$ -firme associados ao nodo se ajustarão conforme Algoritmo 3.

Neste algoritmo é possível observar algumas dinâmicas implementadas, onde é possível observar a implementação do mesmo bit delta visto no Gureen Game de Ayers e Liang (2011), que conterà a informação se a rede está acima ou abaixo do objetivo desejado. A próxima dinâmica é que os nodos terão a informação de quanta energia eles possuem em suas baterias, as variáveis do padrão  $(m, k)$  de envios serão ajustadas conforme a energia restante. Esta se torna a maior diferença perante ao Gur Game, ao alcançar o objetivo de rede desejado, apenas o último conjunto de nodos continuará transmitindo, até que eles esgotem suas baterias ou aconteça alguma outra falha, já o  $(m, k)$ Gur Game distribuirá o envio de mensagens com todos os nodos, realizando um melhor aproveitamento energético, com uma maior diversidade espacial, o Algoritmo 3 demonstra o seu funcionamento.



---

**Algoritmo 3:** Algoritmo (m,k)Gur Game
 

---

```

1 input: acao, estado, bitdelta, m , k , energia  $r(k)$  =
   probabilidadep(k);
2 if  $r(k) == 0$  then
3   | acao = penalizar;
4 else
5   | acao = recompensar;
6 switch acao do
7   case penaliza do
8     | muda_para_estado_mais_central();
9     | if estado_central() == TRUE then
10      | switch bitdelta do
11        | case ACIMA do
12          | | if  $m > 1$  then
13            | | | m = m - 1;
14          | | else
15            | | | k = k + 1;
16          | | end
17        | case ABAIXO do
18          | | if  $m < k$  then
19            | | | m = m + 1;
20          | | else
21            | | | if  $m > 1$  then
22              | | | | m = m - 1
23            | | | else
24              | | | | k = k + 1
25            | | end
26          | end
27        | end
28   case Recompensar do
29     | mover_p_estado_extremo();
30     | if ativacao_mandatoria then
31       | | transmitir_msg();
32     | if energia < 20% e bitdelta == acima then
33       | | if  $m > 1$  then
34         | | | m = m - 1;
35       | | else
36         | | | k = k + 1 ;
37       | | end
38     | end
39 end

```

---

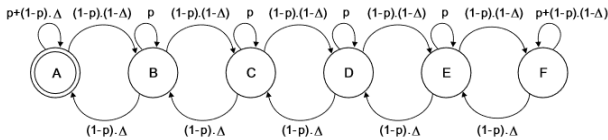
Pela maneira com que  $(m, k)$ Gur Game altera seus valores  $m$  e  $k$  isto fará com que ele possua uma quantidade enorme de possíveis estados.

### 3.5 ALGORITMO SKIP GAME

Uma das mais recentes abordagens de algoritmos autônomos, o Skip Game de Semprebom et al. (2015) utiliza das técnicas de Kay e Frolik (2008) de ACK para que a transmissão dos nodos siga um padrão específico, entrando em *sleep* durante estas transmissões, permitindo um incremento ainda maior de economia de energia perante as outras abordagens. Cada nodo possui sua máquina de estados finita autônoma e distribuída, onde os nodos trabalham junto do coordenador (C) tentando alcançar os objetivos da aplicação.

Cada Sensor  $S_i$  possui um autômato  $M_i$  de estados finitos, a Figura 10 exemplifica uma máquina de 6 estados. Caso os nodos não estejam em um dos estados extremos ele se recompensará com probabilidade  $p$  ou penaliza-se com probabilidade  $1 - p$ , movendo-se para à esquerda ou para à direita. Assim como os outros algoritmos já citados, estas condições de penalização ou recompensa são enviadas pelo Coordenador (C), para todos os nodos da rede, utilizando da Equação 3.2, também implementando o bit delta, os nodos saberão se estão acima ou abaixo do objetivo da rede, e caso sejam penalizados, se penalizam de acordo com o estado da rede informado pelo bit delta.

Figura 10 – Máquina de estados do Skip Game.



Fonte: (SEMPREBOM et al., 2015).

Cada estado do autômato representa um padrão específico e bem determinado de ativações obrigatórias e opcionais. A ideia desta sequência é se comportar como  $(m, k)$ -firme ou uma sequência *skip over* (SEMPREBOM et al., 2013), (KOREN; SHASHA, 1995), onde os nodos transmitem durante períodos mandatórios e desligam-se durante períodos opcionais.

Os padrões de transmissão são desenvolvidos previamente de ma-

neira arbitrária. A tabela 3 exemplifica uma configuração que pode ser aplicada em cada estado do Skip Game, o estado inicial A representa uma execução (1, 1)-firme, significa que o nodo envia mensagens a todo período. O estado F representa a menor taxa de transmissão com 1 transmissão a cada 50 períodos uma constante (1, 50)-firme.

