

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE ENERGIA

JOELTON DEONEI GOTZ

**AVALIAÇÃO DO CONSUMO ENERGÉTICO DO  
BLUETOOTH *MESH***

DISSERTAÇÃO

CURITIBA

2020

JOELTON DEONEI GOTZ

**AVALIAÇÃO DO CONSUMO ENERGÉTICO DO  
BLUETOOTH *MESH***

Dissertação apresentada ao Programa de Pós-graduação em Sistemas de Energia da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Mestre em Engenharia Elétrica” – Área de Concentração: Automação e Sistemas de Energia.

Orientador: Prof. Dr. Guilherme Luiz Moritz

CURITIBA  
2020

#### Dados Internacionais de Catalogação na Publicação

Gotz, Joelton Deonei

Avaliação do consumo energético do *Bluetooth Mesh* [recurso eletrônico]  
/ Joelton Deonei Gotz. -- 2020.

1 arquivo eletrônico (54 f.): PDF; 843 KB.

Modo de acesso: World Wide Web.

Texto em português com resumo em inglês.

Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná.  
Programa de Pós-graduação em Sistemas de Energia. Área de  
Concentração: Automação e Sistemas de Energia. Linha de pesquisa:  
Sistemas de Automação e Instrumentação, Curitiba, 2020.

Bibliografia: f. 50-54.

1. Sistemas de energia elétrica - Dissertações. 2. Tecnologia bluetooth.  
3. Redes de sensores sem fio. 4. Energia - Consumo. 5. Redes de  
computadores - Escalabilidade. 6. Análise por agrupamento. 7. Algoritmos  
computacionais. 8. Sistemas de transmissão de dados. 9. Sistemas de  
comunicação sem fio. 10. Processamento de sinais - Técnicas digitais. 11.  
Simulação (Computadores). I. Moritz, Guilherme Luiz, orient. II. Universidade  
Tecnológica Federal do Paraná. Programa de Pós-graduação em Sistemas  
de Energia. III. Título.

CDD: Ed. 23 -- 621.31

**Biblioteca Central do Câmpus Curitiba - UTFPR**  
**Bibliotecária: Luiza Aquemi Matsumoto CRB-9/794**

## TERMO DE APROVAÇÃO DE DISSERTAÇÃO

A Dissertação de Mestrado intitulada “**Avaliação do Consumo Energético do Bluetooth Mesh**”, defendida em sessão pública pelo candidato(a) **Joelton Deonei Gotz**, no dia **08 de Abril de 2020**, foi julgada para a obtenção do Título de Mestre em Engenharia Elétrica, Área de concentração , Linha de pesquisa **Sistemas de Automação e Instrumentação**, e aprovada em sua forma final, pelo Programa de Pós-Graduação em Sistemas de Energia - PPGSE.

BANCA EXAMINADORA:

Prof. Dr. Guilherme Luiz Moritz – Presidente – UTFPR

Prof. Dr. Marcos Eduardo Pívaro Monteiro – UTFPR

Prof. Dr. André Augusto Mariano – UFPR

A via original deste documento encontra-se arquivada na Secretaria do Programa, contendo a assinatura da Coordenação após a entrega da versão corrigida do trabalho.

Curitiba, 08 de Abril de 2020.

---

Carimbo e Assinatura do(a) Coordenador(a) do Programa

## AGRADECIMENTOS

Primeiro de tudo, gostaria de agradecer a Deus por me guiar, me conceder sabedoria e me dar força para seguir em frente e superar as dificuldades. A esta Universidade, seu corpo docente, direção e administração pela oportunidade de me tornar um mestre em Ciências, cujo apoio estiveram presentes em todos os momentos.

Agradeço ao meu orientador, Professor Doutor Guilherme Luiz Moritz, pela orientação prestada, pelo seu incentivo, disponibilidade e apoio que sempre demonstrou. Aqui lhe exprimo a minha total gratidão.

Aos colegas de trabalho na empresa Ubivis por apoiarem o desenvolvimento desse estudo e por compreenderem minha ausência em horário comercial com frequente ajuda no dia a dia do trabalho.

Aos meus pais Artemio e Dirlei, pelo exemplo de dignidade e perseverança, pela educação e por terem me incentivado em seguir meus sonhos sempre. A todos os meus familiares que de alguma forma me incentivaram e apoiaram nesta etapa.

A minha esposa, Querulin Malena Perret Gotz, meu porto seguro, companheira de todas as horas, por ter caminhado ao meu lado, pela paciência, compreensão e ajuda prestada durante a elaboração da presente dissertação, especialmente por sempre estar com um sorriso e por todo seu amor.

A todos o meu sincero e profundo Muito Obrigado!

## RESUMO

GOTZ, Joelton Deonei. Avaliação do consumo energético do Bluetooth *Mesh*. 54 f. Dissertação – Programa de Pós-graduação em Sistemas de Energia, Universidade Tecnológica Federal do Paraná. Curitiba, 2020.

Esse trabalho propõe o método de clusterização baseado em LEACH para melhorar a eficiência energética de redes com Bluetooth *Mesh*. E para reduzir ainda mais a carga sofrida sobre os nós *Cluster Heads*, um algoritmo de *Radio Duty Cycling* que requer nenhuma modificação no protocolo padrão Bluetooth *Mesh* foi utilizado, com somente uma simples modificação na lógica da transmissão de pacotes.

Simulações em computadores mostram que o *Radio Duty Cycling* proposto e o método de clusterização são muito efetivos na redução do tempo ocioso de escuta do canal e no aumento da eficiência energética.

É observado que o *Radio Duty Cycling* incrementa em até 90% a melhoria em eficiência energética. Além do mais, as simulações mostram que a técnica proposta de Clusterização é efetiva no controle da replicação excessiva de mensagens, as quais são inerentes da operação do *Flooding*, que por sua vez tem um impacto positivo na taxa de entrega de pacotes e na escalabilidade da rede.

Finalmente, pode ser observado que o método proposto de Clusterização, quando aplicado com a técnica de *Radio Duty Cycling* pode fornecer uma melhoria na eficiência energética quando comparada com o protocolo Bluetooth *Mesh* original.

**Palavras-chave:** Bluetooth *Mesh*, Rede *Mesh*, Rede de Sensores sem fio.

## ABSTRACT

GOTZ, Joelton Deonei. Bluetooth mesh energy consumption assessment. 54 f. Dissertação – Programa de Pós-graduação em Sistemas de Energia, Universidade Tecnológica Federal do Paraná. Curitiba, 2020.

This work proposes a Clustering Algorithm based on LEACH to improve the energy efficiency of Bluetooth Mesh networks. To further reduce the burden over the Cluster Heads, an Radio Duty Cycling algorithm that requires no modification in the current Bluetooth Mesh Profile is needed, only a simple modification on the packet transmission logic.

Computer simulations shows that the proposed radio duty cycling and clustering methods for reducing idle listening are very effective in improving energy efficiency.

It was observed that duty cycling brings up to 90% of improvement in energy efficiency. In addition, simulations shows that the proposed clustering technique is effective in controlling the excessive message replication that is inherent in flooding operation, which in turn have a positive impact on packet delivery ratio and network scalability.

Finally, it can be observed that the proposed clustering algorithm, when applied with the proposed radio duty cycling algorithm can provide an improvement in energy efficiency when compared to the baseline Bluetooth Mesh profile.

**Keywords:** Bluetooth Mesh, Bluetooth, wireless network.

## LISTA DE FIGURAS

Figura 1	–	Pilha da arquitetura do Bluetooth	17
Figura 2	–	Comunicação entre mestre e escravos através do método de TDMA	18
Figura 3	–	Esquema de troca de mensagens entre LPN e <i>Friend</i>	21
Figura 4	–	Fluxograma de execução de um pacote recebido	22
Figura 5	–	Esquema de estabelecimento de uma <i>Friendship</i>	24
Figura 6	–	Esquema do funcionamento do protocolo <i>Radio Duty Cycling</i>	29
Figura 7	–	Ilustração do funcionamento do método de <i>Radio Duty Cycling</i>	32
Figura 8	–	Ilustração do funcionamento do processo de clusterização	33
Figura 9	–	Taxa de entrega de pacotes x número de nós para diferentes TTL's	37
Figura 10	–	Taxa de entrega de pacotes x intervalo de geração de pacotes	38
Figura 11	–	Eficiência energética x número de nós para diferentes TTL's	39
Figura 12	–	Porcentagem do consumo energético do rádio RX/TX	40
Figura 13	–	Taxa de entrega de pacotes x número de nós para diferentes <i>Duty-Cycles</i>	41
Figura 14	–	Eficiência energética x número de nós para diferentes <i>Duty-Cycles</i>	42
Figura 15	–	Taxa de entrega de pacotes x número de nós para uma rede clusterizada	43
Figura 16	–	Eficiência energética x número de nós de uma rede clusterizada	44

## LISTA DE TABELAS

Tabela 1	–	Parâmetros do Sistema. ....	36
Tabela 2	–	Ganhos do processo. ....	47

## LISTA DE SIGLAS

ACK	Acknowledgement
APP	Application Layer
ATT	Attribute Protocol
BLE	Bluetooth Low Energy
C-LEACH	Cell LEACH
CHs	Cluster Heads
D2D	Dispositivo à dispositivo
DST	Destino
GAP	Generic Access Profile
GATT	Generic Attribute Protocol
GMSK	Gaussian Minimal Shift Keyin
IDC	Internet Data Corporation
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IETF	Internet Engineering Task Force
IOT	Internet of Things
ISM	Industrial Sientific and Medical
L2CAP	Logical Link Control Adaptation Protocol
LEACH	Low Energy adaptive clustering hierarchy
LL	Link Layer
LMR	Light-weight mobile routing
MD	More Data
PHY	Physical Layer
RF	Relay-Friend
RFCH	Relay-Friend Cluster Head
RPL	Routing Protocol for Low Power and Lossy Networks
RX	Receptor
SIG	Special Interest Group
SMP	Security Manager
SoC	System on a Chip
TDMA	Tempo de Divisão múltiplo de acesso
TSCH	IEEE Time Switching Channel Hopping
TTL	Time to Live
TX	Transmissor

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	MOTIVAÇÃO	10
1.2	OBJETIVOS	14
1.2.1	Objetivos Gerais	14
1.2.2	Objetivos Específicos	14
1.3	ESTRUTURA DO TRABALHO	15
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>16</b>
2.1	BLUETOOTH <i>LOW ENERGY</i>	16
2.1.1	Bluetooth <i>Mesh</i>	19
2.1.2	Desempenho do Bluetooth <i>Mesh</i>	24
2.2	INCREMENTANDO DESEMPENHO DO <i>FLOODING</i>	26
2.2.1	Algoritmo de Clusterização	26
2.2.2	Algoritmo de <i>Radio Duty Cycling</i>	28
<b>3</b>	<b>ALGORITMO PROPOSTO</b>	<b>31</b>
<b>4</b>	<b>RESULTADOS</b>	<b>35</b>
4.1	AMBIENTE DE SIMULAÇÃO	35
4.2	RESULTADO DAS SIMULAÇÕES	36
<b>5</b>	<b>COMENTÁRIOS FINAIS</b>	<b>46</b>
5.1	CONCLUSÃO	46
5.2	TRABALHOS FUTUROS	47
	<b>REFERÊNCIAS</b>	<b>50</b>

# 1 INTRODUÇÃO

## 1.1 MOTIVAÇÃO

Há um consenso de que o futuro da internet é se expandir para que todas as coisas, lugares e objetos possam se beneficiar da conectividade. Por esta razão, tem-se observado um aumento de documentos e aplicações no campo da Internet das Coisas (IOT). Institutos de respeito com a Forbes (COLUMBUS, 2018), IDC (IDC CORPORATE USA, 2019) e Ericsson (ERICSSON, 2019), fazem previsões de um crescimento do mercado e uma explosão da quantidade de dispositivos conectados. Isso leva a expectativa de que o número de dispositivos IOT superará a barreira de trilhões de dólares nos gastos com tecnologia em todo o mundo e o montante de 28 bilhões de dólares no número de dispositivos conectados até 2022.

Por outro lado, ainda não se observa um mundo totalmente conectado com todas as coisas. Há um grande número de aplicações de IOT, como sistemas de automação predial, residencial, *smart grid*, *smartphones* e *smartwatches* que monitoram diferentes parâmetros dos usuários e até mesmo plantas industriais conectadas à internet. No entanto, uma diferença qualitativa no estilo de vida da população só será notada quando a densidade de sensores instalados atingir níveis que são o dobro ou o triplo do que é observado no cenário atual (STANKOVIC, 2014). Portanto, para chegar a esse número de dispositivos há muito trabalho para ser feito. Os dispositivos IOT são muito heterogêneos, variando desde pequenos sensores até grandes dispositivos como carros. Além disso, para atender esses requisitos, cada dispositivo possui diferentes necessidades de comunicação, de cobertura, de consumo de energia, de manutenção e escalabilidade.

Para atender aos diferentes requisitos mencionados, diversos protocolos são propostos, cada um indicado para um nicho específico de aplicações. Por exemplo, um protocolo bem aceito e empregado na indústria é o *IEEE Time Switching Channel Hopping* (TSCH) (IEEE STD 802.15.4E, 2012; VILAJOSANA et al., 2014). Em uma rede TSCH, os nós são sincronizados e o tempo dividido em *slots*. Sendo assim, períodos de transmissão e recepção são agendados para serem repetidos ao longo do tempo. Já para transmissões em longas distâncias onde exigem uma alta latência e uma baixa taxa de transmissão de dados como visto em *Smart Cities* e *Smart Metering*, tecnologias como LoRaWAN (SORNIN; YEGIN, 2017) e SigFox (SIGFOX, 2019) são empregadas. Esses protocolos são desenvolvidos para trabalharem com baixa disponibilidade de energia e

são distribuídos em uma larga área de cobertura. Redes *Low Power Wide Area Network* (LPWAN) foram desenvolvidas para transmitir baixíssimas taxas, porém por vários quilômetros. Por outro lado, para ambientes domésticos como aplicações de automação residencial e predial, nicho no qual esse trabalho tem foco, outras tecnologias se destacam como o Thread (THREAD, 2019), Zigbee (ZIGBEE ALLIANCE, 2007), Wi-fi (IEEE STD 802.11AH, 2017) e Bluetooth (BLUETOOTH SIG, 2018).

Dentre os pares citados, Thread e Zigbee são baseadas na norma IEEE 802.15.4 vista em (IEEE STD 802.15.4, 2006), o qual especifica transmissão em baixa potência e permite que sua utilização pode ser estendida para redes *Mesh*. Já outros protocolos como o Wi-Fi e o Bluetooth possuem desvantagens sobre 802.15.4, já que o Wi-Fi foi desenvolvido com foco na transmissão de altas taxas de dados para ser utilizado para nós com menor preocupação com o consumo energético.

O Bluetooth seguindo a mesma linha, originalmente foi desenvolvido para substituir os cabos periféricos por uma conexão sem fio de curta distância. Porém, não apenas o Bluetooth, mas também o Wi-Fi possuem duas grandes vantagens em relação aos demais. Primeiro, é a facilidade de encontrar os dispositivos Wi-Fi e Bluetooth no cotidiando das pessoas e também não necessitam de um *Gateway* especial para gerenciar a rede como acontecem com os protocolos 802.15.4.

Um protocolo mais adequado para sistemas IOT utilizando Wi-Fi surgiu em 2017, denominado 802.11ah (IEEE STD 802.11AH, 2017), o qual permite que o Wi-Fi trabalhe com frequência abaixo de 1 GHz, o que incrementa o alcance e diminui o consumo. Por outro lado, até o momento, o protocolo não foi bem aceito na indústria, já que os atuais hardwares que trabalham com frequências de 2.4GHz e 5GHz não incluem a faixa dos 900MHz prevista no 802.11ah. Nesse contexto, para ingressar no mercado IOT, o Bluetooth *Special Interest Group* (SIG) lançou a versão *Mesh* (BLUETOOTH SIG, 2017) em 2017 com foco em melhorias em relação às versões anteriores na área de cobertura do protocolo. Esse padrão especifica uma versão *Flooding Mesh* sobre os canais de *Advertisement* do Bluetooth, no qual não necessita nenhuma modificação na pilha de comunicação do Bluetooth, o que torna o padrão Bluetooth *Mesh* compatível com qualquer versão superior ao Bluetooth 4.0, que foi a primeira versão a suportar a camada *Low Energy*. O *Flooding* é empregado no Bluetooth *Mesh* para manter simplicidade de implantação e permitir a compatibilidade com versões anteriores, porém pode ocasionar problemas de transmissões (ZHANG; JIANG, 2004), baixa escalabilidade e baixa eficiência energética (SILICON LABS INC, 2018).

Um trabalho de comparação com o padrão Bluetooth *Mesh* é visto em (SILICON LABS INC, 2018). Esta comparação indica que o padrão Bluetooth *Mesh* possui menor taxa de transferência de pacotes e ainda possui uma latência maior se comparado com o padrão 802.15.4 mesmo tendo uma taxa de transferência em camada física de 2 Mbps, enquanto o 802.15.4 opera em 250 kbit/s. Outro ponto observado foi a degradação da taxa de entrega de pacotes com o aumento da rede principalmente para pacotes maiores que 8 bytes que é a carga máxima do pacote Bluetooth *Mesh* sem que ocorra fragmentação (BLUETOOTH SIG, 2017).

Antes do lançamento oficial do Bluetooth *Mesh*, vários trabalhos já exploravam redes com BLE *Mesh* usando ideias utilizadas com sucesso em outros tipos de redes de sensores, as quais podem ser usadas para melhorar o protocolo oficial Bluetooth *Mesh*. Uma extensa revisão da proposta está disponível em (DARROUDI; GOMEZ, 2017), e na sequência as principais propostas serão discutidas.

