

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**ANÁLISE COMPARATIVA DA VAZÃO E DO CONSUMO
DE ENERGIA EM REDES DE SENSORES SEM FIO COM
TOPOLOGIA ESTRELA UTILIZANDO O PADRÃO IEEE
802.15.4 NO MODO *BEACON-ENABLED* E *NONBEACON***

LUÍS FELIPE MENESES NERES DE BRITO

ORIENTADOR: GEORGES DANIEL AMVAME NZE

**MONOGRAFIA APRESENTADA COMO REQUISITO PARCIAL
PARA CONCLUSÃO DO CURSO DE ENGENHARIA DE REDES DE
COMUNICAÇÃO**

BRASÍLIA/DF: DEZEMBRO – 2009

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**ANÁLISE COMPARATIVA DA VAZÃO E DO CONSUMO DE
ENERGIA EM REDES DE SENSORES SEM FIO COM TOPOLOGIA
ESTRELA UTILIZANDO O PADRÃO IEEE 802.15.4 NO MODO
*BEACON-ENABLED E NONBEACON***

LUÍS FELIPE MENESES NERES DE BRITO

MONOGRAFIA SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA
FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS
REQUISITOS PARA A OBTENÇÃO DE GRAU DE ENGENHEIRO DE REDES DE COMUNICAÇÃO.

APROVADA POR:

GEORGES AMVAME NZE, DOUTOR, UnB
(ORIENTADOR)

LAERTE PEOTTA DE MELO, MESTRE, UnB
(EXAMINADOR INTERNO)

BRASÍLIA/DF, 16 DE DEZEMBRO DE 2009

FICHA CATALOGRÁFICA

BRITO, LUÍS FELIPE MENESES NERES DE

Análise comparativa da taxa da vazão e do consumo de energia em redes de sensores sem fio com topologia estrela em modos *Beacon-enabled* e *nonbeacon*.

xiv, 74p., 210 x 297 mm

Monografia (Graduação) – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. Redes de Sensores

2. IEEE 802.15.4

3. Zigbee

4. NS2

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

BRITO, L. F. M N. (2009). Análise comparativa da vazão e do consumo de energia em redes de sensores sem fio em topologia estrela. Monografia de Graduação, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 67p.

CESSÃO DE DIREITOS

AUTOR: Luís Felipe Meneses Neres de Brito.

TÍTULO: Análise comparativa da vazão e do consumo de energia em redes de sensores sem fio com topologia estrela do padrão IEEE 802.15.4 no modo *beacon-enabled* e *nonbeacon*

GRAU: Engenheiro de Redes de Comunicação

ANO: 2009

É concedida à Universidade de Brasília permissão para reproduzir cópias desta monografia e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa monografia pode ser reproduzida sem autorização por escrito do autor.

Universidade de Brasília

Campus Universitário Darcy Ribeiro – Asa Norte

CEP 70.910.900 Brasília/DF – Brasil

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus, pela saúde e força que me permitiram cumprir mais uma etapa da minha vida. Em seguida, um agradecimento especial aos meus pais, Pedro e Bárbara, por confiarem em mim e aceitarem as minhas escolhas sem nunca medir esforços para me ajudar.

Gostaria de agradecer também aos meus irmãos, Isabella e Pedro, pela sua amizade e compreensão. À minha avó, pelo exemplo de dedicação e bondade. Aos meus primos, em especial ao Marcos André, pela sua generosidade e à Ana Lina e Pedro Henrique que, mesmo distantes, estiveram presentes em todos os momentos. Agradeço também a todos os meus Tios, especialmente as tias Deusa e Daura, por sempre me recordarem do quão bom é estar em casa.

Agradeço também aos meus amigos, presentes em todos os momentos, bons ou ruins. Em especial, gostaria de agradecer a Cecília, Danilo, Vanilda, Michell e Wright. Mesmo sem saber eles possibilitaram que eu não desistisse nunca. A todos os professores da UnB que me ajudaram a ter uma base firme para o desenvolvimento deste projeto. A Ingrid pelos conselhos iniciais que me ajudaram a definir os rumos deste trabalho. A Beatriz Santana que foi imprescindível para o término deste projeto e cuja ajuda foi imensurável e eu serei eternamente grato. Ao professor Georges, pela sua compreensão, comprometimento e conselhos durante toda a elaboração deste projeto.