---

**Algoritmo 4:** Algoritmo Skip Game

---

```

1 input:  $p(k)$ , bitdelta
2  $r(k_t) = \text{probabilidade}(p(k_t))$ 
3 if  $r(k) == 0$  then
4 |   acao = penalizar;
5 else
6 |   acao = recompensar;
7 switch acao do
8 |   case penaliza do
9 |       if bitdelta == aumenta then
10 |           if not estado mais esquerda then
11 |               | muda_estado_p.esquerda();
12 |           else
13 |               | mantem_estado()
14 |           else
15 |               if not estado mais direita then
16 |                   | muda_estado_p.direita();
17 |               else
18 |                   | mantem_estado()
19 |           end
20 |   case Recompensar do
21 |       | manter_estado();
22 |   end
23 end
24 sleep_ate_prox_ativacao();

```

---

Quando um nodo penaliza-se, o valor do bit delta é usado para melhorar a performance do algoritmo. Este valor representa a probabilidade de um nodo aumentar seu número de transmissões, ou seja, a probabilidade de mover-se para o estado mais a esquerda, caso contrário o valor  $(1 - \Delta)$  representa a probabilidade de se mover para o estado mais a direita.

O Algoritmo 4 descreve o seu funcionamento.

Tabela 3 – Estados designados do skip game.

Estado	(m,k)-firm limiar	Mensagens enviadas
A	(1,1)-firm	100%
B	(1,2)-firm	50%
C	(1,3)-firm	33%
D	(1,4)-firm	25%
E	(1,5)-firm	20%
F	(1,50)-firm	2%

### 3.6 CONFIGURAÇÃO DOS ALGORITMOS

Os algoritmos foram configurados considerando seus parâmetros individuais. A Tabela 3.6 mostra os parâmetros selecionados para os testes, o número de estados foi arbitrado com base nos dados adquiridos segundo a análise da Figura 11. Por exemplo, o Gur Game apresentou um melhor comportamento com 12 estados, 6 ativos e 6 inativos, sem entrar no modo sleep e sem o bit delta. Já o Gureen Game funciona com apenas 2 estados, porém depende de um bit delta para executar a mudança de estados, seu *sleep* se baseia em um estouro de *timer* ao ficar muito tempo ativo/inativo. O algoritmo (m,k)Gur Game usa também do bit delta, porém não realiza o escalonamento de *sleep* e, por fim, o Skip game pode ser entendido como aquele que utiliza todas as técnicas citadas.

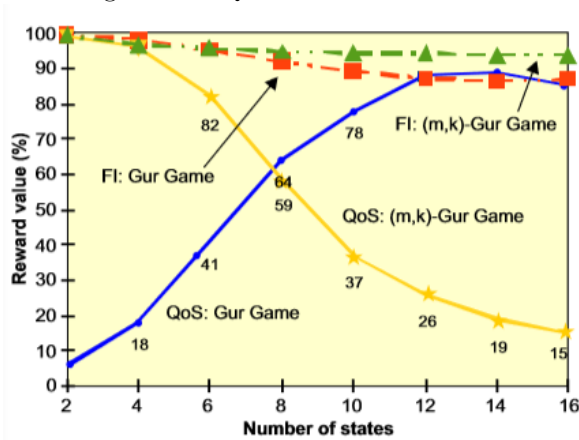
Tabela 4 – Algoritmos e suas configurações.

Algoritmo	Nº de estados	Bit delta	Sono
Gur Game	12	Não	Não
Gureen Game	2	Sim	Time-out(50/100) rounds
(m,k)Gur Game	14	Sim	Não
Skip Game	6	Sim	Cíclico

### 3.7 MÉTRICAS

Para avaliar os algoritmos, um conjunto de métricas foram selecionadas, sendo elas: QoS, imparcialidade, nodos mortos, nível de bateria. Estas métricas foram escolhidas com base nos trabalhos prévios de Semprebom et al. (2015) e Ayers e Liang (2011).

Figura 11 – QoS vs número de estados.



Fonte: (SEMPREBOM et al., 2015).

### 3.7.1 QoS - Quality of Service

A métrica de *QoS* das redes dependerá do objetivo da rede em mensagens por *round*, ela poderá estar acima ou abaixo do esperado, abaixo estará tendo performance reduzida, e acima estará gastando recursos sem necessidade.

$$QoS = 1 - \left| \frac{k - o_b}{N} \right| \quad (3.4)$$

Sendo  $k$  o número de mensagens recebidas,  $o_b$  como objetivo, e  $N$  a quantidade de nodos. A métrica de qualidade de serviço indicará se a rede consegue manter o número de mensagens perto do objetivo (mensagens/*round*) estabelecido pelo coordenador.

### 3.7.2 Imparcialidade

A métrica de imparcialidade será responsável por analisar como está a distribuição do algoritmo, e pode ser interpretado, conforme a Tabela 5. Segundo Jain (1991) é possível através da equação 3.5 deduzir a imparcialidade da rede, desde que seja monitorado o fluxo de pacote

de todos os nodos.

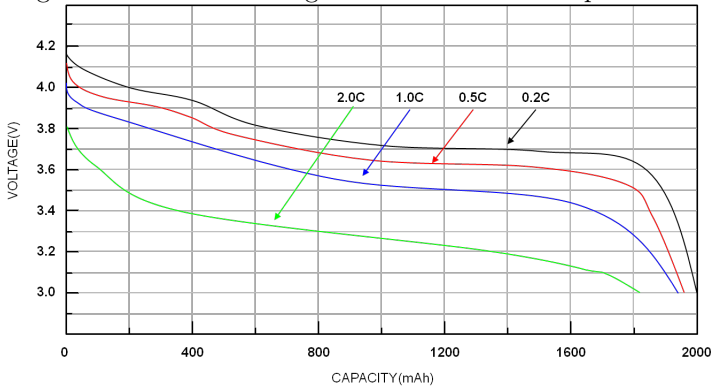
$$f(Txnode_1, Txnode_2, \dots, Txnode_n) = \frac{(\sum Txnode_i)^2}{N * (\sum Txnode_i)^2} \quad (3.5)$$

Sendo  $Txnode$ , a quantidade de mensagens enviadas por cada nodo, e  $N$  a quantidades de nodos da rede.