Muitos trabalhos tentam limitar a repetição de mensagens geradas pelo *Flooding* com o objetivo de aumentar a eficiência energética e melhorar a taxa de entrega de pacotes. Como por exemplo (KYASANUR et al., 2006), que propôs uma rede *Mesh* baseada no algoritmo Gossip. Nesse trabalho, cada nó possui uma certa probabilidade de retransmissão de um pacote já retransmitido, oposto ao *Flooding* original, no qual cada mensagem recebida é retransmitida pelo menos uma vez. Usando uma estratégia de limitação de transmissões, (GOGIC et al., 2016) propôs uma rede Bluetooth *Mesh* com *Flooding* onde os nós monitoram o canal afim de checar por mensagens repetidas. Assim, uma mensagem é somente retransmitida se ela não for observada por uma janela de tempo  $c$  cujo tamanho é calculado pelo algoritmo de Trickle (LEVIS et al., 2004). De acordo com (GOGIC et al., 2016), a proposta incrementou a escalabilidade e melhorou a taxa de entrega comparada com o *Flooding* convencional.

Diferente da solução original do Bluetooth *Mesh* ou de trabalhos mencionados até o momento, uma outra versão *open source* chamada *FruityMesh* (M-WAY SOLUTIONS GMBH, 2018) é apresentada por (GOGIC et al., 2016) e comunica via canais de dados. Nela, uma comunicação por *Scatternet*, a qual é uma comunicação de longo alcance que utiliza nós intermediários como ponte é formada. Essa abordagem indica que pacotes de *Advertisement* com informações dos grupos de *Scatternet* são periodicamente enviados e utilizados pelos nós para encontrar disponibilidade de conexão. O roteamento básico é inundar cada conexão com mensagens recebidas, porém, essa inundação não interfere na entrega de pacotes, uma vez que as transmissões Bluetooth orientadas à conexão usam

transmissões ortogonais no tempo de canal para retransmissão.

Como motivação para escolher a melhor maneira de criar uma rede Bluetooth *Mesh*, seja ao usar canais de *Advertisement* ou canais de dados, o autor (MURILLO et al., 2017) comparou na prática o trabalho de Trickle citado por (LEVIS et al., 2004) e *FruityMesh* (M-WAY SOLUTIONS GMBH, 2018) usando rádios reais. Nesse cenário, uma rede composta por 9 nós foi testada com o incremento na taxa de geração de pacotes. O estudo mostra que a taxa de entrega de pacotes cai relativamente igual para ambas as implementações conforme aumenta a taxa de geração de pacotes. No entanto, a latência incrementa de forma significativa para o método de *FruityMesh*, enquanto não é afetada no Trickle. Por outro lado, o método de *FruityMesh* gasta em torno de 3 vezes menos energia para gerar a mesma quantidade de pacotes. Com base nas análises descritas, esse trabalho não apresenta nenhuma conclusão significativa sobre qual método possui melhor desempenho.

Outros trabalhos propuseram usar algoritmos de roteamento para trabalhar com o Bluetooth *Mesh*, como por exemplo (SIRUR et al., 2015), que propôs usar uma rede *Mesh* utilizando o protocolo RPL (ALEXANDER et al., 2012). O método foi implementado utilizando celulares, os quais executavam sistema operacional Android. Por essa razão, o consumo de energia e a latência foram significativamente influenciados pela escolha do hardware e dessa forma não podem ser comparados com outras técnicas. Um trabalho similar foi proposto por (LEE et al., 2016), o qual implementou a solução utilizando uma placa Raspberry Pi e comparando a solução com 802.15.4 (IEEE STD 802.15.4, 2006). Os resultados mostraram que o Bluetooth *Mesh* sobre RPL pode melhorar o 802.15.4 na entrega de pacotes e no *Duty-Cycle*.

Outra técnica que pode melhorar o desempenho em redes *Flooding* é o método de clusterização. Uma rede clusterizada é dividida em várias células, onde um nó assume um papel especial de coordenar as transmissões e retransmissões de dados acumulados para o destino. Essa arquitetura pode melhorar a eficiência energética e a escalabilidade o qual reduz o número de troca de mensagens entre sensores (SINGH et al., 2017). Um dos primeiros e mais conhecidos casos do uso do método é o *Low Energy adaptive clustering hierarchy* (LEACH) (HEINZELMAN et al., 2000). O LEACH tem como objetivo reduzir o consumo energético da rede ao selecionar uma quantidade de nós, os quais são chamados de *Cluster Heads* (CHs) e atribuir a eles uma função especial de armazenar e retransmitir mensagens, enquanto os demais nós se associam a eles.

De acordo com a norma (BLUETOOTH SIG, 2017), devido ao *Flooding*, é

recomendado que os nós permaneçam 100% do tempo com o rádio receptor ligado para evitarem que mensagens sejam perdidas. Essa prática, eleva o consumo energético e resulta no fim da vida útil da bateria, prevendo que não é possível substituí-la. Por isso faz-se necessário utilizar métodos de redução do tempo de rádio ligado. Para reduzir o tempo de rádio ligado, um dos algoritmos mais conhecidos é ContikiMAC (DUNKELS, 2011) que utiliza a técnica de *Duty-Cycle*. Essa técnica possibilita que o nó durma durante certo período de tempo em que não deveria receber pacote. Esse algoritmo reduz significamente o consumo energético dos nós.

Portanto, este trabalho se propõe a utilizar as técnicas de LEACH aliadas as métricas de *Duty-Cycle* no protocolo oficial Bluetooth *Mesh*, para resultar em redução do consumo energético, diminuição do número de colisões e aumento da entrega de pacotes. Este trabalho também resultou em um artigo publicado que aguarda revisão, mas potencialmente irá gerar a seguinte publicação (GOTZ et al., 2020).

## 1.2 OBJETIVOS

### 1.2.1 Objetivos Gerais

Avaliar o desempenho do BLE *Mesh* tanto em termos de escalabilidade e de eficiência energética além de propor métodos para otimizar o desempenho obtido.

### 1.2.2 Objetivos Específicos

- Implementar um simulador do BLE *Mesh* baseado nas regras estabelecidas e encontradas na norma (BLUETOOTH SIG, 2017).
- Verificar o sucesso da taxa de entrega de pacotes utilizando o simulador desenvolvido.
- Verificar a escalabilidade do protocolo BLE *Mesh*.
- Verificar a eficiência energética do protocolo.
- Propor técnicas que diminuam os problemas de escalabilidade e baixa eficiência energética no BLE *Mesh*.

### 1.3 ESTRUTURA DO TRABALHO

Após a introdução vista no Capítulo 1, o Capítulo 2 deste trabalho apresenta um estudo do conceito e funcionamento do Bluetooth, de sua história de evolução até chegar ao BLE *Mesh*. Também será realizado um estudo em temas que podem ser empregados nesse trabalho como o *Duty-Cycle* e o protocolo LEACH. O Capítulo 3 apresenta a descrição do desenvolvimento do algoritmo proposto para realizar a simulação e testes necessários quanto à escalabilidade do BLE *Mesh* e o seu consumo energético.

No Capítulo 4 serão apresentados os resultados obtidos com a implementação e testes realizados com o BLE *Mesh*. No Capítulo 5 serão apresentados os trabalhos futuros que podem ser desenvolvidos nessa área e por último a conclusão deste trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 BLUETOOTH *LOW ENERGY*

O protocolo Bluetooth (BLUETOOTH SIG, 2018) foi proposto em 1998 com o objetivo de substituir as conexões cabeadas entre computadores e dispositivos por uma conexão sem fio. Para a primeira versão, foi implementado um *piconet* composto por um mestre com até 7 escravos conectados a ele. Nos primeiros anos, o foco do protocolo era ganhar em cobertura e velocidade de transmissão.

Apesar de avanços a cada lançamento de versão, foi em 2010 que chegou ao mercado uma versão não apenas focada em taxa e cobertura, mas também em economia energética. Nessa versão, houve a adição de uma camada chamada *Low Energy Stack* (*BLE Stack*), a qual foca em economia de energia.

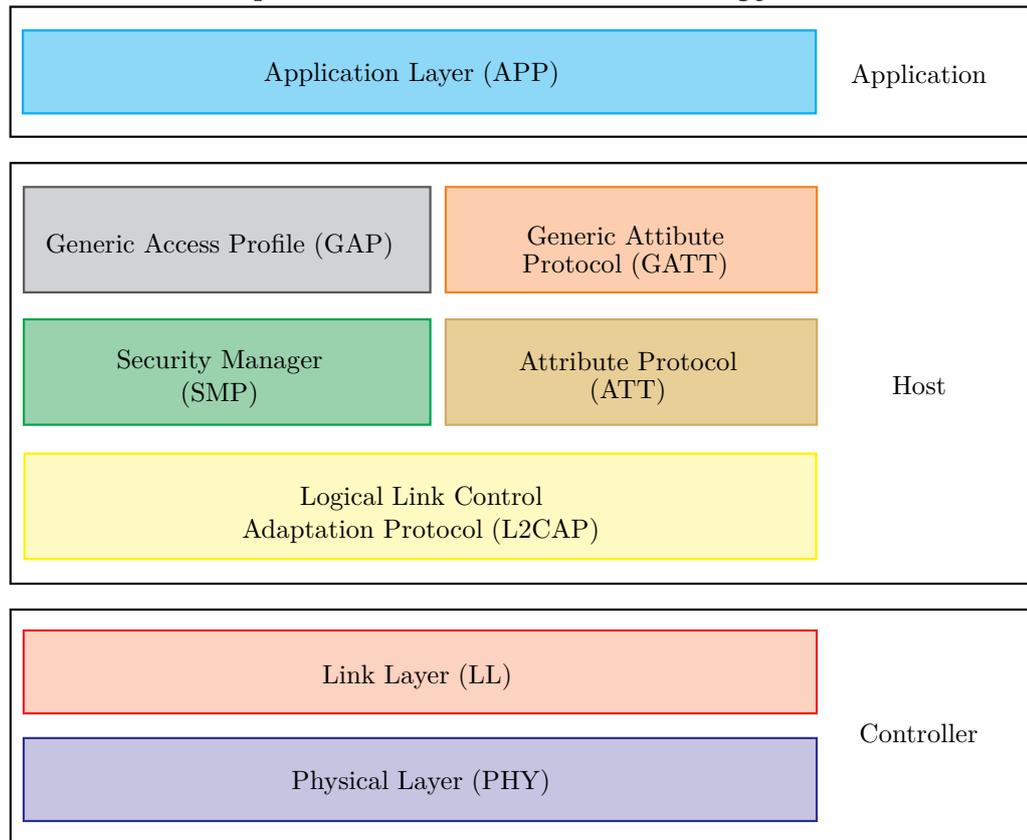
É importante notar que ambas as especificações seguem a mesma filosofia, mas não são interoperáveis. A versão *Low Energy* foi concebida desde a camada física até a camada de aplicação com foco na comunicação D2D (baixo consumo, baixa latência, baixa taxa), enquanto a versão clássica continua com foco na melhoria da taxa de transmissão de dados. Além disso, um dispositivo Bluetooth pode implementar ambas as especificações de forma simultânea. O restante desse trabalho irá focar na especificação Bluetooth *Low Energy* utilizada nas aplicações do Bluetooth *Mesh*.

A Figura 1 mostra as várias camadas da pilha do BLE que será explicada na sequência. A camada física divide a faixa ISM de 2.4 GHz em 40 canais, com espaçamento de 2 MHz entre eles. A modulação GMSK (do inglês *gaussian minimal shift keying*) é usada para transmitir até 1 Mbits/s. A versão Bluetooth 5.0 terá como taxa opcional de transmissão de até 2 Mbits/s.

A camada de roteamento (*Link Layer*) utiliza 40 canais definidos e divididos em 37 de comunicação orientada e 3 canais de *Advertisement*. Os 37 canais de comunicação orientada são utilizados para realizar a troca de pacotes de forma bidirecional entre os nós conectados. Já os 3 canais restantes, possuem a função de descobrir dispositivos, estabelecer conexões e transmitir *broadcast*. Na versão Bluetooth 5.0, os canais de comunicação orientada poderão ser utilizados para o processo de *Advertisement* que será explicado no próximo parágrafo.

Um dispositivo BLE desconectado usa pacotes *Advertisement* para transmitir

**Figura 1 – Pilha da arquitetura do Bluetooth *Low Energy*.**



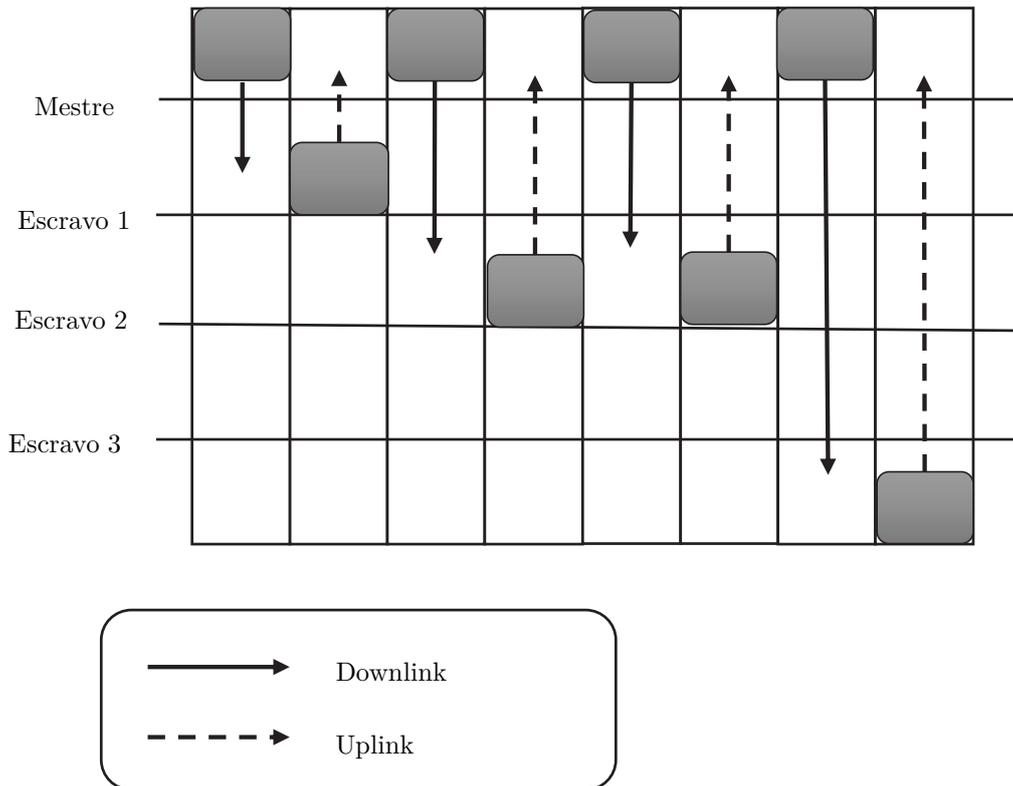
**Fonte:** Adaptado de (MICROCHIP TECHNOLOGY INC., 2019)

informações de emparelhamento e presença para potenciais dispositivos usuários próximos. Cada transmissão é realizada de maneira sucessiva em cada um dos três canais de *Advertisement*, sem o uso de sincronização. Dispositivos receptores próximos que receberem os *broadcasts* podem usar as informações recebidas para iniciar uma conexão com um dispositivo emissor. Após a conexão, o dispositivo emissor entra no *piconet* mestre/escravo o qual é gerenciado pelo dispositivo emissor.

O acesso ao meio dos dispositivos emparelhados é administrado pelo protocolo de tempo de divisão múltiplo de acesso ao canal (TDMA) como observado na Figura 2. Nesse protocolo, cada escravo dorme e acorda de forma pré-estabelecida em intervalos de tempo em que pode escutar e transmitir ao mestre. O mestre estará disponível para esse escravo nesse intervalo de disponibilidade de comunicação entre ambos. Os dispositivos escravos podem apenas transmitir quando endereçados pelo mestre, o qual garante canal livre para troca de mensagens. Além disso, a seleção adequada dos intervalos de conexão permitem uma operação confiável e de baixo consumo energético para cada escravo, levando em consideração os requerimentos da largura de banda de forma individual (ZHOU et al.,

2017; LAYA et al., 2016).

**Figura 2 – Esquema de comunicação entre mestre e escravos através do método de TDMA, onde cada escravo possui um período de tempo de comunicação exclusiva com o mestre, no qual não haverá comunicação entre o mestre e qualquer outro escravo.**



**Fonte: Autoria própria**

No topo da camada de acesso ao meio, a fragmentação e multiplexação é mantida por um protocolo de controle de roteamento e adaptação (L2CAP), o qual providencia de ponta a ponta lógicas de comunicação de dados. Serviços opcionais relacionados à criptografia, como emparelhamento, distribuição de chaves de segurança de dados, são implementados sobre o L2CAP pela camada de gerência de segurança.

Finalmente, dois perfis são apresentados aos usuários da pilha BLE. O primeiro, denominado *Generic Access Profile* (GAP), que é usado para controlar operações BLE, como descobrir e conectar-se com os nós vizinhos, transmitir dados e estabelecer conexões seguras. E o segundo denominado *Generic Attribute Profile* (GATT), estabelece a maneira como os dados serão organizados e transmitidos através de uma conexão BLE. Este protocolo usa o ATT (*Attribute Protocol*) como um mecanismo de transporte.

Um ATT é um protocolo, o qual trabalha em modo *client/server*, baseado em atributos que estão no servidor e podem ser acessados pelo cliente. O GATT agrupa um conjunto de ATT's relacionados em características que são posteriormente organizados como um serviço. Serviço é um conjunto de características e relacionamentos com outros serviços que encapsulam o comportamento de parte de um dispositivo. O comportamento do dispositivo é definido por um perfil que contém um conjunto de serviços. Vários perfis são padronizados pelo Bluetooth SIG como, por exemplo, o serviço de posicionamento interno ou a especificação do protocolo BLE *Mesh*, o qual será detalhado na Seção 2.1.1.