Enfim, a todos que, direta ou indiretamente, contribuíram para a minha formação acadêmica e para a conclusão deste trabalho.

RESUMO

O grande avanço tecnológico nas áreas das comunicações sem fio e na microeletrônica permitiu o desenvolvimento de redes formadas por sensores. Este tipo de rede foi inicialmente utilizado em aplicações militares. Porém, com o aumento da demanda por aplicações de controle e automação, as redes de sensores sem fio passaram a ser utilizadas nos mais diversos meios. A especificação Zigbee foi criada para definir um padrão completo, aberto e confiável para atender grandes redes de sensores com baixo consumo de energia, baixa taxa de dados e por um bom custo-benefício.

Este trabalho tem como objetivo apresentar um estudo abrangente sobre as redes de sensores sem fio, demonstrando suas características, aplicações, bem como, por meio de simulações, avaliar o desempenho das redes Zigbee, levando em consideração parâmetros importantes como a vazão e o consumo de energia.

ABSTRACT

The large technological progress in areas such as wireless communications and microelectronics over the last decade has enabled the development of Wireless Sensor Networks (WSN). This type of network was first developed for use in military applications. However, with the increasing demand of technologies for control and automation purposes, WSN began to be used in a diverse range of operations. The Zigbee specification was design to define a pattern, open and reliable, to make possible the use of large, low-power, low data rate, cost-efficient sensor networks.

This paper aims to present a comprehensive study on WSN, showing its characteristics, applications and, through simulations, evaluate the Zigbee's network performance, taking into account important parameters such as throughput and power consumption.

LISTA DE FIGURAS

| | |
|-------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 2.1: Redes infra-estruturadas e MANET, respectivamente [13]..... | 18 |
| Figura 2.2: Exemplos de possíveis aplicações para as redes de sensores [29]..... | 20 |
| Figura 2.3: Dispositivos ou nós sensores distribuídos em uma rede doméstica [5]..... | 21 |
| Figura 2.4: Redes Zigbee usada para monitoramento remoto de paciente [5]..... | 22 |
| Figura 2.5: Exemplos de nós sensores [6]..... | 25 |
| Figura 3.1: Zigbee™ Alliance [5]..... | 27 |
| Figura 3.2: Comparação de Tecnologias de redes sem fio[6]..... | 28 |
| Figura 3.3: Modelo de referência ISO de sete camadas | 29 |
| Figura 3.4: Camadas de protocolo do padrão Zigbee [5]..... | 30 |
| Figura 3.5: Nomenclaturas para as funções dos dispositivos nos padrões Zigbee e IEEE 802.15.4 [5]..... | 33 |
| Figura 3.6: Rede com Topologia Estrela [5]..... | 34 |
| Figura 3.7: Rede com topologia em malha [5]..... | 34 |
| Figura 3.8: Rede com topologia em árvore [5]..... | 35 |
| Figura 3.9 (a): Transferência de dados do dispositivo para o coordenador em redes com utilização de <i>beacons</i> . [5]..... | 38 |
| Figura 3.9 (b): Transferência de dados do dispositivo para o coordenador em redes sem <i>beacons</i> . [5]..... | 39 |
| Figura 3.10: Transferência de dados do coordenador para o dispositivo em redes com <i>beacons</i> . [5]..... | 39 |
| Figura 3.11: Transferência de dados dos dispositivos para o coordenador em redes com <i>beacons</i> . [5]..... | 40 |
| Figura 3.12: Camadas de protocolo do padrão Zigbee [5]..... | 42 |
| Figura 3.13: Estrutura de pacotes de redes Zigbee [5]..... | 44 |
| Figura 3.14: Estrutura do MAC <i>beacon frame</i> [5]..... | 45 |
| Figura 3.15: Estrutura do <i>Superframe</i> [5]..... | 46 |
| Figura 3.16: Estrutura do MAC <i>data frame</i> [5]..... | 47 |
| Figura 3.17: Estrutura do MAC <i>Acknowledgment frame</i> [5]..... | 48 |
| Figura 3.18: Estrutura do MAC <i>command frame</i> [5]..... | 48 |
| Figura 4.1: Topologia da Rede..... | 52 |
| Figura 4.2: Descarga de corrente constante [28]..... | 55 |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 4.3: Variação do Throughput com o aumento da quantidade de nós ativos para a rede sem beacons..... | 57 |
| Figura 4.4: Variação do consumo médio de energia com o aumento da quantidade de nós ativos para a rede sem beacons. | 58 |
| Figura 4.5: Variação do Throughput com o aumento da quantidade de nós ativos para a rede com $B0=S0=1$ | 59 |
| Figura 4.6: Variação do consumo médio de energia com o aumento da quantidade de nós ativos para a rede com $B0=S0=1$ | 59 |
| Figura 4.7: Variação do Throughput com o aumento da quantidade de nós ativos para a rede com $B0=S0=3$ | 60 |
| Figura 4.8: Variação do consumo médio de energia com o aumento da quantidade de nós ativos para a rede com $B0=S0=3$ | 60 |
| Figura 4.9: Variação do Throughput com o aumento da quantidade de nós ativos para a rede com $B0=S0=5$ | 61 |
| Figura 4.10: Variação do consumo médio de energia com o aumento da quantidade de nós ativos para a rede com $B0=S0=5$ | 62 |
| Figura 4.11: Comparação da variação do Throughput para a rede operando nos Modo Beacon com diferentes tamanhos de superframe e sem Beacon..... | 63 |
| Figura 4.12: Comparação da variação da taxa de entrega para a rede operando nos Modo Beacon com diferentes tamanhos de superframe e sem Beacon..... | 64 |
| Figura 4.13: Comparação do consumo de energia médio para a rede operando nos Modo Beacon com diferentes tamanhos de superframe e sem Beacon..... | 65 |