### 3.7.3 Consumo de energia

Ao começo e final de cada experimento, a tensão das baterias de LiPo - *Lithium Ion Polymer* de cada nodo será medida, e com base no funcionamento das mesmas, a carga usada será estipulada.

Figura 12 – Curva de carga de uma bateria de Lipo comum.



Fonte: fonte: [www.richtek.com](http://www.richtek.com).

Os experimentos realizados terão um comportamento idêntico a da curva de carga com valor de 0.2C pois este é o nível de consumo que os nodos terão durante os experimentos.

### 3.7.4 Nodos Mortos

Com o decorrer do experimento, poderá haver ou não, nodos que terão suas baterias esgotadas. Dependendo do tempo de simulação e da quantidade de nodos mortos, essa métrica poderá dizer o quão eficientemente a bateria dos nodos foram utilizadas, em relação a quantidade

Tabela 5 – Valores Exemplo de imparcialidade.

Valor Txnodo	Imparcialidade
100 Valores Iguais	1.0
99 Valores iguais e um com valor 10 vezes maior	0.6
99 Valores iguais e um com valor 100 vezes maior	0.04
35 Variáveis com valor 4000 e 65 com valor 50	0.37

média de bateria da rede por exemplo.





## 4 EXPERIMENTOS

Neste capítulo se apresenta o cenário dos testes e discute-se os resultados dos múltiplos experimentos, analisando os dados obtidos para as métricas previamente citadas, por fim com dados prévios da literatura.

### 4.1 DESCRIÇÃO DO CENÁRIO

Este trabalho tomará como referência o cenário proposto por Semprebom et al. (2015), no entanto com proporções reduzidas. Semprebom et al. (2015) realizou simulações utilizando o simulador OMNeT++ em que uma rede composta por radios CC2420 que implementam o padrão IEEE 802.15.4. Os parâmetros que foram utilizados nas simulações são apresentados na Tabela 6.

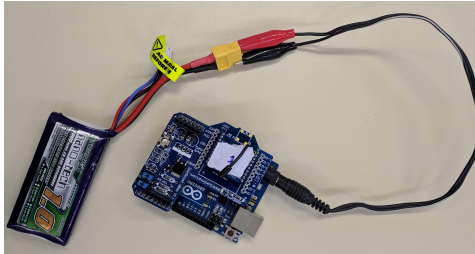
Tabela 6 – Parâmetros da simulação no OMNeT++.

Parâmetro	Valor
Tempo de simulação	2000 segundos
Número de nodos	100
Raio do círculo	90 metros
Potência de transmissão	1.1 mW
Sensibilidade	-100 dBm
Tamanho do pacote	240 bits
Tamanho do cabeçalho	48 bits
Taxa de bits	250 kbps
<i>Beacon Interval</i>	491.5 ms
<i>Beacon Order</i>	BO = 5
<i>Superframe Order</i>	SO = 4

Neste trabalho, o cenário terá sua escala reduzida devido a limitação de recursos. Na implementação utilizar-se-á nodos compostos de um microcontrolador Arduino®<sup>®</sup>, que se trata de uma plataforma de prototipação de código aberto, em conjunto com módulos XBee, em que ambos estão conectados através de um *shield*, este módulo implementa as camadas física e MAC do padrão IEEE 802.15.4. O código dos algoritmos foram implementados utilizando C++ através do *framework* do Arduino®<sup>®</sup>, este código se comunica através do modo API

de comunicação por meio de pacotes especiais. Estes nodos alimentados por uma bateria de LiPo - *Lithium ion Polymer* de duas células. Esta bateria é conectada no conector P4 do Arduino® , que através do controlador de tensão, transformará a operação da bateria que varia de 6.4 Volts a 8.4 Volts, para uma tensão constante de 5V para alimentar o microcontrolador, o exemplo de um nodo em funcionamento pode ser visto na Figura 13.

Figura 13 – Nodo conectado à bateria.