### 2.1.1 Bluetooth *Mesh*

O perfil *Mesh* (BLUETOOTH SIG, 2017) é uma arquitetura em camadas lançada em 2017 para permitir a comunicação de múltiplos saltos utilizando a pilha BLE. A primeira camada da especificação (do inglês *Bearer Layer*) define como as mensagens são transportadas entre os nós usando um perfil BLE subjacente. Atualmente, são definidos dois métodos de transmissão, *Advertisement Bearer* e *GATT Bearer*. O método de transmissão preferido é o *Advertisement Bearer*, que utiliza recursos de *Advertisement* e *Scanning* da camada GAP do BLE.

Um dispositivo que não suporta o *Advertisement Bearer* pode utilizar o *GATT Bearer*, o qual é um mecanismo que possibilita que um dispositivo antigo se comunique em uma rede *Mesh*. Como a comunicação GATT não suporta *broadcast*, um nó Bluetooth *Mesh* utilizando GATT apenas envia mensagens ao mestre em que está emparelhado. Este mestre então retransmite utilizando o *Advertisement Bearer*. Dessa forma, a principal camada de transmissão do Bluetooth *Mesh* é o *Advertisement Bearer*, por esse motivo, este trabalho se concentrará nele.

Um nó Bluetooth *Mesh* implementa dois papéis de comunicação GAP para formar um roteamento em *Flooding*. O primeiro possui função de observador para sempre monitorar os canais *Advertisement* com o propósito de interceptar qualquer mensagem *Mesh* recebida de nós próximos. Quando receber uma mensagem, ela será retransmitida após um *Delay* randômico usando sucessivamente os 3 canais de *Advertisement*. Isso permite que uma mensagem possa ser recebida por outros nós próximos, os quais farão retransmissão do pacote. Esse processo permite que a mensagem se propague pela rede, até eventualmente chegar ao destino, independente da distância entre a origem e o destino e da quantidade de saltos mínimos necessários para a mensagem chegar no receptor, desde que o TTL seja suficiente.

No entanto, esse processo gera um aumento do número de pacotes circulando pela rede. Mas para evitar esse congestionamento de mensagens, o protocolo trabalha com algumas lógicas que evitam *loops* repetitivos. Essa prática, impede que um pacote já retransmitido por esse nó possa ser enviado novamente (TIAN, 2014). Além disso, em todo pacote enviado é inserido um contador chamado *time to live* (TTL) que decrementa a cada retransmissão. Dessa maneira, para evitar retransmissões desnecessárias, somente pacotes com TTL acima ou igual a 1 podem ser retransmitidos.

Para otimizar o TTL inicial de cada pacote, cada nó envia frequentemente uma mensagem chamada *Heartbeat*, a qual contém o endereço do nó gerador, o número de saltos iniciais e número de saltos restantes até a chegada do pacote. Com essa informação, os nós obtêm a quantidade necessária de saltos para enviar um pacote para um determinado destino.

Desde a primeira mensagem *Heartbeat* recebida, cada nó atualiza uma tabela interna que contém o número mínimo de saltos necessário para alcançar o nó receptor a partir do gerador do *Heartbeat*. Em seguida, o nó decrementa o valor do TTL e retransmite o *Heartbeat*. O valor mínimo de saltos contidos na tabela é utilizado para calcular o TTL inicial de cada mensagem endereçada para determinado nó destinatário. Se não houver um valor mínimo contido na tabela, um valor fixo de 127 será utilizado como TTL daquele pacote.

O perfil Bluetooth *Mesh* define 4 diferentes papéis que um nó pode assumir, e inclusive de forma simultânea. O primeiro papel detalhado é o chamado *Relay Node*, o qual possui a função de sempre retransmitir mensagens recebidas, desde que não seja o destinatário. Porém, para diminuir o congestionamento gerado pelo *Flooding*, cada *Relay Node* mantém uma lista das mensagens transmitidas. Dessa forma, se uma mensagem recebida estiver nessa lista, significa que já foi retransmitida e então ela será descartada sem seguir adiante, o que reduz de forma significativa o tráfego de pacotes circulando na rede.

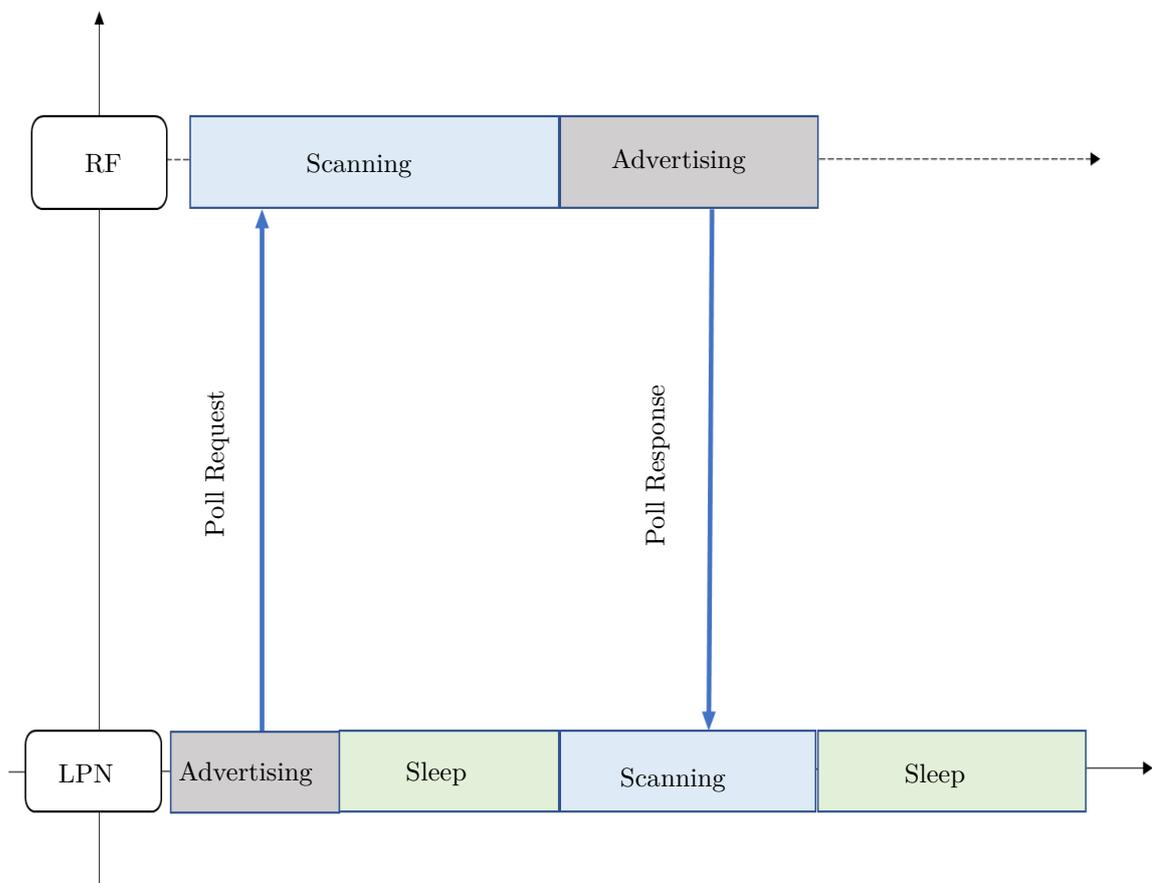
No entanto, esse nó precisa ficar 100% do tempo com o rádio receptor ligado para possibilitar que todas as mensagens sejam recebidas pelo nó, e evitar perda de pacotes. Porém tal comportamento, gera ineficiência energética. Apesar disso, é importante ressaltar que enquanto o receptor permanece 100% do tempo ligado, o rádio transmissor só é utilizado quando há necessidade de transmitir um pacote. Dessa forma, o consumo energético total do nó é denominado pela recepção de pacotes.

Para evitar o alto consumo energético gerado pelo nó *Relay-Friend*, o nó *Low*

*Power Node* (LPN) foi desenvolvido para operar em intervalos de tempo reduzidos. Para realizar economia de energia, é permitido para o nó LPN desligar o rádio receptor por certo período de tempo e acordar periodicamente para verificar se há mensagens destinadas a ele.

O período em que o nó LPN dorme pode resultar em perda de mensagens. Por isso, um nó com função chamada *Friend* foi especificado pela norma. Esse nó é responsável por armazenar mensagens destinadas a nós LPNs associados.

**Figura 3** – Ilustração da troca de pacotes *Poll Request* e *Poll Response* entre LPN e *Friend*. Um pacote *Poll Request* é enviado pelo LPN enquanto o RF estiver em modo *Scanning*. Após o envio, o LPN entra em modo *Sleep*. Depois de receber o pacote, o RF entra em modo *Advertising* e envia uma resposta *Poll Response* enquanto o nó LPN estiver disponível e em modo *Scanning*. Após esse período o nó LPN entra em modo *Sleep*.



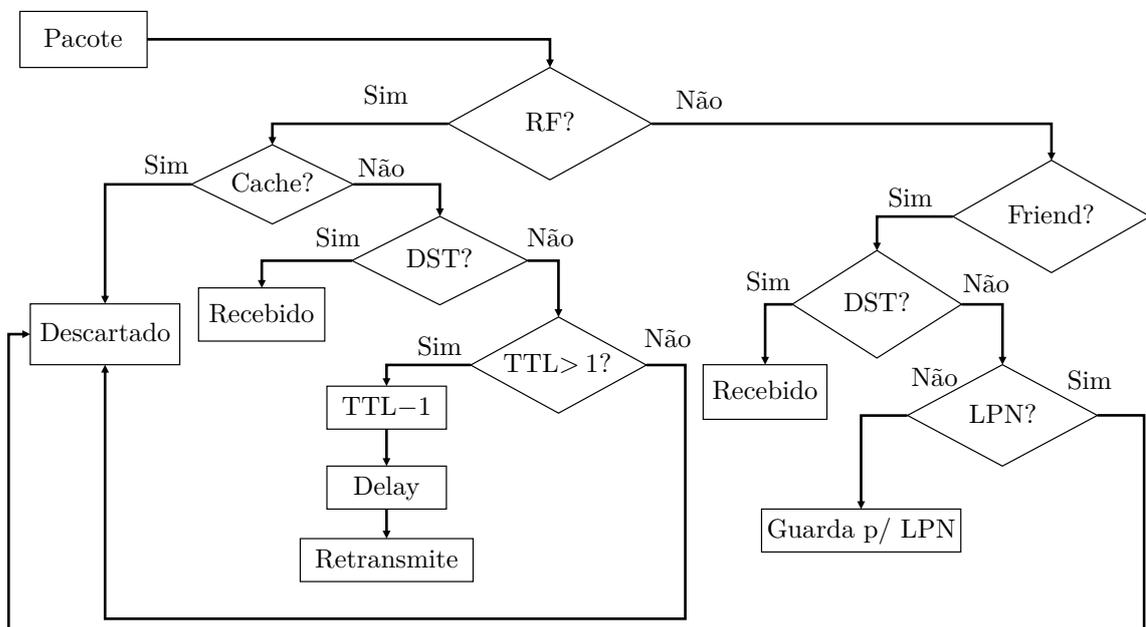
**Fonte:** Adaptado de (BLUETOOTH SIG, 2017)

Portanto, em uma rede Bluetooth *Mesh* cada LPN deve ter pelo menos um *Friend* para poder receber mensagens. O processo de troca de mensagens entre eles é pelo processo de *Poll Request/Response*. O nó LPN ao acordar envia um *Poll Request*

ao *Friend* associado. A resposta volta através de um *Poll Response* com a mensagem armazenada, caso ela exista, e o número de pacotes restantes.

O processo se repete até que todas as mensagens armazenadas sejam entregues ao nó LPN. Após a entrega de todos os pacotes, o nó LPN volta a dormir como visto na Figura 3.

**Figura 4 – Lógicas executadas por um nó Bluetooth Mesh após o recebimento de um pacote. Se o nó for um *Relay-Friend*, ele irá retransmitir o pacote caso não for um destino. Já se for um nó *Friend*, ele irá armazenar a mensagem para o LPN destino.**



Fonte: Autoria própria

A Figura 4 mostra o fluxograma da lógica de funcionamento do protocolo Bluetooth Mesh. Ao receber um pacote, se o nó desempenhar o papel de *Relay-Friend*, em um primeiro momento será verificado se a mensagem já foi retransmitida por este nó, através do histórico de mensagens no *Cache*. Caso afirmativo, essa mensagem será descartada. Não sendo descartada, o próximo passo é verificar o destino. Se o pacote for endereçado para o receptor, a mensagem chegou ao destinatário. Caso contrário, haverá uma verificação do valor do TTL. Se ele for maior que 1, o TTL do pacote será decrementado em 1 e após um *Delay* randômico será retransmitida.

Ainda de acordo com a Figura 4, se o pacote for recebido por um *Friend*, haverá a verificação do destino. Se for o próprio nó, então a mensagem foi entregue com sucesso. Caso contrário será verificado se o pacote foi endereçado a um dos seus nós LPNs. Se sim,

a mensagem será armazenada ou caso contrário será descartada.

Como já visto, um nó LPN pode apenas receber pacotes através do processo *Poll Request/Response*. Dessa maneira, um nó LPN não associado irá periodicamente enviar um *Broadcast Request* com  $TTL = 1$  a *Friends* próximos. Os nós *Friends* próximos que receberem esse *Poll* irão responder com um *Poll Offer* que contém informações como quantidade de memória disponível para armazenamento de pacotes, *Receive Window* (indica o intervalo de tempo disponível para receber um *Poll* desse LPN), o tamanho da lista de armazenamento para os nós LPNs, a intensidade de sinal e o número de *Poll Offer* que ele já enviou. Com o recebimento de vários *Poll Offer*, o nó LPN utiliza a informação desse pacote junto com a intensidade de sinal para decidir qual é o nó mais adequado para estabelecer a associação chamada *Friendship* entre os nós. O processo é ilustrado na Figura 5.

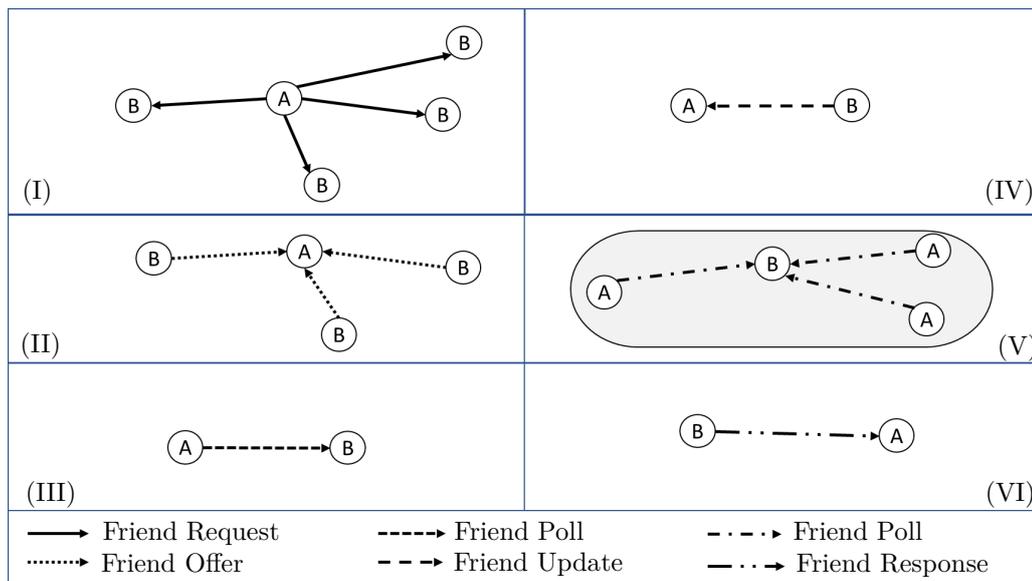
O *Poll Request* gera vários efeitos positivos na duração da carga da bateria dos nós LPNs. Por outro lado, a *Friendship* pode causar uma sobrecarga no instante de tentativa de estabelecimento da associação. Pois os nós LPNs irão continuar enviando *Poll Request* até o recebimento de *Request/Response*. Apesar disso, essas mensagens são transmitidas com  $TTL=1$ , o que não causa *loops* de mensagens na rede, desde que os nós estejam localizados em uma área de cobertura um do outro. Complementar a isso, o nó LPN não participa do processo de retransmissão de mensagens, o que evita e diminui o congestionamento da rede. Mas esse processo, pode decrementar a entrega de pacotes. Esse comportamento será investigado no Capítulo 4.

O último tipo de nó Bluetooth *Mesh* foi proposto para ser compatível com versões anteriores de Bluetooth. O nó *Mesh Proxy* utiliza o GATT e está disponível para receber pacotes de dispositivos sem suporte Bluetooth *Mesh*. Além disso, a norma apresenta a possibilidade de um nó assumir vários papéis, sendo assim, esse nó pode possuir também a função *Relay* e retransmitir mensagens por *Flooding*.

Através do esquema proposto, o Bluetooth *Special Interest Group* (SIG) apresentou a versão Bluetooth *Mesh* sem alteração da especificação do Bluetooth *Low Energy* pelas quais, as mensagens são apenas enviadas quando os dispositivos estão emparelhados. Por outro lado, as regras de roteamento *Flooding* podem gerar uma repetição descontrolada de pacotes, o que causa problemas de eficiência energética e congestionamento da rede.