LISTA DE TABELAS

| | |
|---------------------------------------------------------------------------------------|----|
| TABELA 3.1 – TAXA DE DADOS E FREQUÊNCIAS DE OPEREÇÃO DO PADRÃO IEEE 802.15.4 [5]..... | 31 |
| TABELA 4.1: PARÂMETROS DE TRÁFEGO E NÓS..... | 51 |
| TABELA 4.2: PARÂMETROS FÍSICOS..... | 53 |
| TABELA 4.3: PARÂMETROS DE ROTEAMENTO..... | 53 |
| TABELA 4.4: PARÂMETROS DE POTÊNCIA E ENERGIA..... | 54 |

LISTA DE ACRÔNIMOS

| | |
|----------------|---------------------------------------------------------------|
| ACK | <i>Acknowledge</i> |
| AODV | <i>Ad-Hoc On-Demand Distance Vector</i> |
| APL | <i>Application</i> |
| APS | <i>Subcamada de Suporte a Aplicação</i> |
| ASK | <i>Amplitude Shift Keying</i> |
| BI | <i>Beacon Interval</i> |
| BO | <i>Beacon Order</i> |
| BPSK | <i>Binary Phase Shift Keying</i> |
| CAP | <i>Contention Access Period</i> |
| CCA | <i>Clear Channel Assessment</i> |
| CFP | <i>Contention Free Period</i> |
| CSMA/CA | <i>Carrier Sense Multiple Access with Collision Avoidance</i> |
| DD | <i>Designated Device</i> |
| DSSS | <i>Direct Sequence Spread Spectrum</i> |
| ED | <i>Energy Detection</i> |
| ERB | <i>Estação Rádio Base</i> |
| FCS | <i>Frame Check Sequence</i> |
| FFD | <i>Full Function Device</i> |
| GTS | <i>Guaranteed Time Slot</i> |
| IEEE | <i>Institute of Electrical and Electronic Engineers</i> |
| ISO | <i>International Organization for Standardization</i> |
| LQI | <i>Link Quality Indicator</i> |
| LSB | <i>Least Significant Bit</i> |
| LR-WPAN | <i>Low Rate Wireless Personal Area Network</i> |
| HR-WPAN | <i>High Rate Wireless Personal Area Network</i> |
| MR-WPAN | <i>Medium Rate Wireless Personal Area Network</i> |
| MAC | <i>Media Access Control</i> |
| MANET | <i>Mobile Ad Hoc Network</i> |
| MCPS | <i>MAC Common Part Layer</i> |
| MFR | <i>MAC Footer</i> |
| MHR | <i>MAC Header</i> |