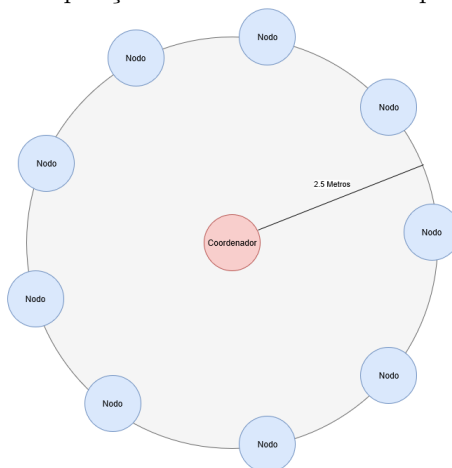
No cenário utilizado foram posicionados 10 nodos XBee, sendo um deles o coordenador, os outros nove nodos foram posicionados em uma área radial com 5 metros de diâmetro. Como se trata de uma área pequena, os nodos foram configurados para ter a mínima potência de transmissão, os módulos XBee podem ser configurados em um mínimo de -5dBm (0.31 mW) e um máximo de +5dBm (1.1w). A configuração do tamanho dos pacotes foi reduzida para 2 bytes, sem contar o cabeçalho padrão, pois devido ao tamanho limitado de *buffer* do microcontrolador Arduino® de apenas 64 bytes, o coordenador em situações com alto *throughput* poderia entrar em estado de *overflow*. A simulação utiliza os nodos em modo *beacon*, porém a série 1 do XBee que foi utilizada nos experimentos não possui esta capacidade, então o controle de rounds serão realizado via software, por meio do *timer* interno do coordenador da rede, e os pequenos ciclos entre os rounds também são controlados pelos respectivos *timers* internos dos nodos.

A Tabela 7 mostra a configuração do cenário experimental, sendo possível comparar com a Tabela 6 que contém os dados da simulação, a Figura 14 mostra o local que os testes foram realizados. Cada experimento de 600 segundos foi repetido 10 vezes.

Tabela 7 – Parâmetros da experimentação.

Parâmetro	Valor
Tempo de experimento	600 segundos
Número de nodos	10
Raio do círculo	2.5 metros
Potência de transmissão	0.31 mW
Sensibilidade	-100 dBm
Tamanho do pacote	16 bits
Tamanho do cabeçalho	48 bits
Taxa de bits	250 kbps
Intervalo <i>Round</i>	1000 ms

Figura 14 – Disposição radial dos nodos na experimentação.

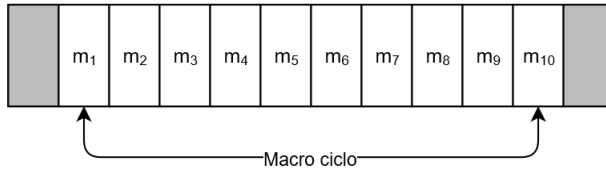


#### 4.1.1 Modelo de comunicação

O modelo de comunicação proposto por Semprebom et al. (2015) divide o período de comunicação em intervalos de macro ciclos e de micro ciclos. Macro ciclo é como o autor denomina o intervalo entre beacons, ou seja, um macro ciclo representa um *round* de comunicação na rede. A cada macro ciclo, o coordenador contabiliza os pacotes recebidos e envia para os nodos escravos o valor da probabilidade  $p$ . Já

os micro ciclos representam o período em que os nodos estarão transmitindo mensagens, ou seja, cada micro ciclo representa o *timeslot* utilizado por um nodo para realizar uma transmissão, como apresentado na Figura 15.

Figura 15 – Modelo de comunicação do experimento.

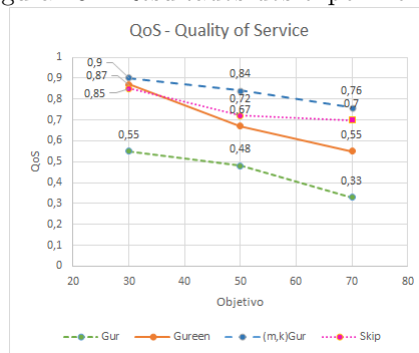


## 4.2 RESULTADOS

Esta seção está dividida em 3 partes, a primeira aborda os resultados da QoS de cada algoritmo, a segunda analisa os resultados de imparcialidade e, por fim, uma discussão acerca do uso energético de cada algoritmo é realizada.

### 4.2.1 QoS

Figura 16 – Resultados dos experimentos.

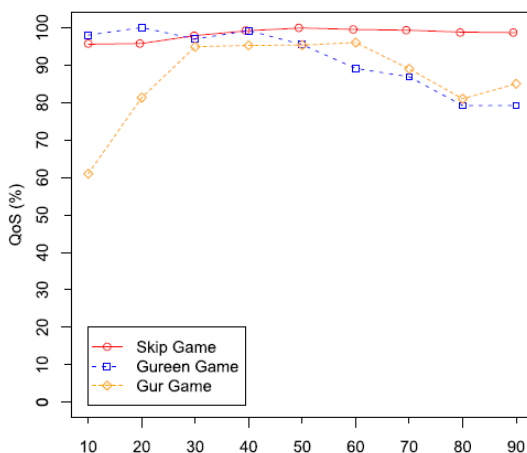


Como pode ser observado na Figura 16, ao se definir o objetivo de 30% da capacidade, todos os algoritmos possuem uma performance

parecida, sendo capazes de prover uma boa QoS, exceto o Gur Game que alcança uma quantidade relativamente baixa. Já com 50%, os algoritmos que implementam *sleeps* cíclicos se mostram inferiores ao (m,k)Gur Game, o algoritmo Skip Game que antes estava um pouco abaixo do Gureen Game em QoS, agora se encontra levemente acima. A principal razão do Gur Game ser incapaz de manter sua QoS está relacionada com a falta do bit delta que faz com que ele falhe em convergir para o objetivo, técnica que está presente em todos os outros algoritmos. Ao definir o objetivo de 70%, todos os algoritmos tiveram perda de performance, principalmente o Gur Game, devido ao problema anteriormente relatado, e também devido a outros fatores como perda e colisão de pacotes, então algoritmos como Skip Game e (m,k)Gur Game que possuem as ativações de envio menos sincronizadas, tiveram performance melhor devido a menos colisões.