Como um fator agravante, os canais de *Advertisement* usam um acesso ao meio do tipo Aloha (LAYA et al., 2016) não sincronizado. Esse fato diminui o desempenho da

Figura 5 – Para estabelecer uma associação entre um nó LPN (A) com um *Friend* (B), algumas etapas são respeitadas: (I) LPN envia um *Friend Request* para os nós *Friends* próximos. (II) Ao receber um *Friend Request*, *Friends* próximos enviam para o nó emissor um pacote *Friend Offer* com informações de intensidade de sinal e tamanho para armazenar mensagens. (III) Ao receber vários *Poll's Offer*, o LPN decide o melhor *Friend* para se associar, e então envia um *Friend Poll* mostrando a intenção da associação. (IV) Ao receber o pacote, o *Friend* envia um *Friend update* e a *Friendship* é estabelecida. (V) O nó *Friend* começa a armazenar os pacotes para o LPN. (VI) Após receber um *Poll Request*, o nó *Friend* envia pacotes recebidos ao LPN.



Fonte: Autoria própria

entrega de pacotes em grandes redes e obriga com que os nós mantenham os receptores ligados em 100% do tempo o canal enquanto não estiverem transmitindo. Esse processo gera maior consumo de energia e diminui severamente a carga da bateria dos dispositivos. Nos próximos Capítulos, serão revisados os trabalhos recentes da comunidade para implementar soluções de economia de energia que supostamente podem ser aplicados às redes Bluetooth *Mesh*.

### 2.1.2 Desempenho do Bluetooth *Mesh*

Após a publicação da norma Bluetooth *Mesh*, autores ligados a institutos de pesquisa ou a própria indústria de hardware iniciaram trabalhos para verificar o desempenho de redes Bluetooth *Mesh*.

Testes realizados por (DI MARCO et al., 2017) foram elaborados para diferentes tamanhos e taxas de transmissão para verificar a taxa de entrega de pacotes de uma rede

Bluetooth *Mesh*. Foram realizados testes para uma rede de baixa densidade composta de apenas 12 nós, e testes para uma rede de 49 nós. Para cada rede foram testados 3 diferentes taxas de transmissão, sendo 150 bps, 1 kbps e 3 kbps. A simulação com menor densidade teve em torno de 15% de queda na taxa de entrega de pacotes para uma maior taxa de transmissão. No entanto para uma rede mais densa (composta de 49 nós), o aumento no número de perdas chegou até 30% para um tráfego intenso de 3 kbps. Nesses testes, foi observado que a redução na taxa de transmissão eleva o sucesso na taxa de entrega de pacotes.

Porém, para melhorar o desempenho da rede, foi adicionado um método de replicação de cada mensagem retransmitida. Nessa modificação, cada mensagem retransmitida, era replicada 3 vezes para aumentar a possibilidade de chegar ao destino. Essa modificação segundo os testes, levou a taxa de entrega chegar aos 99% para as 3 taxas de transmissão. Apesar disso, o próprio estudo indica que o método de *Flooding* do Bluetooth *Mesh* aliado ao método de replicação utilizado nos testes pode trazer problemas de congestionamento para redes maiores, o qual será investigado nesse trabalho.

Outra bateria de testes foi realizada pela Silabs (SILICON LABS, 2018) para verificar o desempenho do Bluetooth *Mesh* em relação a escalabilidade e latência. Para os testes, cada mensagem recebida era retransmitida após um *Delay* de 10 milissegundos. Outro aspecto testado foi o tamanho do pacote e a utilização de pacotes fragmentados em *payloads* acima de 11 bytes. Os testes foram executados em um escritório, onde os sensores foram instalados em salas de reunião, corredores e salas abertas. Havia concorrência com redes sem fio de outros protocolos como o *Zigbee* e o Wi-Fi.

Os testes foram divididos em 2 partes, para pacotes simples de até 8 bytes e para pacotes fragmentados. Os resultados mostraram que o aumento da latência varia de acordo com o crescimento da rede para pacotes não fragmentados. Nos testes foi visto que a média de latência varia de 20 milissegundos para 24 nós, passando por 30 milissegundos para 48 nós, 40 milissegundos para 144 nós e 50 milissegundos para 192 nós. Apesar desses resultados, um teste para um pacote de 32 bytes fragmentado em mais pacotes foi realizado. Na mesma quantidade de nós, foi visto que para 24 nós, a latência passou de 20 para 40 milissegundos, enquanto para 48 nós a latência que era de 30 milissegundos, agora é de 320 milissegundos. Para 96 nós a latência é de 480 milissegundos, para 144 nós é de 660 milissegundos e para 192 nós é de 800 milissegundos. Estes testes mostraram o aumento da latência com o crescimento da rede para pacotes únicos, e um incremento ainda maior para pacotes fragmentados, o que pode ser justificado pelo maior número de mensagens

circulando pela rede.

Um último teste foi realizado para identificar o efeito da quantidade de nós *Relay* na rede. Para uma rede de 240 nós, todos sendo *Relay*, testes mostraram que 10.21% dos pacotes recebidos foram com uma latência superior a 200 milisegundos. Já para a mesma quantidade de nós, porém com uma proporção de 1 nó *Relay* para cada 6 nós, apenas 1.44% dos pacotes foram recebidos após 200 milisegundos. Segundo (SILICON LABS, 2018), em uma rede com uma proporção mais baixa de nós *Relay* ocorre uma latência menor se comparado com uma rede 100% *Relay*, o que pode ser facilmente justificado pela redução na circulação de cópias retransmitidas na rede, tendo dessa maneira, uma redução de congestionamento. O autor (SILICON LABS, 2018) ainda conclui que o ideal para a rede Bluetooth *Mesh* é utilizar pacotes não fragmentados, e conforme a rede aumenta, o ideal é diminuir a proporção de número de *Relay* para reduzir o congestionamento.

Devido ao baixo desempenho do Bluetooth *Mesh* visto nesse trabalho, no próximo Capítulo serão apresentados métodos já utilizados em outros protocolos para incrementar o desempenho do Bluetooth *Mesh* tanto em taxa de entrega de pacotes quanto em eficiência energética.

## 2.2 INCREMENTANDO DESEMPENHO DO *FLOODING*

Esta Seção descreve os métodos que serão aplicados no padrão Bluetooth *Mesh* para melhorar a eficiência energética.

### 2.2.1 Algoritmo de Clusterização

Uma abordagem utilizada para melhorar a eficiência energética em redes com *Flooding* é o método de clusterização. Em uma rede clusterizada, o roteamento é realizado de forma hierárquica onde alguns nós possuem papéis especiais dentro da rede (chamados de *Cluster Heads* - CH). No entanto, em redes com *Flooding* como o Bluetooth *Mesh*, é implementado a mesma função de retransmissão para todos os nós, o que pode causar congestionamento e consumo expressivo de energia. Desse modo, a clusterização, com a aplicação de roteamento hierárquico pode melhorar a eficiência energética e a escalabilidade, onde reduz o número de troca de pacotes entre os sensores. Como a norma do Bluetooth *Mesh* permite que um nó assuma mais de uma função na rede, o método de roteamento hierárquico pode ser utilizado nesse trabalho.

Um dos primeiros e mais conhecidos esquemas de roteamento hierárquico é o chamado *Low Energy adaptive clustering hierarchy* (LEACH) (HEINZELMAN et al., 2000). Devido a sua ampla aceitação e fácil implementação, essa técnica será utilizada como base no esquema de clusterização proposto nesse trabalho. O principal objetivo do LEACH é melhorar a eficiência energética através da seleção arbitrária e randômica de nós em uma rede para se tornarem CHs. A probabilidade de um nó se tornar um CH é atribuída para que a distribuição do consumo de energia entre os nós seja uniforme. Isso é realizado pela alternância periódica dos CHs, toda vez que o consumo de energia desse nó especial seja superior aos demais. Nós não CH irão escolher um CH baseados na intensidade do sinal para se associarem e entrarem no seu *Cluster*. No LEACH original visto em (HEINZELMAN et al., 2000), os CHs são responsáveis pela criação de um agendamento TDMA para coordenar as transmissões e associações dos nós no *Cluster*. Esse processo tende a economizar uma vasta quantidade de energia, já que permite que nós que não estão em modo de transmissão possam dormir. Isso é permitido, pois é de responsabilidade do CH transmitir, armazenar e receber pacotes que são destinados aos membros do *Cluster*.

Várias melhorias para aplicações do protocolo LEACH são encontradas com foco em objetivos particulares para formar uma rede. Uma pesquisa detalhada sobre as modificações do LEACH podem ser encontradas em (SINGH et al., 2017) e variantes que serão utilizadas no desenvolvimento do algoritmo serão apresentadas nos próximos parágrafos.

Os trabalhos de (DEPEDRI et al., 2003; BURATTI et al., 2005) modificaram o protocolo LEACH para permitir a comunicação com multi saltos entre *Cluster Heads* e o nó central. Esses trabalhos foram realizados através de modificações matemáticas que apresentaram um melhor balanceamento na distribuição dos CHs. Também apresentou uma melhoria na fase de formação de um *Cluster*, onde cada nó seleciona um CH, o qual necessita menor consumo de energia na transmissão de pacotes para o nó central.

Outro trabalho com LEACH é encontrado em (YASSEIN et al., 2009). O autor em seu trabalho, cita que a operação da rede inicia assim como no LEACH original. Para manter a operação da rede, a energia residual de cada sensor é usado como parâmetro para escolher um novo CH. Ainda pode ser encontrado o C-LEACH (*cell leach*) (YEKTAPARAST et al., 2012), o qual propõe uma rede dividida em várias células em forma hexagonal. Nesse método, cada hexágono contém vários nós e um nó com papel de CH que participa de um rodízio periodicamente. Sete células próximas formam

um *Cluster* da rede, o qual contém um *Cluster Head*. O TDMA gerencia a transmissão de dados entre os nós e as *Cell Heads* e entre *Cell Heads* e o CHs. De acordo com o autor, esse protocolo aumenta de forma eficiente a cobertura pela rede, além de ser mais escalável e também economizar maior quantidade de energia quando comparado ao LEACH original. Por outro lado, esse ganho é acompanhado de maior complexidade e sobrecarga de mensagens.

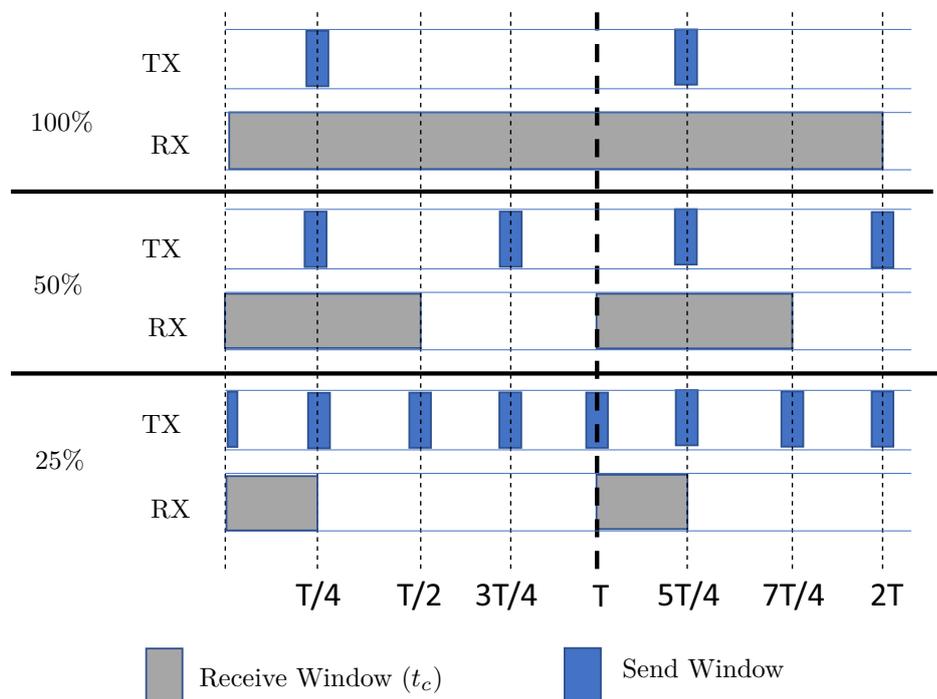
### 2.2.2 Algoritmo de *Radio Duty Cycling*

A especificação dada pela SIG para o Bluetooth *Mesh Profile Specification* (BLUETOOTH SIG, 2017) apresenta que para evitar perda de mensagens recebidas, o nó deve ouvir o canal 100% do tempo. Isso se deve a aplicação da técnica de *Flooding*, a qual gera deficiência na entrega de pacotes e consumo exagerado de energia. Consequentemente, essa atividade extensiva acaba esgotando a vida útil da bateria, especialmente para *Cluster Heads* que precisam reencaminhar mensagens de todos os nós participantes do *Cluster*. Para diminuir esse efeito, será proposto o uso da técnica de *Duty-Cycle*, as quais diminuem o tempo em que o rádio receptor do nó fica ligado. Seguindo essa direção, algumas mensagens podem ser perdidas, caso os nós estejam desligados no momento da chegada de pacotes, mas esse efeito pode ser reduzido com a utilização do algoritmo de *Radio Duty Cycling*.

Algoritmos de *Radio Duty Cycling* são classificados como assíncronos ou síncronos. Entre os modelos síncronos pode-se citar o Bluetooth *Low Energy* quando opera em modo Mestre/Escravo e o IEEE *Time Switching Channel Hopping* (IEEE STD 802.15.4E, 2012). Esses algoritmos apresentam grande robustez porque combinam operações livres de colisão devido a multiplexação no tempo e salto em frequência. Por outro lado, os esforços de sincronização tornam o método síncrono mais complexo de operar. Algoritmos assíncronos, por outro lado, não necessitam sincronização entre mestre e escravos, o que facilita sua implementação em processadores com menor capacidade de processamento. Um dos principais algoritmos encontrados nesse contexto é o ContikiMAC (DUNKELS, 2011), o qual define em sua operação um período ( $t_c$ ) para ouvir o canal. Como visto na Figura 6, cada nó receptor ouve o canal a cada  $t_c$  segundos. Dessa forma, cada emissor precisa repetir a mensagem pelo menos uma vez a cada tempo  $t_c$  segundos para mensagens *broadcast* ou até o recebimento de um ACK no caso de transmissões *unicast*. Se a transmissão do pacote for detectada durante o período de escuta do canal, o receptor permanece habilitado para receber o pacote. Esse tempo

garante que pelo menos um envio de mensagem seja recebida enquanto os receptores estão ouvindo o canal.

**Figura 6** – A Figura mostra o esquema de *Duty-Cycle*. Para um *Duty-Cycle* de 100%, o rádio de recepção fica ligado o período todo, enquanto um pacote é enviado pela transmissão. Para 50%, é necessário que 2 pacotes sejam enviados a cada meio período para garantir que pelo menos um pacote chegue ao destino que possui o receptor ligado em meio período. Já para 25%, será necessário enviar um pacote a cada 1/4 de período para pelo menos um pacote chegar ao destino, o qual possui o receptor ligado 25% do tempo.



Fonte: Adaptado de (DUNKELS, 2011)

O método de *Duty-Cycle* é efetivo em rede de sensores sem fio, pois vários rádios podem economizar energia em recepção enquanto apenas um transmissor gasta mais energia em transmissão (SUN et al., 2009; DUNKELS, 2011), visto que o consumo em RX é similar a TX. Esse comportamento é benéfico para rádios de baixa disponibilidade de energia porque pode prolongar a vida útil de sua bateria. Por outro lado, esse processo tende a aumentar o congestionamento de pacotes em redes *Flooding* com a diminuição do *Duty-Cycle*. Isso acontece pois a diminuição do tempo de rádio ligado, necessita maior quantidade de cópias de pacotes que devem ser enviados, para pelo menos um chegar no receptor, quando o mesmo estiver com rádio ligado. Esse efeito será investigado no Capítulo 4.

No Capítulo 3, será apresentado a proposta de adaptação do protocolo Bluetooth *Mesh* com a utilização da técnica de clusterização e *Duty-Cycle* presentes na Seção 2.2.

### 3 ALGORITMO PROPOSTO

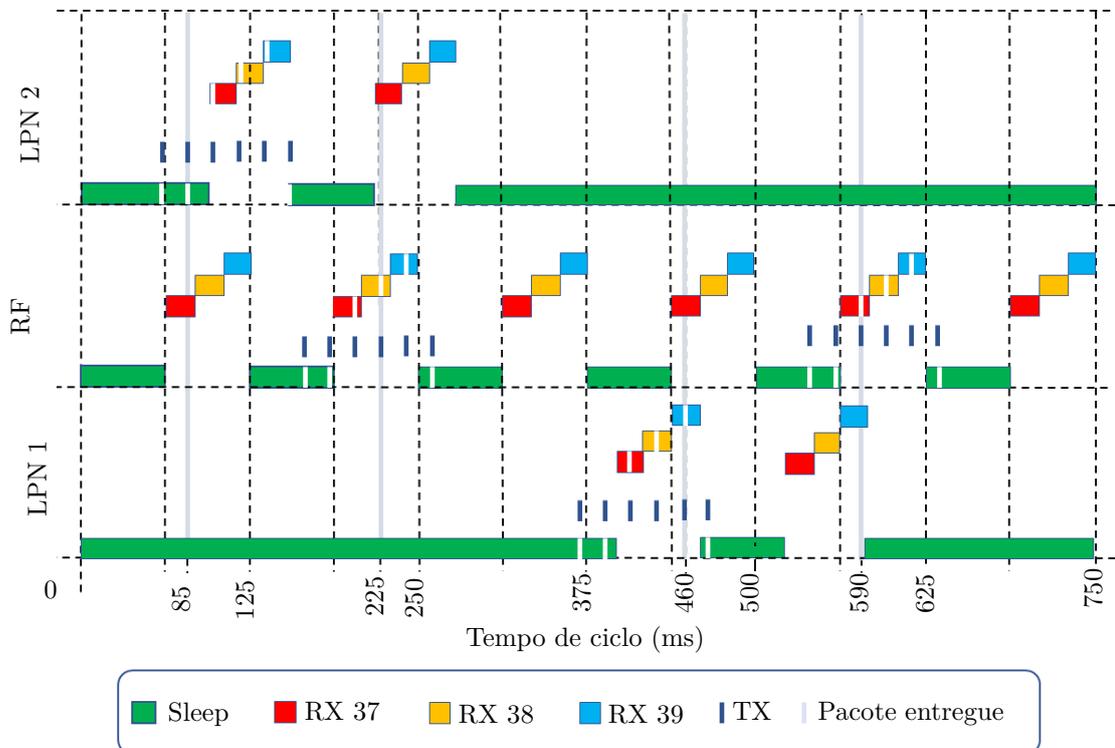
Devido aos problemas de baixo desempenho em entrega de pacotes e eficiência energética em redes *Flooding*, uma extensão da operação do protocolo BLE *Mesh* será apresentada. Nos próximos Capítulos, os resultados da proposta serão demonstrados a partir de simulações realizadas.