| | |
|---------------|---------------------------------------------|
| MLME | <i>MAC Sublayer Management Entity</i> |
| MPDU | <i>MAC Protocol Data Unit</i> |
| MSDU | <i>MAC Service Data Unit</i> |
| NS2 | <i>Network Simulator 2</i> |
| NWK | <i>Network</i> |
| O-QPSK | <i>Offset Quadrature Phase Shift Keying</i> |
| OSI | <i>Open Systems Interconnection</i> |
| PAN | <i>Personal Area Network</i> |
| PHR | <i>PHY Header</i> |
| PHY | <i>Physical</i> |
| PPDU | <i>Physical Protocol Data Unit</i> |
| PSDU | <i>Physical Service Data Unit</i> |
| PSSS | <i>Parallel Sequence Spread Spectrum</i> |
| RF | <i>Radio Frequency</i> |
| RFD | <i>Reduced Function Device</i> |
| RSSF | <i>Redes de Sensores Sem Fio</i> |
| SD | <i>Superframe Duration</i> |
| SHR | <i>Synchronization Header</i> |
| SO | <i>Superframe Order</i> |
| WSN | <i>Wireless Sensor Network</i> |
| ZC | <i>ZigBee Coordinator</i> |
| ZED | <i>ZigBee End Device</i> |
| ZR | <i>ZigBee Router</i> |

SUMÁRIO

| | |
|----------------------------------------------------|----|
| 1. INTRODUÇÃO | 15 |
| 1.1. OBJETIVOS | 16 |
| 1.2. METODOLOGIA | 16 |
| 1.3. ORGANIZAÇÃO DO TEXTO | 16 |
| 2. REDES DE SENSORES SEM FIO | 18 |
| 2.1. APLICAÇÃO DAS REDES DE SENSORES SEM FIO | 19 |
| 2.1.2. Aplicações Militares..... | 20 |
| 2.1.2. Automação Residencial | 20 |
| 2.1.3. Aplicações Ambientais | 21 |
| 2.1.4. Aplicações em Sistemas de Saúde | 22 |
| 2.1.5. Automação Industrial | 22 |
| 2.2. CARACTERÍSTICAS DAS REDES DE SENSORES | 23 |
| 2.2.1. Endereçamento dos nós sensores | 23 |
| 2.2.2. Agregação de dados | 23 |
| 2.2.3. Mobilidade dos sensores | 23 |
| 2.2.4. Restrição dos dados coletados | 23 |
| 2.2.5. Quantidade de Sensores | 23 |
| 2.2.6. Limitação de Energia | 23 |
| 2.2.7. Auto Organização | 24 |
| 2.3. COMPONENTES DE UMA REDE DE SENSORES | 24 |
| 2.3.1. Bateria | 25 |
| 2.3.2. Rádio | 25 |
| 2.3.3. Unidade de Armazenamento..... | 26 |
| 2.3.4. Unidade de Processamento | 26 |
| 2.3.5. Sensor..... | 26 |

| | |
|---------------------------------------------------------------|----|
| 3. ZIGBEE | 27 |
| 3.1. CLASSIFICAÇÃO DAS REDES DE SENSORES SEM FIO | 28 |
| 3.2. RELAÇÃO ENTRE OS PADRÕES ZIGBEE E IEEE 802.15.4 | 29 |
| 3.3. CARACTERÍSTICAS DAS REDES ZIGBEE | 31 |
| 3.3.1. Frequência de Operação e Taxa de Dados | 31 |
| 3.3.2. Dispositivos..... | 32 |
| 3.3.3. Topologias..... | 33 |
| 3.4. COMUNICAÇÃO EM REDES ZIGBEE | 36 |
| 3.4.1. CSMA/CA..... | 36 |
| 3.4.2. Presença de <i>Beacons</i> em redes Zigbee..... | 36 |
| 3.4.3. Métodos de transferência de dados | 37 |
| 3.4.3.1 Transferência do dispositivo para o coordenador | 37 |
| 3.4.3.2 Transferência do coordenador para o dispositivo | 39 |
| 3.4.3.3 Transferência de dados entre dois dispositivos..... | 40 |
| 3.5. ENDEREÇAMENTO EM REDES DO PADRÃO IEEE 802.15.4 | 40 |
| 3.6. VERIFICAÇÃO DE DADOS..... | 41 |
| 3.7. PROCESSOS DE ASSOCIAÇÃO E DISSOCIAÇÃO | 41 |
| 3.8. CAMADAS DO PROTOCOLO ZIGBEE..... | 42 |
| 3.8.1. Camada PHY do Padrão IEEE 802.15.4..... | 43 |
| 3.8.2. Camada MAC do Padrão IEEE 802.15.4..... | 44 |
| 3.8.2.1 <i>Beacon Frame</i> | 44 |
| 3.8.2.2 <i>Data Frame</i> | 47 |
| 3.8.2.3 <i>Acknowledgment Frame</i> | 47 |
| 3.8.2.4 <i>Command Frame</i> | 48 |
| 3.8.3. Camada de rede (NWK) do padrão Zigbee..... | 48 |
| 3.8.4. Camada de aplicação (APL) do padrão Zigbee | 49 |
| 4. SIMULAÇÃO..... | 50 |
| 4.1. PARÂMETROS..... | 50 |
| 4.1.1. Parâmetros de Tráfego e nós..... | 51 |