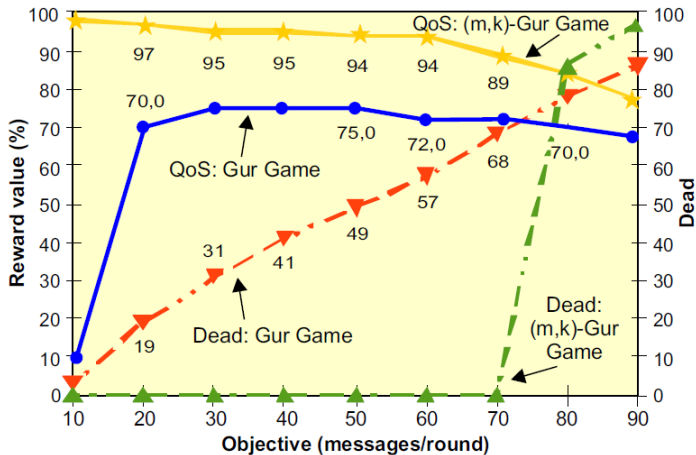
Em comparação com os resultados da simulação de Semprebom et al. (2015) ilustrados na Figura 17, o Skip Game obteve melhor performance de maneira geral, ficando um pouco inferior ao Gureen Game com objetivos de 10 e 20, se mantendo próximo do máximo e no trabalho de Semprebom et al. (2013) representado pela Figura 18, temos um comparativo do Gur Game com o (m,k)Gur Game, em que mostra a sua prevalência em todos os objetivos de rede definidos e decaindo de maneira gradual com o aumento do objetivo.

Figura 17 – Resultados Simulação.



Fonte : (SEMPREBOM et al., 2015).

Figura 18 – Resultados Simulação.



Fonte: Semprebom et al. (2013).

Além da QoS média dos algoritmos também se obteve o desvio padrão da QoS, que é apresentada na Tabela 8. Apesar de possuir uma QoS superior a do Gur Game, o Gureen game possui um desvio padrão maior, isto se deve ao fato de sua dinâmica de rotação em que nodos constantemente entram e saem do estado de *sleep*. Este mesmo efeito do *sleep* pode ser visto ao comparar os desvios padrões do (m,k)Gur Game com Skip Game.

Na Tabela 9 é possível notar a performance instável do Gur Game, que apesar de não possuir uma performance muito inferior ao (m,k)Gur Game com objetivo de 70% tem um desvio padrão de QoS maior do que a dos demais algoritmos, isso mostra o seu comportamento imprevisível, em contrapartida ambos Skip Game e Gureen Game estão extremamente estáveis.

Tabela 8 – Desvio padrão da QoS entre os testes experimentais.

Algoritmo	30%	50%	70%
Gur Game	2.353%	2.136%	1.349%
Gureen Game	2.624%	2.032%	2.21%
(m,k)Gur Game	0.973%	1.076%	0.778%
Skip Game	1.247%	1.378%	1.428%

Tabela 9 – Desvio padrão da QoS entre as simulações.

Algoritmo	30%	50%	70%
Gur Game	4.771%	0.577%	4.24%
Gureen Game	0.017%	0.005%	0.006%
Skip Game	0.010%	0.003%	0.004%

Fonte Semprebom et al. (2015)

#### 4.2.2 Imparcialidade

Imparcialidade demonstra como que a tarefa de monitoramento é distribuída pelos nodos da rede. E, por consequência, como o uso energético da rede está distribuído entre os nodos da rede. Semprebom et al. (2013) utilizou desta métrica para avaliar o (m,k)Gur Game e Gureen Game, mas no entanto, este trabalho considerará esta métrica para avaliar todos os algoritmos.

Tabela 10 – Imparcialidade média dos testes.

Algoritmo	30%	50%	70%
Gur Game	42,4%	57,7%	69,1%
Gureen Game	76,4%	85,9%	89%
(m,k)Gur Game	89,6%	94,7%	96,3%
Skip Game	73,0%	82,4%	85,2%

A Tabela 10 ilustra como a dinâmica de rotação de jogadores do Gureen Game ajuda a reduzir a imparcialidade da rede de maneira direta, e como a distribuição da imparcialidade é excelente no algoritmo do (m,k)Gur Game, este obteve um resultado parecido com o observado na figura 18 em que os resultados se conferem de maneira parecida, por fim o Skip Game possui uma imparcialidade não muito diferente da apresentada no Gureen Game.

#### 4.2.3 Consumo energético

Efetuando a medição de todos os nodos no começo e no fim dos testes, foi possível verificar quais que tiveram maior consumo energético e o uso médio de energia de cada nodo. A Tabela 11 mostra o consumo médio de energia da rede, o nodo que consumiu mais energia com o que consumiu menos.