O primeiro ponto de melhoria apresentado nesse artigo para aumentar a eficiência energética do Bluetooth *Mesh* é a técnica de *Radio Duty Cycling*. Nesse trabalho foi optado utilizar o modelo simplificado do ContikiMAC (DUNKELS, 2011) como mencionado na Seção 2.2.2 pois a avaliação do canal livre nem sempre está disponível para os comandos padrões do Bluetooth. Nesse caso, a janela de tempo em modo de escuta do canal deve ser reduzida a um valor arbitrário e várias repetições espaçadas devem ser feitas para garantir que pelo menos uma mensagem seja enviada enquanto os rádios receptores estiverem ouvindo como visto na Figura 7. Por exemplo, para um *Duty-Cycle* de 50%, duas repetições de pacotes precisam ser realizadas dentro de um período de ciclo. Assim, pelo menos um pacote chegará ao receptor enquanto ele estiver ouvindo o canal. Se o tempo de escuta do *Duty-Cycle* for menor ou igual a 25%, então 4 transmissões dentro de um período de tempo precisam ser realizadas, como visto na Figura 6.

Um exemplo de transmissão utilizando o protocolo proposto é observado na Figura 7. Nela, é possível observar a troca de mensagens entre 3 nós, sendo um RF, e dois LPNs com um *Duty-Cycle* de 50%, onde cada período é composto de 125 milissegundos, nos quais cada nó RF permanece 62.5 milissegundos em modo *Sleep* e o restante do tempo ouvindo pelos canais de *Advertisiment* 37, 38 e 39. Desse modo, 2 tentativas de retransmissão são efetuadas para cada canal para que pelo menos um pacote chegue ao destino. Nessa imagem, é possível observar que o nó LPN2 envia 2 pacotes sucessivamente pelos canais 37, 38 e 39 e que uma tentativa *Poll Request* realizada no instante 80 milissegundos alcança o nó RF. O nó RF recebe o pacote pelo canal 37. Após receber verifica se há pacotes armazenados e disponíveis para o LPN2. Como há somente um pacote, ele desliga o rádio RX no instante 125 milissegundos e envia várias tentativas de transmissão para o LPN2, sendo que uma, no instante 225 milissegundos com um MD=0 chega ao destino pelo canal 37. Ao identificar que não há mais pacote armazenado para ele, volta a dormir instantes depois.

Ainda é possível observar na Figura 7 a troca de mensagens entre LPN1 e RF. Próximo ao instante 375 milissegundos, prestes a acordar, o LPN1 inicia o envio de 2

Figura 7 – Processo de troca de mensagens para um *Duty-Cycle* de 50%, onde cada nó RF permanece 50% em modo *Sleep* e o restante do tempo ouvindo através dos canais 37 ao 39. Os nós LPNs dormem por um longo período de tempo e ao acordar requisitam mensagens ao RF. Após receberem todas as mensagens, os nós LPNs voltam a dormir.



Fonte: Autoria própria

pacotes por canal para atingir ao RF. Um deles chega ao canal 37 no instante 460 milissegundos. Esse *Poll Request* é recebido pelo RF que verifica em sua *queue* quantas mensagens estão armazenadas para esse LPN. Após identificar que há somente um pacote, ele responde com um *Poll Response* sucessivamente enviando dois pacotes por canal. Um deles chega ao destino no instante 490 milissegundos com MD=0. Logo ao receber a mensagem pelo canal 39 e perceber que não há mais mensagens armazenadas para esse nó, ele volta a dormir.

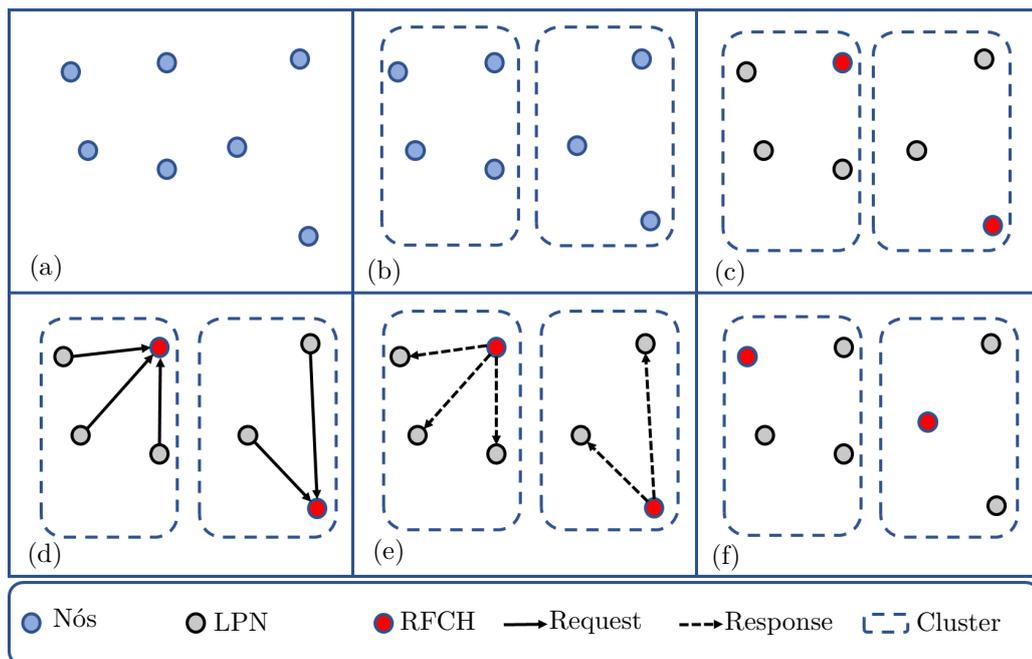
Assim finaliza-se um processo completo de troca de mensagens entre RF e LPNs. Esse processo reduz significativamente o consumo energético, pois possibilita que o nó RF possa dormir por meio período de ciclo, onde em teoria reduz 50% o consumo de energia total do rádio receptor para *Duty-Cycle* de 50%. Porém, por outro lado dobra o consumo de energia utilizada pelo transmissor. Mas como o tempo de transmissão é insignificante comparado com a recepção, esse método justifica sua aplicação.

A segunda técnica para melhorar o desempenho tanto em eficiência energética

como na entrega de pacotes é a limitação da quantidade de retransmissões de pacotes (LEVIS et al., 2004). Essa limitação pode beneficiar redes susceptíveis a congestionamentos, onde o *Flooding* produz redundância de pacotes e aumenta colisões. Como os nós LPN não retransmitem mensagens, será utilizado o modelo geográfico de clusterização baseado no C-LEACH (YEKTAPARAST et al., 2012). Nesse método, cada CH é configurado como *Relay-Friend* e os demais membros do *Cluster* são configurados como LPNs.

Na Figura 8 é possível observar que em (a) os nós estão distribuídos em uma área. Em (b), ocorre a formação dos *Clusters* de acordo com a localização dos nós. Em (c) um dos nós é escolhido inicialmente de forma randômica para se tornar o *Cluster Head* dentro de cada *Cluster*. Os demais se tornam LPN. Em (d) e (e), ocorre o processo de troca de mensagens entre os nós LPN e o *Relay-Friend Cluster Head* (RFCH), onde cada LPN da rede irá enviar um *Poll Request* para o RFCH e será respondido via *Poll Response*. Através do *Poll Request*, o RFCH recebe informações de energia residual de cada nó da rede. E após o RFCH atingir um limite crítico de energia, ele seleciona um dos seus LPNs com maior quantidade de energia para ser o novo RFCH do *Cluster* e comunica todos os nós do *Cluster* através do *Poll Response*. Em (f) o *Cluster* é novamente formado com um novo RFCH e o processo se repete a partir do item (d).

**Figura 8 – Demonstrativo do processo de formação do *Cluster*, escolha e rodízio dos CHs.**



Fonte: Autoria própria

O papel do nó *Relay Friend Cluster Head* (RFCH) é revezado por todos os nós do *Cluster*, quando um RFCH atingir o nível máximo configurado de consumo de energia. O nó LPN com maior disponibilidade de energia residual no *Cluster* se torna o novo RFCH. Nesse momento, o antigo RFCH ao avisar todos os nós LPNs da mudança, assume o papel de LPN e pode então dormir. O processo de formação do *Cluster* e revezamento dos nós pode ser observado na Figura 8. Esse procedimento, além de limitar a quantidade de pacotes que congestionam a rede, também aumentam consideravelmente a duração da bateria dos nós, já que todos os nós do *Cluster* se tornarão LPN após atingirem um limite máximo de consumo energético. Logo é possível obter um equilíbrio energético entre todos os nós da rede. Esse processo otimiza e escala a rede se comparado com o Bluetooth *Mesh* convencional, onde não há um rodízio de funções realizadas pelos nós.

O método de troca de mensagens entre RF e LPN por *Poll Request/Response* é utilizado para gerenciar o *Cluster*. A energia residual de cada LPN é enviada via *Poll Request* para o RFCH. Este ao receber o pacote de todos os LPNs, monitora o momento de mudança de *Cluster Head*. Após verificar que sua energia chegou ao limite, o CH observa a energia residual de todos os nós do grupo e envia um *Poll Response* com a indicação de qual nó LPN irá se tornar o novo RFCH. Apesar do aumento de geração de pacotes, essas mensagens já são trocadas entre LPNs e *Friends* para a operação de uma *Friendship*. Logo, esse procedimento de troca de mensagens gera uma sobrecarga imperceptível na rede comparada aos efeitos negativos gerados por uma rede Bluetooth *Mesh* não clusterizada.

Além disso, o método de clusterização irá controlar o número de cópias replicadas de pacotes gerados pelo *Flooding*, devido a menor quantidade de nós RF's, o que impacta positivamente na questão da escalabilidade da rede. Nesse estudo preliminar, cada nó tem a informação inicial da localização geográfica, a qual é usada para iniciar a formação dos *Clusters*. O primeiro *Cluster* é escolhido de forma randômica e troca de CHs é realizada via *Poll Request/Response*.

## 4 RESULTADOS

Nesse Capítulo serão apresentados os resultados obtidos das simulações de uma rede Bluetooth *Mesh*. Durante o desenvolvimento do trabalho não foi encontrado um simulador que modelava as camadas do Bluetooth e BLE *Mesh* que eram necessárias para a avaliação da técnica proposta, por isso foi desenvolvido um ambiente de simulação o qual será descrito na Seção 4.1.

### 4.1 AMBIENTE DE SIMULAÇÃO

Essa Seção apresenta de forma detalhada a construção do ambiente de simulação. Esse simulador foi construído sobre um gerenciador de eventos discretos chamado Simpy (SIMPY, 2002). Enquanto o Simpy gerencia os eventos discretos, as lógicas de transmissão das camadas física e de roteamento vistas na norma (BLUETOOTH SIG, 2017) foram implementadas.

Para a camada física, os rádios receptores e transmissores operam pelos 3 canais de *Advertisement*. Nesta versão inicial, o modelo de propagação foi baseado no *Cooja Unit Disk Graph Radio Medium* (OSTERLIND et al., 2006), onde apenas a perda de percurso entre nós é considerada para decodificação e interferência. Porém outras métricas de perdas por *Fading* e multipercurso devem ser estudadas e implantadas no simulador em trabalhos futuros. O coeficiente da perda de propagação é  $\alpha = 2.08$ , os parâmetros do rádio são baseados no *Nordic Semiconductor nRF51822* (NORDIC SEMICONDUCTOR, 2019) Bluetooth SoC, o qual é demonstrado na Tabela 1. O sucesso da recepção ocorre quando a potência recebida está acima da sensibilidade do nRF51822 e nenhuma outra transmissão acima da faixa de sensibilidade for recebida no canal atual durante toda a duração da transmissão de pacotes.

A primeira versão do simulador teve como foco implementar as lógicas de operação para os nós *Relay-Friend* e LPN. Embora seja possível realizar a implantação e simulação do nó *Proxy* para realizar a comunicação através do GATT *Bearer* com dispositivos com versão igual ou superior ao BLE 4.0. Por isso, esse desenvolvimento deve constar nas próximas versões do simulador.

Na camada de roteamento foram aplicadas as regras de cache de retransmissão, para evitar a retransmissão de um mensagem por mais de uma vez pelo mesmo nó, e a transmissão dos pacotes com o TTL para reduzir o congestionamento da rede. Para os

Tabela 1 – Parâmetros do Sistema.

Parameter	Value	Unit
Coefficiente de perda de percurso	2.08	-
Sensibilidade do receptor	-93.0	dBm
Potência de transmissão	4.0	dBm
Tensão do SoC	3.0	V
Corrente de transmissão do SoC	10.5	mA
Corrente de recepção do SoC	13.0	mA
Corrente de CPU ligada do SoC	4.8	mA
Corrente de CPU em sleep do SoC	2.8	$\mu$ A
Frequência de transmissão	2.4	GHz

Fonte: Adaptado de (NORDIC SEMICONDUCTOR, 2019)

nós LPNs foi desenvolvido as trocas de mensagens por *Poll Request/Response*, inclusive para realizar a troca do CH em um *Cluster*. Mas devido a complexidade do trabalho, o processo de iniciação de uma *Friendship*, o qual possibilita que um nó LPN encontre um nó *Friend* na rede deve ser construído para a próxima versão do ambiente de simulação.

## 4.2 RESULTADO DAS SIMULAÇÕES

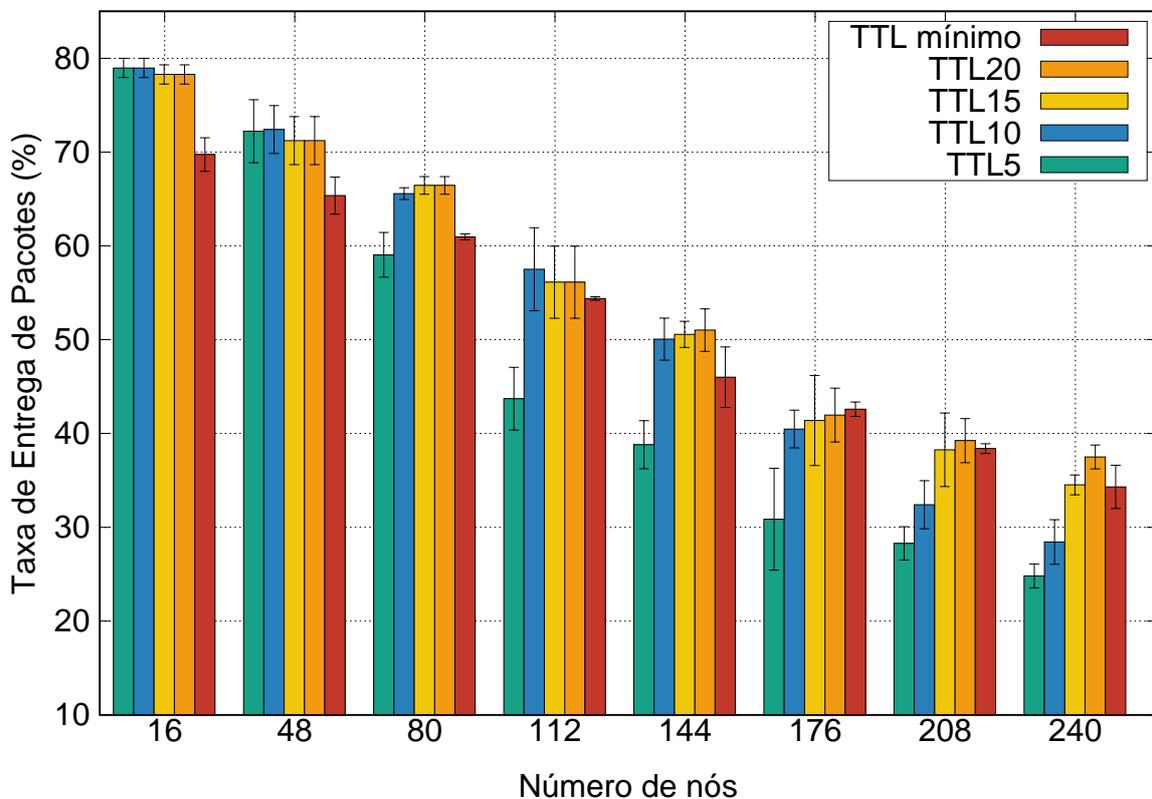
Nessa Seção serão apresentados os resultados obtidos das simulações de uma rede Bluetooth *Mesh* utilizando o ambiente de simulação descrito na Seção 4.1. Após sua construção, o trabalho consistiu na implementação de alguns métodos que podem aprimorar os resultados do Bluetooth *Mesh* e reduzir os efeitos negativos do *Flooding*. A simulação inicial consistiu em verificar o desempenho do protocolo Bluetooth *Mesh* convencional.