| | | |
|--------|------------------------------------------------------------|----|
| 4.1.2. | Parâmetros Físicos | 52 |
| 4.1.3. | Parâmetros de Roteamento | 53 |
| 4.1.4. | Parâmetros de Potência e Energia..... | 53 |
| 4.2. | CENÁRIOS..... | 55 |
| 4.2.1. | Cenário 1: <i>Nonbeacon Network</i> | 57 |
| 4.2.2. | Cenário 2: <i>Beacon-enabled network</i> com BO=SO=1 | 58 |
| 4.2.3. | Cenário 3: <i>Beacon-enabled network</i> com BO=SO=3..... | 59 |
| 4.2.4. | Cenário 4: <i>Beacon-enabled network</i> com BO=SO=5..... | 61 |
| 4.3. | ANÁLISE E RESULTADOS | 62 |
| 5. | CONCLUSÃO | 66 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 68 |

1. INTRODUÇÃO

As redes de sensores sem fio consistem de um grande número de dispositivos autônomos que são distribuídos espacialmente para monitorar cooperativamente condições físicas ou ambientais de uma determinada área. Esses dispositivos, ou nós sensores, são conectados através de comunicações sem fio e repassam os dados coletados para um nó coordenador ou central que irá analisar as informações recebidas.

A eficiência no gasto de energia é provavelmente o parâmetro mais importante nas redes de sensores sem fio. Muitos sensores são desenvolvidos para serem utilizados em ambientes de difícil acesso em que a troca de dispositivos pode se tornar impraticável. Inclusive, muitos desses sensores já são fabricados com a intenção de serem descartados após a incidência de uma falha ou o esgotamento da energia. As redes de sensores sem fio, portanto, devem ser arquitetadas de forma a estender o tempo de vida dos nós sensores e conseqüentemente de toda a rede. Devido a essas restrições, a evolução das técnicas para evitar o consumo desnecessário de energia, seja em nível de hardware ou software, dita o avanço deste tipo de tecnologia de redes.

As redes de sensores guiadas pelo padrão IEEE 802.15.4 devem operar em um dos dois tipos de topologias definidas pelo protocolo: topologia estrela ou *peer-to-peer* [5]. Nas redes em topologia estrela os dispositivos só podem se comunicar com o nó central ou coordenador, enquanto que nas redes em *peer-to-peer* todos os dispositivos podem estabelecer comunicação entre si. Embora as redes em topologia *peer-to-peer* apresentem muitos caminhos de comunicação alternativos e dessa forma serem redes mais robustas em relação às falhas, elas consomem mais energia do que as redes em topologia estrela. A topologia deve ser escolhida de acordo com a aplicação designada e fatores como escalabilidade, tolerância a falhas e potência consumida.

Inicialmente, as redes de sensores sem fio foram desenvolvidas para aplicações militares, motivadas por questões de segurança, como a vigilância nos campos de batalha, mas hoje são utilizadas nas mais diversas áreas que vão desde a automação doméstica ou industrial até o monitoramento de funções do corpo humano. Apesar da limitação de certos recursos, como energia e memória, a busca cada vez maior por mobilidade e automação demonstram que a tendência é que o número de aplicações usando redes de sensores seja ampliado e diversificado ainda mais [15] [18] [19].