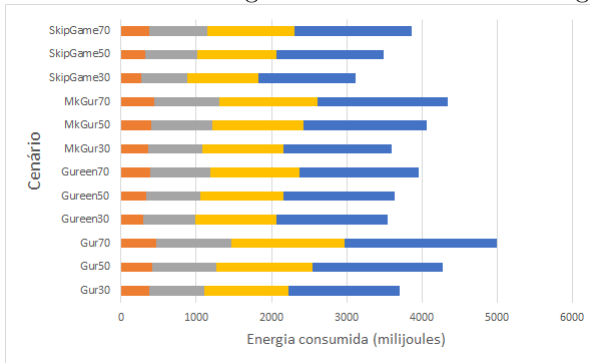
Tabela 11 – Consumo de energia para objetivo em 30%.

Algoritmo	Consumo médio	Maior	Menor
Gur Game	16,6 mAh	20 mAh	13 mAh
Gureen Game	13,5 mAh	16 mAh	10 mAh
(m,k)Gur Game	16,6 mAh	14 mAh	7,5 mAh
Skip Game	12 mAh	14 mAh	9,5 mAh

É possível observar que a aplicação de *sleep* cíclico é primordial para fazer com que a rede possua uma maior eficiência energética no geral, porém isso não garante com que exista consumo de energia igualitário para cada nodo.

O consumo de cada configuração de algoritmo pode ser visualizado na Figura 19, em que há uma expressão geral do consumo total de energia, em milijoules.

Figura 19 – Consumo energético da rede em cada configuração



Os resultados apresentados na Figura 19 demonstram o consumo energético de toda a rede, sendo possível observar que o Gureen game é de 10% até 20% mais eficiente energeticamente que o Gur Game. Apesar de não implementar dinâmicas de *sleep*, (m,k)Gur Game obteve uma melhora no consumo energético, principalmente com objetivo em 70%. O Skip Game é 3% mais eficiente que o Gureen Game, sendo o algoritmo com melhor aproveitamento em todos os objetivos.

Durante os experimentos, após um certo tempo em que as baterias estavam com um nível de tensão abaixo de 3.8V por célula, significando pouca carga restante, foi executado um experimento para testar a eficiência energética dos mesmos, o resultado deste teste pode ser



visto na Tabela 12. Mais uma vez sendo possível observar as vantagens de uma dinâmica de *sleep* e como a imparcialidade pode ajudar a rede a se manter por mais tempo.

Tabela 12 – Número de nodos mortos experimento.

Algoritmo	30%	50%	70%
Gur Game	2	3	4
Gureen Game	0	0	1
(m,k)Gur Game	1	2	3
Skip Game	0	0	1

Realizando uma comparação com os resultados de nodos mortos de Semprebom et al. (2015), podemos ter uma conclusão parecida por meio da Tabela 13, assim que se aumenta o objetivo da rede, maior é o consumo energético, porém é possível observar a baixar imparcialidade com nodos morrendo em objetivos baixos.

#### 4.3 DIFICULDADES ENCONTRADAS

Algumas dificuldades foram encontradas durante a realização do TCC. A primeira dificuldade foi implementar os algoritmos nos nodos, uma vez que se tratam de Arduinos® conectados a *shields* que implementam o padrão IEEE 802.15.4. A algumas funcionalidades como o escalonamento de *sleep* tiveram de ser implementadas de uma maneira compatível, já que o Arduino® possui uma configuração de *sleep* de pouco tempo, ao estarem em *sleep* os nodos na verdade estavam realizando pequenos *mini-sleeps*. Outro fator é que o *clock drift* do Arduino® chega a mais de 2%, isso já é o suficiente para que os últimos pacotes de um *round* não sejam contabilizados pelo coordenador em alguma das vezes, e o *clock* interno do *watchdog* possui uma variação ainda maior de até 15% e este *timer* controla o *sleep* do Arduino®.

O modo *beacon* em que os trabalhos previamente realizados uti-

Tabela 13 – Número de nodos mortos nas simulações.

Approach	10	20	30	40	50	60	70	80	90
Skip Game	0	0	0	0	0	0	0	9	12
Gureen Game	0	0	0	6	6	10	13	14	17
Gur Game	3	5	31	40	52	59	58	57	59

lizaram não é compatível com a tecnologia de nodo que foi utilizada neste experimento, fazendo com que na verdade, o *beacon interval* seja apenas o estouro do *timer* do coordenador, esta implementação em software causa uma perda de performance.

Houve uma limitação ao adquirir dados da rede, como perda de pacotes devido a nível de bateria baixo dos nodos e colisão dos pacotes. Para avaliar a métrica de imparcialidade é necessário saber o número de pacotes enviados por cada nodo escravo, ou seja, o nodo coordenador necessitaria da informação individual de cada nodo. Este tipo de controle não foi implementado devido a uma limitação de tempo. Desta forma foi utilizado o número de mensagens recebidas pelo coordenador para a análise desta métrica.

## 5 CONCLUSÃO

Com o cenário de avaliação experimental desenvolvido, verificou-se que é factível implementar técnicas de controle de transmissão baseadas em Teoria de Jogos e que os resultados são positivos, mesmo que diferentes dos resultados obtidos por simulação. Desta forma, a ideia por trás do controle de transmissão é viável e possui algumas implementações simples com resultados satisfatórios.

No decorrer da execução dos testes previa-se um experimento de maior escala, com um número maior de algoritmos e de nodos, mas devido ao tempo necessário para realizar um conjunto de testes e do tempo recomendado para recarregar cada bateria de lítio, fez com que o experimento se reduzisse a uma escala menor. Mesmo que os algoritmos originalmente sejam designados para redes de larga escala, os experimentos foram realizados com sucesso.