Para esta primeira simulação, cada pacote foi transmitido sequencialmente pelos 3 canais de *Advertisement*, iniciando pelo 37 até o 39. As regras do Bluetooth *Mesh Profile* (BLUETOOTH SIG, 2017) para as camadas física e MAC foram implementadas de acordo com a configuração dos tipos de nós. Os valores de TTL mínimo para cada mensagem foram encontrados através da aplicação do algoritmo de Dijkstra.

A taxa de entrega de pacotes de uma rede em relação ao crescimento em número de nós para diferentes valores iniciais de TTL e utilizando um intervalo de confiança de 95% podem ser vistas na Figura 9. Nessa rede 25% dos nós são configurados como *Relay-Friend* e 75% como nós LPNs. A geração de pacotes segue a distribuição de *Poisson* com

10 segundos de tempo entre transmissão de pacotes. Os nós são distribuídos de forma randômica em uma área retangular de tamanho  $(22 * i)$ , onde  $i$  é o número total de nós da simulação. Cada simulação é dividida em quatro rodadas com posicionamentos aleatórios diferentes para cada nó em cada rodada. A simulação roda até cada nó enviar pelo menos 70 pacotes.

**Figura 9 – Taxa de entrega de pacotes % vs número de nós para diferente valores de TTL para uma rede composta de 25% de nós *Relay-Friend* e 75% de LPNs. TTL mínimo composto pelo mínimo de saltos mais uma margem de 40%. Média de geração de pacotes: 10 segundos.**

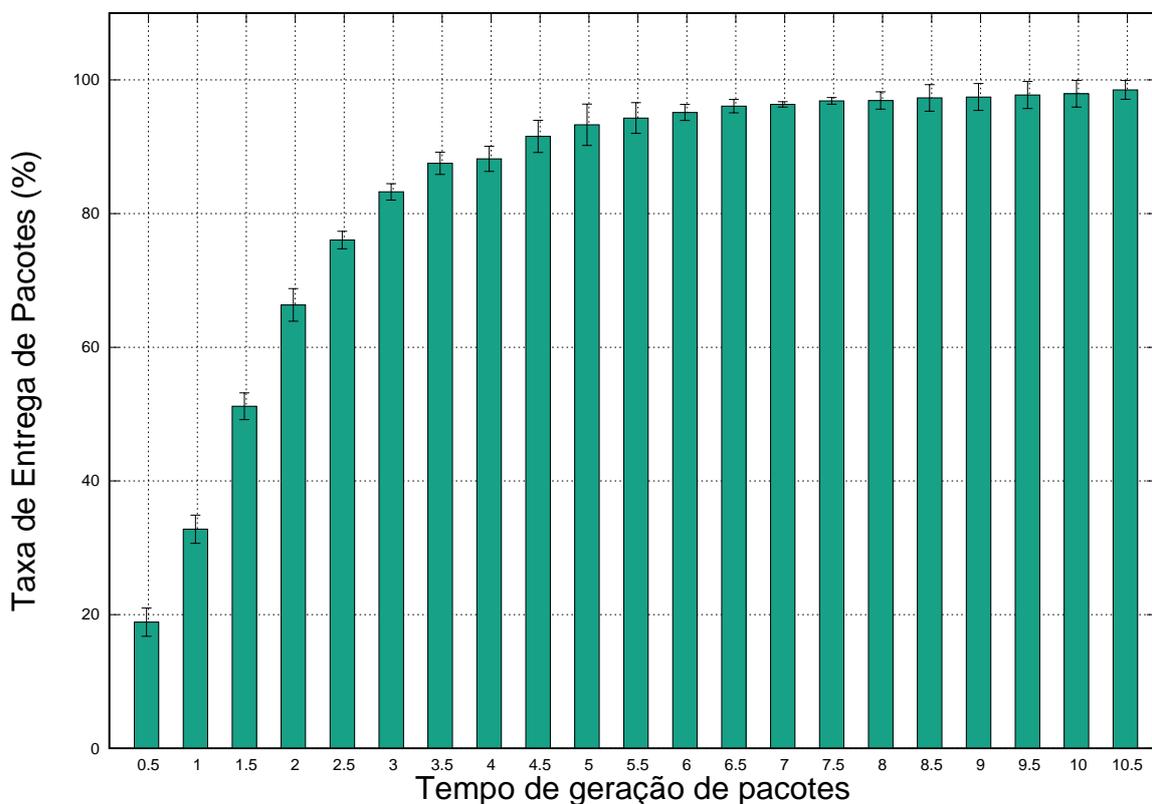


Fonte: Autoria própria

Na Figura 9, pode ser observado que a entrega de pacotes decremента com o aumento da rede. Esse comportamento é causado por dois efeitos combinados. Primeiro o número de saltos necessários para uma mensagem sair de um emissor e chegar à um receptor aumenta com a adição de mais nós, e dessa forma se uma mensagem for configurada com um TTL baixo, ela será perdida no caminho. Por outro lado, se o TTL da mensagem for maior que o necessário, excessivas retransmissões serão geradas, o que congestiona a rede e gera eventos de colisão. No entanto, o aumento do TTL, mesmo considerando o TTL ideal tem um certo efeito positivo sobre a entrega de pacotes, pois

essas perdas acabam anulando as perdas por TTL baixo. Além disso, o TTL mínimo para esse trabalho, foi gerado pelo algoritmo de Dijkstra e teve um aumento de 40%. Essa margem foi utilizada devido a testes empíricos que mostraram a baixa variação tanto na entrega de pacotes como na eficiência energética para diferentes margens. Por isso, pode-se concluir que o TTL otimizado não é suficiente para controlar a escalabilidade da rede. Dessa maneira, outros métodos precisam ser investigados.

**Figura 10 – Taxa de entrega de pacotes (%) vs intervalo de geração de pacotes. Tamanho da rede de 16 nós, geração de pacotes com TTL=20.**



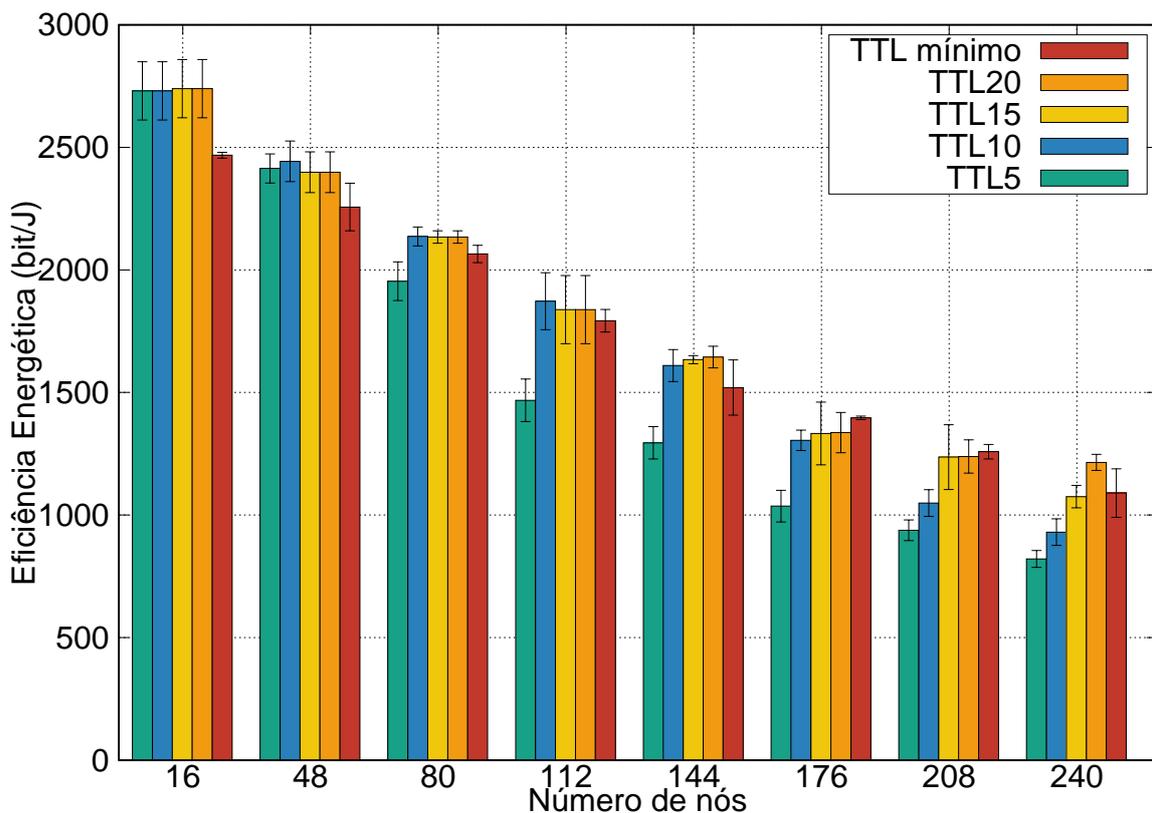
**Fonte: Autoria própria**

Para complementar os resultados do experimento anterior, a Figura 10 mostra a entrega de pacotes (%) em função da média de tempo entre a geração de pacotes para uma rede de 16 nós com um TTL=20. Dessa figura, pode-se observar uma melhora na entrega de pacotes com o aumento do tempo médio entre a geração de pacotes. Por outro lado, quando a taxa de geração de pacotes é aumentada, o roteamento por *Flooding* satura rapidamente os canais de transmissão, o que diminui a entrega de pacotes devido as colisões. Além disso a questão das replicações de pacotes gerados pelo *Flooding* se tornam uma preocupação grande na questão da eficiência energética.

A Figura 11 mostra a eficiência energética em função do número de nós. Nela

pode-se observar que o consumo energético está fortemente relacionado com a taxa de entrega de pacotes (Figura 9). Pode-se argumentar que a eficiência energética deve diminuir quando o número de nós ou quando o TTL inicial da rede aumenta, pois, mais réplicas de pacotes são geradas.

**Figura 11 – Eficiência energética do Bluetooth *Mesh* em função do número de nós para diferente valores de TTL para uma rede composta de 25% de nós *Relay-Friend* e 75% de LPNs. TTL mínimo composto pelo mínimo de saltos mais uma margem de 40%. Média de geração de pacotes: 10 segundos.**

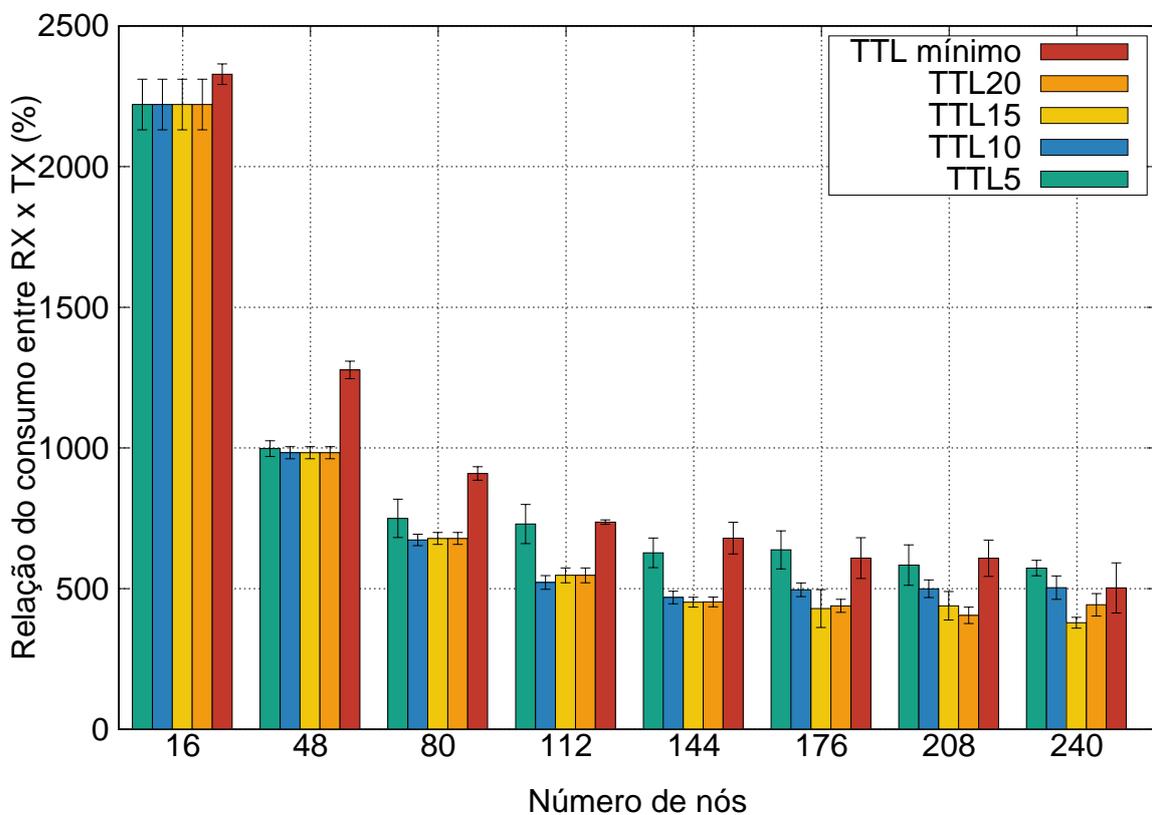


Fonte: Autoria própria

Com relação ao consumo energético, a Figura 12 mostra a porcentagem da relação do consumo energético do rádio RX comparado com o TX. Nela, é possível observar que o consumo total de energia do nó é dominado pelo período em que o nó permanece com o rádio receptor ligado, muito superior ao rádio transmissor. Na imagem, é possível observar que para uma rede com poucos nós, a relação de consumo em RX para TX chega até um valor em torno de 2220% para uma rede com pacotes com TTL 5, 10, 15 e 20. No entanto, o crescimento da rede aumenta o consumo do rádio transmissor. Esse efeito, é gerado pela maior quantidade de réplicas retransmitidas e circulando pela rede. Mesmo assim, o consumo de recepção é duas ordens de magnitude maior que o consumo de transmissão. A mesma figura, mostra que em uma rede com 240 nós e pacotes com

TTL=20, relação entre RX/TX diminui para 360%. Apesar dessa redução, o consumo significativo do rádio receptor motiva o uso da técnica de *Radio Duty Cycling* para reduzir o tempo de escuta do canal. O efeito desses métodos propostos serão investigadas nos próximos testes.

**Figura 12 – Porcentagem do consumo energético do rádio RX comparado com TX para uma rede Bluetooth Mesh em função do número de nós para diferente valores de TTL para uma rede composta de 25% de nós *Relay-Friend* e 75% de LPNs. TTL mínimo composto pelo mínimo de saltos mais uma margem de 40%. Média de geração de pacotes: 10 segundos.**

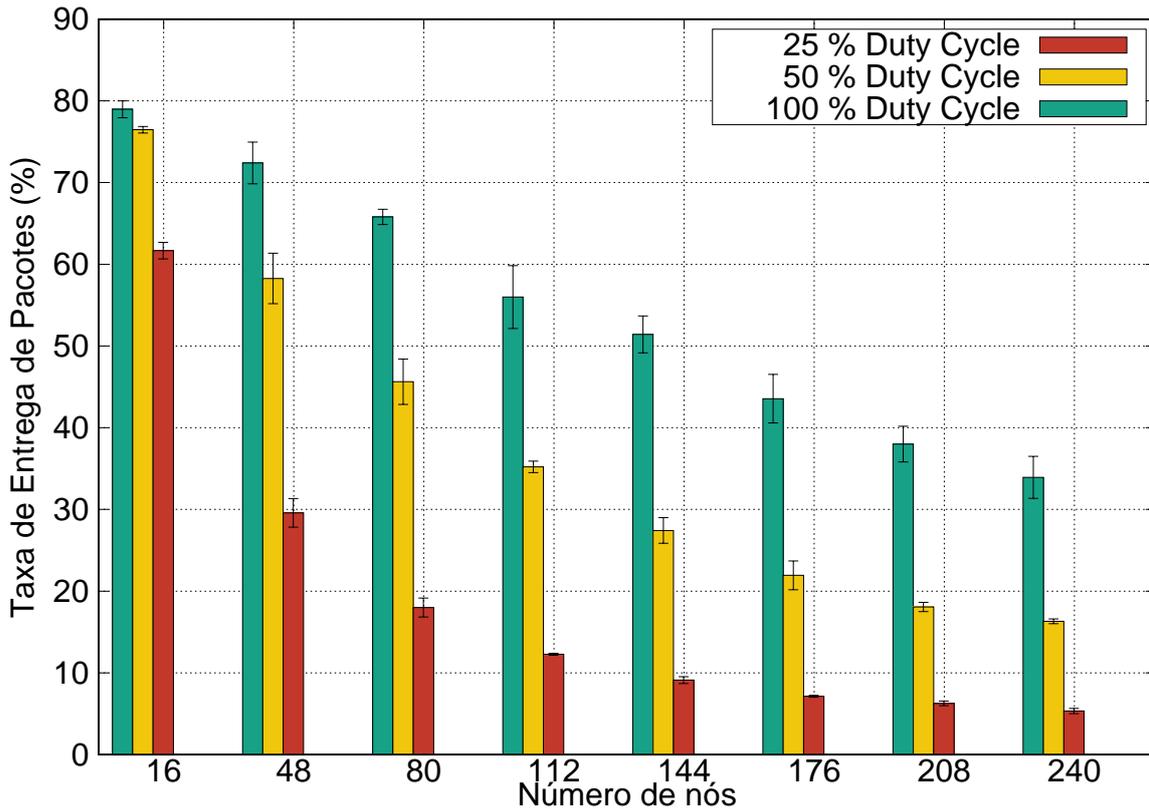


Fonte: Autoria própria

A entrega de pacotes para diferentes *Duty-Cycles* em função do número de nós pode ser vista na Figura 13. Nessa simulação 75% dos nós possuem a função de LPN e os demais 25% são *Relay-Friend*. As mensagens são geradas com um TTL fixo de 20 e a geração de pacotes segue uma distribuição de *Poisson* com média de 10 segundos. A rede com *Duty-Cycle* de 100% pode ser considerada como a aplicação do Bluetooth Mesh convencional, sendo aplicado dessa forma uma variação para 50% e outra para 25% de *Duty-Cycle*. Pode ser observado na figura que a entrega é afetada negativamente com a diminuição do *Duty-Cycle* pois os nós são mais susceptíveis a colisões com o aumento da rede. Esse efeito é observado porque a escolha da técnica do *Radio Duty Cycling* transmite

uma rápida sequência de cópias da mensagem, as quais congestionam a rede, e resultam em um número maior de colisões.