1.1 OBJETIVOS

Entre as aplicações mais difundidas utilizando-se redes de sensores sem fio estão as aplicações de monitoramento e controle. Essas redes, cujos nós sensores muitas vezes encontram-se em locais de difícil acesso, possuem fortes restrições quanto ao gasto de energia. Dessa forma, este trabalho tem como objetivo apresentar um estudo sobre o desempenho de redes de sensores, avaliando parâmetros como a vazão, ou *throughput*, e o consumo de energia, bem como identificando e comparando as técnicas presentes no padrão IEEE 802.15.4 que possibilitam melhorar a eficiência do sistema, citando as vantagens e desvantagens de cada uma delas.

1.2 METODOLOGIA

Para alcançar o objetivo proposto, algumas metodologias foram traçadas. Primeiramente, foi feito um estudo teórico, através da revisão da literatura que aborda o assunto. Foram feitas pesquisas em livros, artigos e trabalhos acadêmicos sobre o tema tratado. Em seguida, realizou-se um estudo do comportamento de redes de sensores através de simulações com o programa NS2, em que vários esquemas foram tratados visando a responder os questionamentos propostos por este trabalho. Após essas etapas e concluídos os objetivos, foi apresentado algumas soluções para melhoria do desempenho dessas redes.

1.3 ORGANIZAÇÃO DO TEXTO

A estrutura do trabalho foi dividida em cinco capítulos. No capítulo 2 foi feita uma revisão sobre as Redes de Sensores Sem Fio (RSSF) e seus componentes. Foram dados exemplos de aplicações e áreas de atuação das redes de sensores bem como suas características e organização. No capítulo 3 é apresentado o padrão Zigbee, bem como sua relação com o padrão IEEE 802.15.4. Foram ilustradas as características desses padrões, seu modo de funcionamento e estrutura bem como foram apresentadas as técnicas que são utilizadas para maximizar a eficiência da rede e controlar as restrições por elas imposta. No capítulo 4, esquemas computacionais para a análise da rede, avaliando a vazão e o consumo de energia em RSSF foram utilizados para simular cenários diversos e através destes fazer um

uma análise comparativa dos resultados encontrados. Finalmente, no capítulo 5 encontra-se a conclusão do trabalho, com considerações a respeito dos resultados obtidos e sugestões para possíveis trabalhos futuros.

2. REDES DE SENSORES SEM FIO

As redes de sensores sem fio (RSSF) consistem de um grande número de dispositivos que são conectados através de comunicações sem fio e que utilizam baixo consumo de energia. Elas são consideradas uma subclasse das redes Ad Hoc. São redes fundamentalmente diferentes de redes como a Internet e geralmente têm aplicações específicas. Dessa forma, o seu desenvolvimento e implementação devem observar parâmetros particulares de acordo com os objetivos os quais a rede se propõe a realizar. Para redes de sensores cujos dispositivos são operados com bateria, a conservação da energia é um fator crucial a ser considerado já que a reposição destas pode ser de difícil acesso ou impossível em certas aplicações [21].

As redes de sensores diferem das redes tradicionais em vários aspectos. Geralmente, essas redes possuem uma grande quantidade de nós, severas restrições de energia e largura de banda e devem possuir mecanismos de auto-configuração e adaptação devido a possíveis falhas nos nós sensores [13].

Do ponto de vista organizacional as RSSF são idênticas às redes MANETs (*Mobile Ad hoc Networks*), pois ambas possuem elementos computacionais que se comunicam diretamente através dos enlaces de comunicação sem fio. Numa rede tradicional, a comunicação entre os elementos computacionais é feita através de estações de rádio-base, como ocorre na Internet. Em redes móveis ad hoc, entretanto, os elementos computacionais trocam dados diretamente entre si, como pode ser observado na Figura 2.1. [13]