Os resultados obtidos através do experimento demonstraram que dos algoritmos comparados, o Skip game tem a melhor eficiência energética, obtendo também a segunda melhor imparcialidade e QoS. O  $(m,k)$ -Gur Game obteve a melhor imparcialidade e QoS em todos os níveis de objetivo, mas com consumo elevado de energia. O algoritmo Gureen Game apesar de ser superado pelas propostas modernas em QoS com objetivos mais altos e em consumo energético, ainda é um algoritmo válido pela simplicidade de implementação e ganhos de eficiência energética quando comparado ao Gur Game, ou seja, ainda consegue obter resultados satisfatórios. A aplicação simplória do Gur Game é capaz de entregar QoS razoável em níveis de objetivo mais baixos, ao custo de uma rede muito parcial e muito ineficiente de maneira energética.

No entanto ao realizar uma comparação direta dos resultados obtidos nos experimentos, com os resultados obtidos nas simulações, observou-se que nos resultados obtidos nas simulações os algoritmos se comportaram em sua grande maioria de maneira ótima, com mínima QoS de 90%, ao contrário dos resultados experimentais, com isso, é possível dizer que a diferença de benefícios apresentadas pelas técnicas em ambiente real é ainda maior do que aquelas apresentadas pelas simulações.

Por fim, mesmo diante das dificuldades do experimento, provou-se que é factível implementar estes algoritmos. Demonstrando as suas diferenças de performance em relação a QoS, imparcialidade e desvio padrão de QoS. Desta forma o objetivo de avaliar experimentalmente

estes algoritmos foi realizado com sucesso.

## 5.1 TRABALHOS FUTUROS

Como trabalhos futuros sugere-se a análise de outras métricas, como perda e colisão de pacotes, na qual permitirão uma análise mais completa dos algoritmos.

Além disso ainda há uma grande gama de algoritmos de RSSF que aplicam outras estratégias computacionais como por exemplo, teoria de jogos cooperativa, aplicações bioinspiradas ou algoritmos genéticos, visando melhorar algum ponto de funcionamento deste tipo de rede. Na literatura encontram-se muitas propostas que ainda não foram testadas experimentalmente em ambientes reais. Desta forma, como trabalhos futuros também sugere-se a implementação experimental destas outras propostas, comparando-os com as algoritmos testados neste trabalho.

## REFERÊNCIAS

- AKYILDIZ, I. F. et al. A survey on sensor networks. **IEEE Communications Magazine**, v. 40, n. 8, p. 102–105, 2002. ISSN 01636804.
- AYERS, M.; LIANG, Y. Gureen Game: An energy-efficient QoS control scheme for wireless sensor networks. **2011 International Green Computing Conference and Workshops**, p. 1–8, 2011.
- BURATTI, C. et al. An overview on wireless sensor networks technology and evolution. **Sensors**, Molecular Diversity Preservation International, v. 9, n. 9, p. 6869–6896, 2009.
- CUI, H. X. et al. A game theoretic approach for power allocation with QoS constraints in wireless multimedia sensor networks. **MULTIMEDIA TOOLS AND APPLICATIONS**, 2011.
- CURIAC, D. I. et al. Redundancy and its applications in wireless sensor networks: A survey. **WSEAS Transactions on Computers**, v. 8, n. 4, p. 705–714, 2009. ISSN 11092750.
- DEHNIE, S.; MEMON, N. Cooperative diversity as a dynamic game with incomplete information. **IEEE Military Communications Conference**, 2008.
- DEIF, D. S.; GADALLAH, Y. Classification of wireless sensor networks deployment techniques. **IEEE Communications Surveys and Tutorials**, 2014. ISSN 1553877X.
- DIETRICH, I.; DRESSLER, F. On the lifetime of wireless sensor networks. **ACM Transactions on Sensor Networks**, 2009. ISSN 15504859.
- GIL, A. C. O projeto na pesquisa fenomenológica. **Anais IV SIPEQ**, 2010. ISSN 0006-2960.
- HORNEBER, J.; HERGENRODER, A. A survey on testbeds and experimentation environments for wireless sensor networks. **IEEE Communications Surveys and Tutorials**, v. 16, n. 4, p. 1820–1838, 2014. ISSN 1553877X.

IEEE. IEEE Standard for Low-Rate Wireless Networks. **IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)**, 2016. ISSN 03952649.

IYER, R.; KLEINROCK, L. {QoS} control for sensor networks. **{IEEE} International Conference on Communications, 2003. {ICC} '03**, 2003. ISSN 05361486.

JAIN, R. **The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling**. [S.l.: s.n.], 1991. ISSN 03029743. ISBN 978-0-471-50336-1.

JIA, J. et al. Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm. **Computers and Mathematics with Applications**, Elsevier Ltd, v. 57, n. 11-12, p. 1756–1766, 2009. ISSN 08981221. Disponível em: <<http://dx.doi.org/10.1016/j.camwa.2008.10.036>>.

KAY, J. M.; FROLIK, J. An expedient wireless sensor automaton with system scalability and efficiency benefits. **IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans**, 2008. ISSN 10834427.