**Figura 13** – Taxa de entrega de pacotes em função do aumento do número de nós, para diferentes *Duty-Cycles* para uma rede composta por 75% de nós LPNs e 25% nós *Relay-Friend* e com uma geração de pacotes com TTL de 20 e utilizando a distribuição de *Poisson* com uma média de 10 segundos.



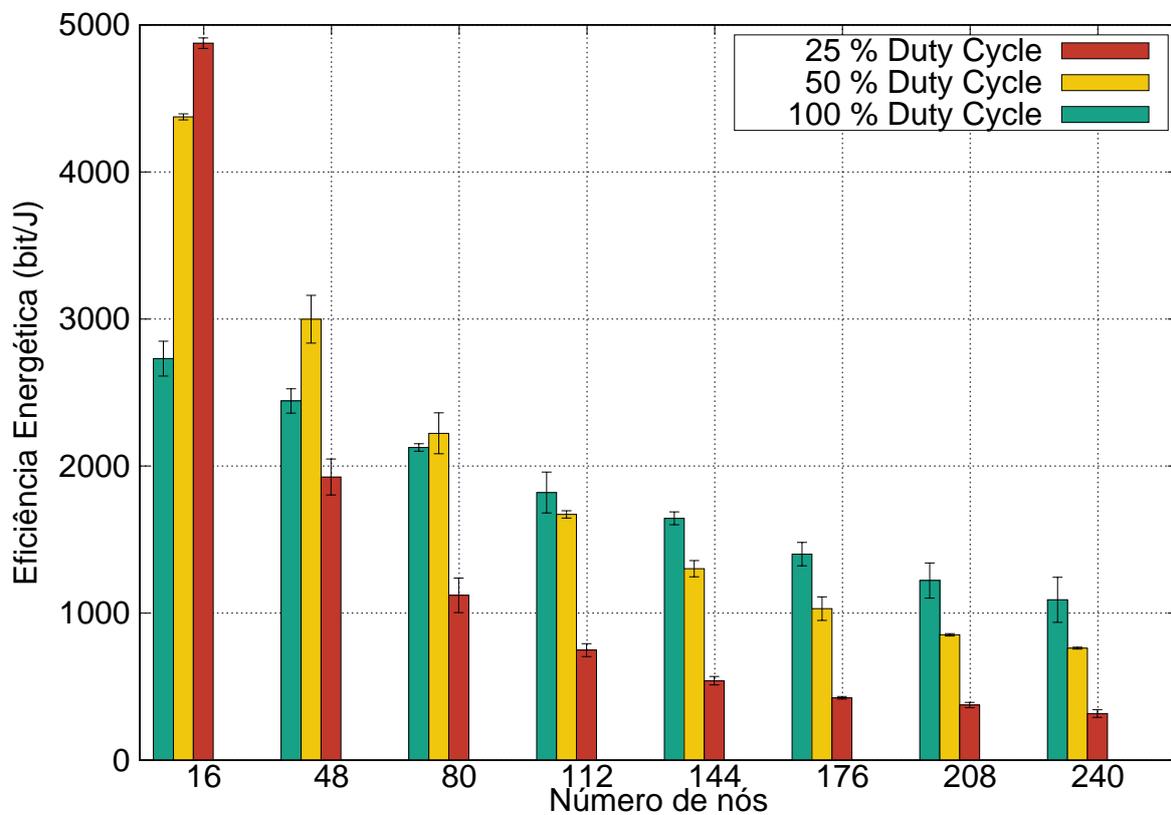
Fonte: Autoria própria

Com relação a eficiência energética, a Figura 14 mostra os resultados da mesma rede operando no regime de *Radio Duty Cycling*. Nela pode ser observado que a técnica traz uma economia que pode chegar até 90% para um *Duty-Cycle* de 25% comparado ao Bluetooth *Mesh* (*Duty-Cycle* de 100%). Na simulação para 16 nós, enquanto a eficiência energética é de 4875 bit/J para 25% de *Duty-Cycle*, o Bluetooth *Mesh* convencional tem 2730 bit/J, o que representa um ganho de 90%. De acordo com a imagem, para redes menores, a melhor estratégia é escolher um *Duty-Cycle* baixo, porém com o aumento da rede, a taxa de entrega degrada com maior rapidez impactando diretamente na eficiência energética, fazendo dessa forma, com que um *Duty-Cycle* maior seja mais eficiente.

Por outro lado, em redes com até 48 nós e menores valores de *Duty-Cycles*, há um expressivo ganho com eficiência energética apesar do aumento da degradação da taxa

de entrega de pacotes, o que justifica o uso da técnica de *Radio Duty Cycling*. Pois mesmo com o aumento da geração de cópias de pacotes transmitidos, é possível obter uma melhoria na eficiência energética. Outra vantagem desse método é que nenhuma modificação no protocolo Bluetooth *Mesh Profile* é necessária, apenas uma simples modificação na lógica de transmissão de pacotes.

**Figura 14 – Eficiência energética em função do aumento do número de nós, para diferentes *Duty-Cycles* em uma rede composta por 75% de nós LPNs e 25% nós *Relay-Friend* e com uma geração de pacotes com TTL de 20 e utilizando a distribuição de *Poisson* com uma média de 10 segundos.**

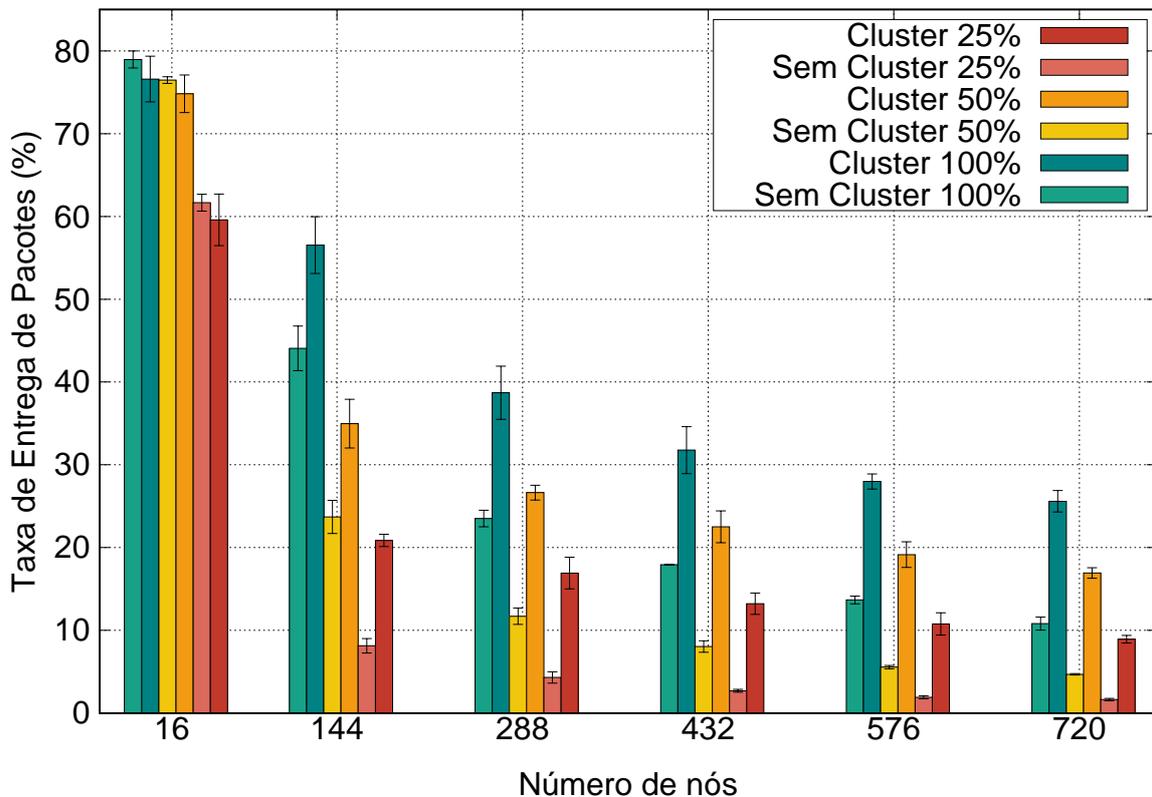


Fonte: Autoria própria

Porém se modificações são permitidas, o protocolo de clusterização pode ser adotado como visto no Capítulo 3. Ele pode ser usado para limitar a replicação de pacotes que geram a degradação em redes com baixo *Duty-Cycle*. Esses resultados estão presentes nos próximos experimentos onde cada RFCH foi adicionado em um *Cluster* com 3 nós LPNs. Para uma comparação similar e justa, uma rede não clusterizada foi criada com a mesma proporção de nós *Relay-Friend* e LPNs. Os pacotes gerados para ambas as redes foram em média de 10 segundos para uma distribuição de *Poisson* e cada rodada da simulação foi finalizada após o uso completo de energia da bateria do primeiro nó da rede. Na rede com clusterização, há rodízio de nós como RFCH, já nas simulações sem

clusterização, o nó possui do início ao fim a mesma função na rede. Nessa simulação, o Bluetooth *Mesh* convencional é representado pela rede Sem Cluster 100%. Também foi inserido na simulação uma rede clusterizada sem *Duty-Cycle* representada pela Cluster 100%.

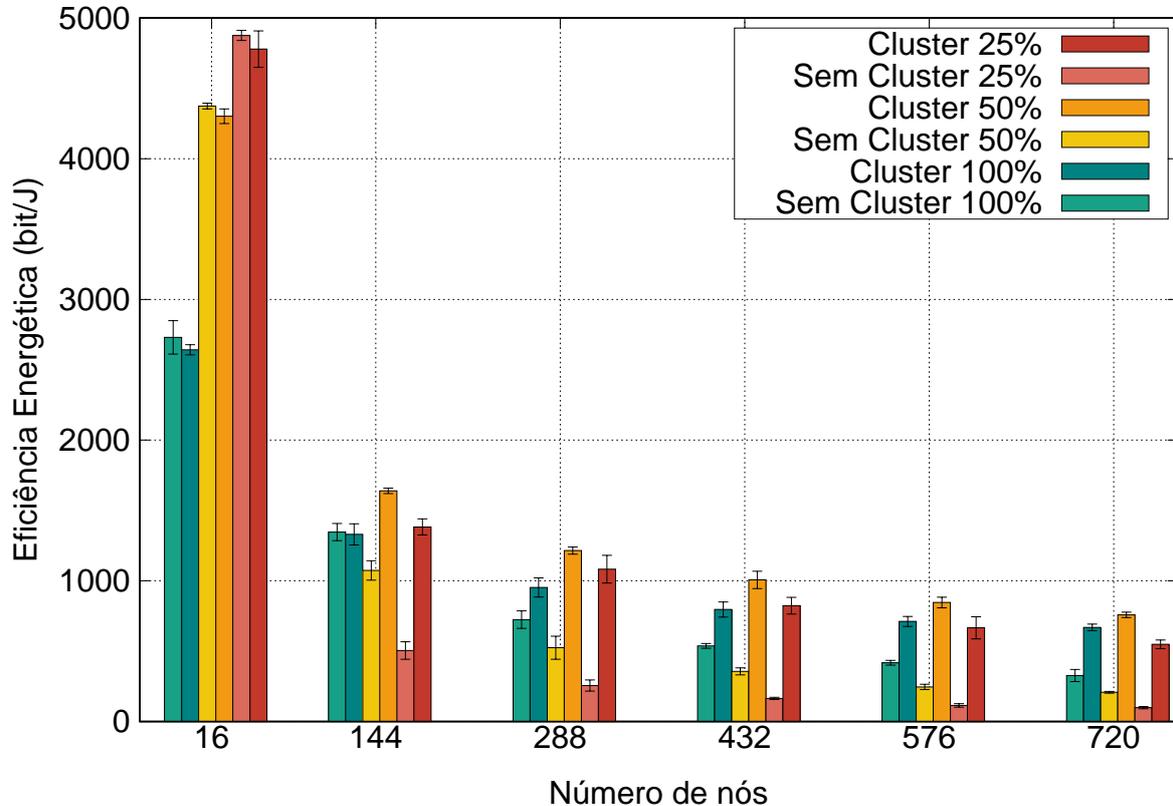
**Figura 15 – Taxa de entrega de pacotes em função do tamanho da rede para uma rede clusterizada e diferentes *Duty-Cycles*. Onde cada pacote é gerado com TTL 20, com distribuição de *Poisson* com 10 segundos de média. Rede composta por 75% de nós LPNs e 25% nós *Relay-Friend*.**



Fonte: Autoria própria

Figura 15 mostra a entrega de pacotes e Figura 16 mostra a eficiência energética de redes clusterizadas em função do tamanho da rede. Pode ser observado em ambas as imagens que a escolha do melhor método depende do tamanho e da prioridade da rede, seja em taxa de entrega de pacotes ou em eficiência energética. Quando comparado com o Bluetooth *Mesh* 100% de *Duty-Cycle*, o método de clusterização e as técnicas de *Radio Duty Cycling* possuem um efeito negativo para redes menores que 288 nós. No entanto, com o crescimento da rede, é possível observar um ganho considerável tanto em taxa de entrega como em eficiência energética, o que gera impacto positivo quanto a escalabilidade da rede.

Figura 16 – Eficiência energética em função do tamanho da rede para uma rede clusterizada e diferentes *Duty-Cycles*. Onde cada pacote é gerado com TTL 20, com distribuição de *Poisson* com 10 segundos de média. Rede composta por 75% de nós LPNs e 25% nós *Relay-Friend*.



Fonte: Autoria própria

Se comparar o método de clusterização com as técnicas de *Radio Duty Cycling*, é possível observar que a partir de 144 nós, é mais indicado utilizar clusterização aplicado com *Radio Duty Cycling* para obter ganhos maiores em taxa de entrega (Figura 15) e em eficiência energética (Figura 16) para *Duty Cycles* de 50%. Isso é observado nas imagens, onde as curvas de decaimento são mais suaves para as simulações de clusterização. Nelas, é possível observar que para a taxa de entrega de pacotes, o aumento da rede gera um melhor desempenho para simulações com clusterização aliadas ao *Duty Cycle* de 100%, seguido por 50%. No entanto, em termos de eficiência energética, é observado um fenômeno diferente, onde uma clusterização com 50% de *Duty Cycle* é melhor que as demais. Esse comportamento é justificável pois a perda de pacotes pode ser suprimida pela redução de consumo energético. O mesmo desempenho não é observado em simulações de redes com apenas as técnicas de *Radio Duty Cycling*, onde para 100% de *Duty Cycle*, há maior eficiência comparado com 50% e 50% maior que 25% tanto em entrega de pacotes como em eficiência energética e ainda degrada com o incremento de nós na rede.

Ainda é possível observar, que para redes menores, com até 16 nós, a melhor estratégia é utilizar o método de *Duty-Cycle* com 25% sem clusterização, o qual retorna um ganho significativo de 90% de aumento em eficiência energética comparado com o Bluetooth *Mesh* convencional, e um ganho de 55% para um *Duty-Cycle* de 50%. Porém o aumento da rede mostra que o método de clusterização possui um efeito melhor que o método de *Radio Duty Cycling*.

Os resultados vistos nas simulações acima, demonstram que os métodos de clusterização são justificados e podem ser utilizados juntamente com as métricas de *Radio Duty Cycling* para melhorar o desempenho de redes Bluetooth *Mesh* em taxa de entrega de pacotes e em eficiência energética.

## 5 COMENTÁRIOS FINAIS

### 5.1 CONCLUSÃO

Esse trabalho teve como propósito melhorar a eficiência energética de redes Bluetooth *Mesh* com a utilização de métodos de clusterização baseados em LEACH. Para reduzir a carga sobre os *Cluster Heads*, métricas do algoritmo de *Radio Duty Cycling* inspiradas no ContikiMAC foram utilizadas. A efetividade da solução proposta foi obtida pela utilização de simulações feitas em computador.

Este documento mostra que a eficiência energética de redes Bluetooth *Mesh* é dominada pelo tempo de escuta em RX do canal de nós *Relay* que é muito superior aos tempos de rádio ligado para transmissão. No entanto, foi observado que tanto o consumo energético como a taxa de entrega de pacotes estão relacionados com proporção de nós RF's na rede. Essa proporção gera efeitos contrários, pois enquanto uma grande proporção de RF's gera um efeito positivo na taxa de entrega de pacotes, a eficiência energética tem maior ganho para uma proporção maior de LPNs. Apesar desse comportamento, tanto a taxa de entrega como a eficiência energética degradam com o crescimento da rede. Por esse motivo o método de *Radio Duty Cycling* foi utilizado para reduzir o tempo de rádio em RX ligado e aliado com a Clusterização para limitar a retransmissão de pacotes e incrementar a eficiência energética.

Pode ser observado que o método de *Radio Duty Cycling* trouxe uma eficiência energética de até 90% para uma rede com 16 nós utilizando um *Duty-Cycle* de 25% e 55% de eficiência para um *Duty-Cycle* de 50%. Outra vantagem da utilização da técnica de *Radio Duty Cycling* é que nenhuma modificação no protocolo oficial do Bluetooth *Mesh* é necessária, apenas uma simples modificação na lógica de transmissão. No entanto esse método degrada com o crescimento da rede.

Ao pensar em escalabilidade, o método de clusterização mostra mais eficiência comparado com o Bluetooth *Mesh* convencional e as aplicações puras de *Radio Duty Cycling*. Isso pode ser explicado que apesar do crescimento da rede, tanto a taxa de entrega de pacotes, como a eficiência energética se mantém constantes. Além disso, as redes em *Cluster* são menos afetadas por colisão, logo elas podem sustentar uma taxa de entrega de pacotes estável para todas as densidades de rede, enquanto as redes não clusterizadas sofrem uma rápida degradação na taxa de entrega de pacotes quando o tamanho da rede aumenta.

No entanto, as análises gerais das simulações indicam que a escolha da melhor métrica dependem da necessidade da construção da rede. A Tabela 2 mostra as melhores relações de custo benefício vistas nas simulações. Elas demonstram que o Bluetooth *Mesh* para redes pequenas (16 nós) e médias (144 nós) tem uma melhor taxa de entrega de pacotes. Porém essa taxa se torna melhor com o método de Clusterização puro sem (*Duty-Cycle*) para redes grandes (acima de 288 nós).

**Tabela 2 – Ganhos do processo.**

Tamanho da Rede	Taxa de Entrega de Pacotes	Eficiência Energética
16 nós	BLE Mesh	Duty-Cycle 25%
144 nós	BLE Mesh	Clusterização 50%
Acima de 288 nós	Clusterização 100%	Clusterização 50%

**Fonte: Autoria própria**

Já os ganhos energéticos são vistos com aplicação da técnica de *Radio Duty Cycling* para redes pequenas (16 nós). No entanto, para redes médias e grandes, o melhor rendimento é obtido com o método de Clusterização aliado a técnica de *Radio Duty Cycling*. Tanto para redes médias (144 nós), como para redes grandes com mais de 288 nós, os nós de cada *Cluster* devem conter um *Duty-Cycle* de 50% para se obter uma melhor eficiência energética.

Como conclusão, pode ser observado que o proposto algoritmo de clusterização, quando aplicado com o método de *Radio Duty Cycling* podem melhorar em taxa de entrega de pacotes, em eficiência energética e escalabilidade se comparado com o Bluetooth original, porém a escolha da melhor configuração dependem da prioridade e necessidade da rede.

## 5.2 TRABALHOS FUTUROS

Um dos primeiros passos desse trabalho foi a construção do simulador com objetivo de gerar os resultados necessários para a validação do protocolo Bluetooth *Mesh*. Nesse desenvolvimento, algumas considerações iniciais foram tomadas como constantes, mas que precisam ser desenvolvidas em futuras melhorias no simulador. São elas, preenchimento inicial do TTL utilizando o processo de *Heartbeat*, utilização do processo de estabelecimento de associação dos nós (*Friendship*), construção inicial dos grupos de clusterização e melhoria no processo de associação de um nó no *Cluster*.

O processo de estabelecimento de uma *Friendship* entre os nós LPNs e RF's é um

dos primeiros pontos de melhoria do simulador. Pois nesse trabalho toda rede foi gerada inicialmente de forma randômica. Onde os nós LPNs eram inicializados com a informação do seu RF e conseqüentemente, o LPN constava na lista de endereços LPNs do RF. Esse processo pode ser utilizado não apenas para realizar a associação entre um nó RF e um LPN, mas também para a formação de um *Cluster*.

Outra melhoria deve ser aplicada na construção inicial do *Cluster*, pois nesse trabalho, cada *Cluster* e cada RFCH foram gerados de forma randômica. E durante o processo, após atingir um nível crítico de energia, o RFCH identifica um novo CH em sua tabela energética e envia um *Poll Response* para os nós LPNs para ocorrer a reconfiguração do *Cluster*. Esse processo se repete até o primeiro nó esvaziar sua bateria. Porém um cenário mais realista deve considerar que após se tornar LPN, o nó deve através do processo de *Friendship* encontrar um novo *Cluster* e conseqüentemente um novo RFCH. Esse processo pode otimizar o processo de formação de um novo *Cluster*, já que no processo atual, caso um nó LPN não receber um pacote de mudança de RFCH, ele pode ficar sem estabelecimento de conexão de maneira indefinida.

Outros pontos de melhoria correspondem ao estudo e implementação de protocolos que podem ser utilizados com o objetivo de otimizar a troca de mensagens pelo protocolo Bluetooth *Mesh*. Nesse trabalho foram utilizados os métodos de clusterização e as métricas de *Radio Duty Cycling*. Porém, trabalhos como *Trickle*, *Light-weight mobile routing (LMR)*, entre outros devem ser estudados. Outro aprimoramento que pode ser adicionado ao simulador seria considerar o efeito de captura, onde as transmissões concorrentes são consideradas como interferência que só configura um evento de outage caso atingir um limiar pré-estabelecido.

Além disso, melhorias no software quanto a geração de perda de pacotes devem ser implementadas para deixar o simulador mais realista. O trabalho atual usou o modelo baseado em UGDM do *Cooja*, o qual utiliza apenas a perda de percurso. No entanto, deve-se utilizar de métodos mais realistas para a avaliação do canal e identificação de perdas como por exemplo *Fading*, a qual alterna a atenuação do sinal por muitas variáveis de forma randômica como tempo, posição, rádio frequência. Nesse contexto, o modelo estatístico de *Rayleigh* e a distribuição de *Nakagami* devem ser adicionados. Enquanto o primeiro, é utilizado quando não há linha de visada entre o transmissor e o receptor, *Nakagami-m* é um modelo ajustável, onde é equivalente a Rayleigh quando  $m=1$ .

Por fim, esse trabalho sugere que o protocolo Bluetooth *Mesh* da IETF seja examinado para verificar possíveis ganhos ou perdas em eficiência energética ou taxa de

entrega de pacotes em relação ao original lançado pela SIG estudado nesse trabalho.

## REFERÊNCIAS

- ALEXANDER, R.; BRANDT, A.; VASSEUR, J.; HUI, J.; PISTER, K.; THUBERT, P.; LEVIS, P.; STRUIK, R.; KELSEY, R.; WINTER, T. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. **IETF**, RFC Editor, n. 6550, mar. 2012. Disponível em: <<https://rfc-editor.org/rfc/rfc6550.txt>>. Acesso em: 03 mar. 2018.
- BLUETOOTH SIG. **Mesh Profile**: Bluetooth specification. 2017. Disponível em: <<https://www.bluetooth.com/specifications/mesh-specifications/>>. Acesso em: 29 dez. 2018.
- BLUETOOTH SIG. **The history of the Bluetooth SIG**. 2018. Disponível em: <<https://www.bluetooth.com/about-us/our-history>>. Acesso em: 02 mar. 2019.
- BURATTI, C.; GIORGETTI, A.; VERDONE, R. Cross-layer design of an energy-efficient cluster formation algorithm with carrier-sensing multiple access for wireless sensor networks. **EURASIP Journal on Wireless Communications and Networking**, v. 2005, jan. 2005.
- COLUMBUS, L. **2018 Roundup Of Internet Of Things Forecasts And Market Estimates**. 2018. Disponível em: <<https://www.forbes.com/sites/louiscolumbus/2018/12/13/2018-roundup-of-internet-of-things-forecasts-and-market-estimates/1568d7327d83>>. Acesso em: 13 fev. 2019.
- DARROUDI, S.; GOMEZ, C. Bluetooth low energy mesh networks: A survey. **Sensors**, v. 17, jun. 2017.
- DEPEDRI, A.; ZANELLA, A.; VERDONE, R. An energy efficient protocol for wireless sensor networks. In: **Autonomous Intelligent Networks and Systems (AINS 2003), Menlo Park, CA**. 2003.
- DI MARCO, P.; SKILLERMARK, P.; LARMO, A.; ARVIDSON, P. **Bluetooth Mesh Networking**. Estocolmo, Suécia, 2017. Disponível em: <<https://www.ericsson.com/en/publications/white-papers/bluetooth-mesh-networking>>. Acesso em: 13 fev. 2020.
- DUNKELS, A. The contikimac radio duty cycling protocol. **SICS Technical Report T2011:13**, Dez. 2011. Disponível em: <<http://dunkels.com/adam/dunkels11contikimac.pdf>>. Acesso em: 11 fev. 2018.
- ERICSSON. **Internet of Things forecast ? Ericsson Mobility Report**. Estocolmo, Suécia, 2019. Disponível em: <<https://www.ericsson.com/49d1d9/assets/local/mobility-report/documents/2019/ericsson-mobility-report-june-2019.pdf>>. Acesso em: 13 jul. 2019.
- GOGIC, A.; MUJCIC, A.; HODZIC, S.; SULJANOVIC, N. Performance analysis of bluetooth low energy mesh routing algorithm in case of disaster prediction. **World**

Academy of Science, Engineering and Technology, **International Journal of Computer, Electrical, Automation, Control and Information Engineering**, v. 10, p. 929–935, jan. 2016.

GOTZ, J. D.; MORITZ, G.; KERUSAUSKAS RAYEL, O. Improving bluetooth mesh energy efficiency using clustering. **Journal of Communication and Information Systems**, 2020.

HEINZELMAN, W. R.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. In: **Proceedings of the 33rd Annual Hawaii International Conference on System Sciences**. 2000. p. 10 pp. vol.2–. Disponível em: <<https://ieeexplore.ieee.org/document/926982>>. Acesso em: 05 mar. 2018.

IDC CORPORATE USA. **IDC Forecasts Worldwide Technology Spending on the Internet of Things to Reach 1.2 Trillion in 2022**. Framingham, Massachusetts, USA, 2019. Disponível em: <<https://www.idc.com/getdoc.jsp?containerId=prUS43994118>>. Acesso em: 13 fev. 2019.

IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4.: Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPANs). **IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)**, p. 1–320, set. 2006. Disponível em: <<https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=11161>>. Acesso em: 13 mar. 2019.

IEEE Standard for Information technology–Telecommunications and information exchange between systems - Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation. **IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)**, p. 1–594, maio 2017. Disponível em: <<https://ieeexplore.ieee.org/document/7920364>>. Acesso em: 22 jul. 2019.

IEEE Standard for Local and metropolitan area networks–Part 15.4.: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer. **IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)**, p. 1–225, abr. 2012. Disponível em: <<https://ieeexplore.ieee.org/servlet/opac?punumber=6185523>>. Acesso em: 15 mar. 2019.

KYASANUR, P.; CHOUDHURY, R. R.; GUPTA, I. Smart gossip: An adaptive gossip-based broadcasting service for sensor networks. In: **2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems**. 2006. p. 91–100. ISSN 2155-6806. Disponível em: <<https://ieeexplore.ieee.org/document/4054017>>. Acesso em: 28 out. 2018.

LAYA, A.; KALALAS, C.; VAZQUEZ-GALLEGO, F.; ALONSO, L.; ALONSO-ZARATE, J. Goodbye, aloha! **IEEE Access**, v. 4, p. 2029–2044, Abr. 2016. ISSN 2169-3536. Disponível em: <<https://ieeexplore.ieee.org/document/7457611>>. Acesso em: 17 mar. 2018.

LEE, T.; LEE, M.; KIM, H.; BAHK, S. A synergistic architecture for rpl over ble. In: **2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)**. 2016. p. 1–9.

LEVIS, P.; PATEL, N.; CULLER, D.; SHENKER, S. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In: **Proc. of the First USENIX Conference on Networked Systems Design and Implementation (NSDI)**. Berkeley, CA, USA: USENIX Association, 2004. (NSDI'04), p. 2–2. Disponível em: <<http://dl.acm.org/citation.cfm?id=1251175.1251177>>.

M-WAY SOLUTIONS GMBH. **Fruity mesh**. 2018. Disponível em: <<https://github.com/mwaylabs/fruitymesh/wiki>>. Acesso em: 01 mar. 2019.

MICROCHIP TECHNOLOGY INC. **Bluetooth Low Energy Physical Layer**. 2019. Disponível em: <<http://microchipdeveloper.com/wireless:ble-phy-layer>>. Acesso em: 03 mar. 2019.

MURILLO, Y.; REYNDERS, B.; CHIUMENTO, A.; MALIK, S.; CROMBEZ, P.; POLLIN, S. Bluetooth now or low energy: Should BLE mesh become a flooding or connection oriented network? In: **2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)**. 2017. p. 1–6. ISSN 2166-9589. Disponível em: <<https://ieeexplore.ieee.org/document/8292705>>. Acesso em: 01 jun. 2018.

NORDIC SEMICONDUCTOR. **Bluetooth Low Energy and 2.4 GHz SoC**. 2019. Disponível em: <<https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF51822>>. Acesso em: 13 out. 2019.

OSTERLIND, F.; DUNKELS, A.; ERIKSSON, J.; FINNE, N.; VOIGT, T. Cross-level sensor network simulation with cooja. In: **Proceedings. 2006 31st IEEE Conference on Local Computer Networks**. 2006. p. 641–648. Disponível em: <<https://ieeexplore.ieee.org/document/4116633>>. Acesso em: 18 dez. 2018.

SIGFOX. **SigFox - the global communications service provider for the internet of things (IoT)**. 2019. Disponível em: <<https://www.sigfox.com/en>>. Acesso em: 05 ago. 2019.

SILICON LABS. **AN1137: Bluetooth® Mesh Network Performance**. Austin, Texas, USA, 2018. Disponível em: <<https://www.silabs.com/documents/public/application-notes/an1137-bluetooth-mesh-network-performance.pdf>>. Acesso em: 13 fev. 2019.

SILICON LABS INC. **AN1142: Mesh Network Performance Comparison**. 1 2018. Rev. 0.2. Disponível em: <<https://www.silabs.com/products/wireless/learning-center/mesh-performance>>. Acesso em: 28 fev. 2020.

SIMPY. **Simpy**: Discrete event simulation for python. 2002. Disponível em: <<https://simpy.readthedocs.io/en/latest/>>. Acesso em: 06 abr. 2019.

SINGH, S. K.; KUMAR, P.; SINGH, J. P. A survey on successors of leach protocol. **IEEE Access**, v. 5, p. 4298–4328, fev. 2017. ISSN 2169-3536.

SIRUR, S.; JUTURU, P.; GUPTA, H. P.; SERIKAR, P. R.; REDDY, Y. K.; BARAK, S.; KIM, B. A mesh network for mobile devices using bluetooth low energy. In: **2015 IEEE SENSORS**. 2015. p. 1–4. Disponível em: <<https://ieeexplore.ieee.org/document/7370451>>. Acesso em: 03 jun. 2018.

SORNIN, N.; YEGIN, A. **LoRaWAN® Specification v1.1: LoRa Alliance?** Beaverton, Oregon, USA, jan. 2017. Disponível em: <<https://lora-alliance.org/>>. Acesso em: 02 ago. 2019.

STANKOVIC, J. A. Research directions for the internet of things. **IEEE Internet of Things Journal**, v. 1, n. 1, p. 3–9, fev. 2014. ISSN 2327-4662. Disponível em: <<https://ieeexplore.ieee.org/document/6774858>>. Acesso em: 14 fev. 2019.

SUN, Y.; GUREWITZ, O.; DU, S.; TANG, L.; JOHNSON, D. B. Adb: An efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks. In: **Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys 2009**. 2009. p. 43–56.

THREAD. **Thread Group**. 2019. Disponível em: <<https://www.threadgroup.org/>>. Acesso em: 13 out. 2019.

TIAN, L. A multi-flooding data transmission mechanism for low-duty-cycle wireless sensor network. In: **The 26th Chinese Control and Decision Conference (2014 CCDC)**. 2014. p. 2181–2184. ISSN 1948-9439. Disponível em: <<https://ieeexplore.ieee.org/document/6852529>>. Acesso em: 07 abr. 2018.

VILAJOSANA, X.; WANG, Q.; CHRAIM, F.; WATTEYNE, T.; CHANG, T.; PISTER, K. S. J. A realistic energy consumption model for tsch networks. **IEEE Sensors Journal**, v. 14, n. 2, p. 482–489, fev. 2014. ISSN 1530-437X. Disponível em: <<https://ieeexplore.ieee.org/document/6627960>>. Acesso em: 20 mar. 2019.

YASSEIN, M. B.; AL-ZOU'BI, A.; KHAMAYSEH, Y.; MARDINI, W. Improvement on leach protocol of wireless sensor network (vleach). **Int. J. Digit. Content Technol. Appl.** **2009**, v. 3, p. 132–136, jan. 2009.

YEKTAPARAST, A.; NABAVI, F.; SARMAST, A. An improvement on leach protocol (cell-leach). In: **2012 14th International Conference on Advanced Communication Technology (ICACT)**. 2012. p. 992–996. ISSN 1738-9445. Disponível em: <<https://ieeexplore.ieee.org/document/6174834?denied=>>. Acesso em: 11 abr. 2018.

ZHANG, H.; JIANG, Z. Performance analysis of broadcasting schemes in mobile ad hoc networks. **IEEE Communications Letters**, v. 8, n. 12, p. 718–720, dez. 2004. ISSN 1089-7798. Disponível em: <<https://ieeexplore.ieee.org/document/1369236>>. Acesso em: 18 nov. 2018.

ZHOU, M.; ZHANG, S.; ZHONG, X. A novel dynamic priority round robin polling scheme for bluetooth piconet. In: **2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)**. 2017. p. 895–898. ISSN 2327-0594. Disponível em: <<https://ieeexplore.ieee.org/document/8343054>>. Acesso em: 12 mar. 2018.

ZIGBEE ALLIANCE. **ZigBee specifications**. 2007. Disponível em: <<https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf>>. Acesso em: 27 jul. 2019.