KONDI, L. P.; BENTLEY, E. S. GAME-theory-based cross-layer optimization for wireless DS-CDMA visual sensor networks. In: **2010 IEEE International Conference on Image Processing**. [S.l.: s.n.], 2010. ISBN 978-1-4244-7992-4.

KOREN, G.; SHASHA, D. Skip-Over: algorithms and complexity for overloaded systems that allow skips. **Proceedings 16th IEEE Real-Time Systems Symposium**, 1995. ISSN 1052-8725.

LAZARESCU, M. T. Design of a wsn platform for long-term environmental monitoring for iot applications. **IEEE Journal on emerging and selected topics in circuits and systems**, IEEE, v. 3, n. 1, p. 45–54, 2013.

MACKENZIE, A. B.; DASILVA, L. A. Game Theory for Wireless Engineers. **Synthesis Lectures on Communications**, 2006. ISSN 1932-1244.

MARCONII, M.; LAKATOS, E. **Fundamentos de metodologia científica**. [S.l.: s.n.], 2003. ISSN 9788522457588. ISBN 8522433976.

MARTINEZ, G.; LI, S.; ZHOU, C. Multi-commodity online maximum lifetime utility routing for energy-harvesting wireless sensor networks. In: **IEEE. Global Communications Conference (GLOBECOM), 2014 IEEE**. [S.l.], 2014. p. 106–111.

MELLO, C. H. P. et al. Pesquisa-ação na engenharia de produção: proposta de estruturação para sua condução. **Production**, 2012. ISSN 0103-6513.

PROVDANOV, C. C.; FREITAS, E. C. D. METODOLOGIA DO TRABALHO CINÉTICO: Métodos e Técnicas de Pesquisa e do Trabalho Acadêmico. In: **Universidade FREEVALE**. [S.l.: s.n.], 2013. ISBN 9788577171583.

RAMANATHAN, P. Overload management in real-time control applications using (m, k)-firm guarantee. **IEEE Transactions on Parallel and Distributed Systems**, 1999. ISSN 10459219.

SCHOUERY, R. C. S. et al. **Topicos da teoria dos jogos em computacao**. [S.l.]: Instituto de Matematica Pura e Aplicada, 2015. ISBN 9788524404122.

SEMPREBOM, T. et al. Skip Game: An autonomic approach for QoS and energy management in IEEE 802.15.4 WSN. In: **Proceedings - IEEE Symposium on Computers and Communications**. [S.l.: s.n.], 2015. ISBN 978-1-4673-7194-0. ISSN 15301346.

SEMPREBOM, T. et al. Dynamic GTS scheduling of periodic skippable slots in IEEE 802.15.4 Wireless Sensor Networks. In: **IFAC Proceedings Volumes (IFAC-PapersOnline)**. [S.l.: s.n.], 2009. ISSN 14746670.

SEMPREBOM, T. et al. Energy consumption and spatial diversity trade-off in autonomic Wireless Sensor Networks: The (m,k)-Gur Game approach. In: **IEEE International Conference on Industrial Informatics (INDIN)**. [S.l.: s.n.], 2013. ISBN 9781479907526. ISSN 19354576.

SEVERINO, A. J. Teoria e prática científica. In: **Metodologia do trabalho científico**. [S.l.: s.n.], 2007. ISBN 9788524913112 8524913118.

SHARMA, S.; BANSAL, R. K.; BANSAL, S. Issues and Challenges in Wireless Sensor Networks. **2013 International Conference on Machine Intelligence and Research Advancement**, n. October,

p. 58–62, 2013. Disponível em:

<<http://ieeexplore.ieee.org/document/6918795/>>.

SHI, H.-Y. et al. Game Theory for Wireless Sensor Networks: A Survey. **Sensors**, v. 12, n. 12, p. 9055–9097, 2012. ISSN 1424-8220. Disponível em: <<http://www.mdpi.com/1424-8220/12/7/9055/>>.

TSUO, F.-Y. et al. Energy-Aware Transmission Control for Wireless Sensor Networks Powered by Ambient Energy Harvesting: A Game-Theoretic Approach. In: **2011 IEEE International Conference on Communications (ICC)**. [S.l.: s.n.], 2011. ISBN 978-1-61284-232-5. ISSN 978-1-61284-232-5.

TUNG, B.; KLEINROCK, L. Using finite state automata to produce self-optimization and self-control. **IEEE Transactions on Parallel and Distributed Systems**, 1996. ISSN 10459219.

WERNER-ALLEN, G. et al. Fidelity and Yield in a Volcano Monitoring Sensor Network. **OsdI . Usenix, 2006.**, p. 381–396, 2006.

YETGIN, H. et al. A survey of network lifetime maximization techniques in wireless sensor networks. **IEEE Communications Surveys & Tutorials**, IEEE, v. 19, n. 2, p. 828–854, 2017.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. **Computer Networks**, 2008. ISSN 13891286.

ZINONOS, Z.; CHRYSOSTOMOU, C.; VASSILIOU, V. Wireless sensor networks mobility management using fuzzy logic. **Ad Hoc Networks**, Elsevier B.V., v. 16, n. 2014, p. 70–87, 2014. ISSN 15708705. Disponível em: <<http://dx.doi.org/10.1016/j.adhoc.2013.12.003>>.