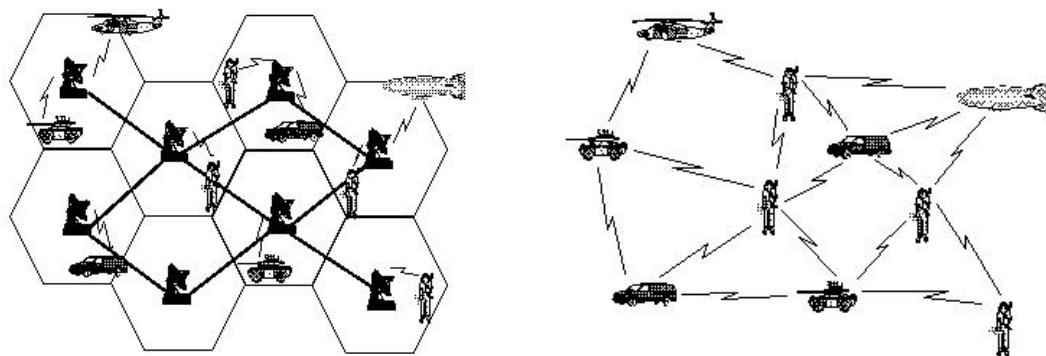


Figura 2.1: Redes infra-estruturadas e MANET, respectivamente. [13]

Segundo a taxonomia de Tikal et al [23] as redes de sensores possuem como principais características o sensor, o observador e o fenômeno. O sensor é o dispositivo que implementa a monitoração física de um fenômeno ambiental e gera relatórios de medidas através de comunicações sem fio. Um sensor deve produzir uma resposta mensurável a mudanças em condições físicas, tais como temperatura, campo magnético e luz. Na maioria dos modelos de sensores, a capacidade de detecção é reduzida com o aumento da distância do sensor ao fenômeno analisado. Ao contrário, a habilidade de detecção é melhorada com o aumento do tempo em que o sensor é exposto à coleta de informação [15].

O observador é o usuário final interessado em obter as informações produzidas pelas redes de sensores em relação a um determinado fenômeno. Ele pode indicar parâmetros e fazer consultas para a rede e receber respostas da mesma. É possível existir múltiplos observadores simultaneamente numa rede de sensores. [15]

O fenômeno é a entidade de interesse do observador, que está sendo monitorada e cuja informação será analisada pela rede de sensores. Múltiplos fenômenos podem ser observados numa rede ao mesmo tempo [15]. Em uma determinada aplicação, o observador está interessado em monitorar o comportamento do fenômeno sob algum requisito de desempenho específico. Numa rede de sensores típica, os sensores individuais apresentam amostras de valores e disseminam a informação, quando necessário, para outros sensores e eventualmente para o observador. As medidas realizadas pelos sensores são amostras discretas do fenômeno físico e estão sujeitas às medidas realizadas pelo sensor individual bem como à localização desse sensor com respeito ao fenômeno.

2.1 APLICAÇÕES DAS REDES DE SENSORES SEM FIO

As redes de sensores são utilizadas em um diverso número de aplicações. Elas podem ser homogêneas ou heterogêneas em relação aos tipos e funcionalidades dos nós sensores [15]. Esta seção exemplifica alguns cenários em que as redes de sensores podem ser aplicadas para aumentar a eficiência e reduzir os custos. A Figura 2.2 resume diversas aplicações em que as redes de sensores podem ser utilizadas.



Figura 2.2: Exemplos de possíveis aplicações para as redes de sensores. [29]

2.1.1 Aplicações Militares

As primeiras redes de sensores foram usadas para aplicações militares. Com dispositivos de baixo custo, a perda ou destruição de sensores por ataques hostis nos campos de batalha não afetavam de forma significativa a operação militar. A robustez, a possibilidade de auto-organização e tolerância a faltas fizeram com que as redes de sensores fossem muito utilizadas para obter informações sobre ameaças, bem como para o monitoramento de armas e munição.

2.1.2 Automação Residencial

Automação residencial é um dos maiores campos de aplicação das redes de sensores sem fio. Normalmente, a taxa de dados exigidos nesse tipo de aplicação não ultrapassa os 10Kbps [5]. Serviços de segurança, medições e mecanismos de controle de luz são exemplos de como as redes de sensores podem ser usadas em um domicílio. Em um sistema de segurança, por exemplo, inúmeros dispositivos com sensores de detecção de movimento e câmeras podem ser utilizados para reportar as informações obtidas para um nó central. A

Figura 2.3 ilustra possíveis usos para os nós sensores em uma rede doméstica. Como pode ser observado na imagem, sensores podem ser aplicados em todos os ambientes de uma residência executando as mais diversas tarefas: regulação da temperatura de ambientes; controle de aparelhos de áudio e vídeo; aquecimento de água; controle da iluminação; controle de portões e garagem, etc.

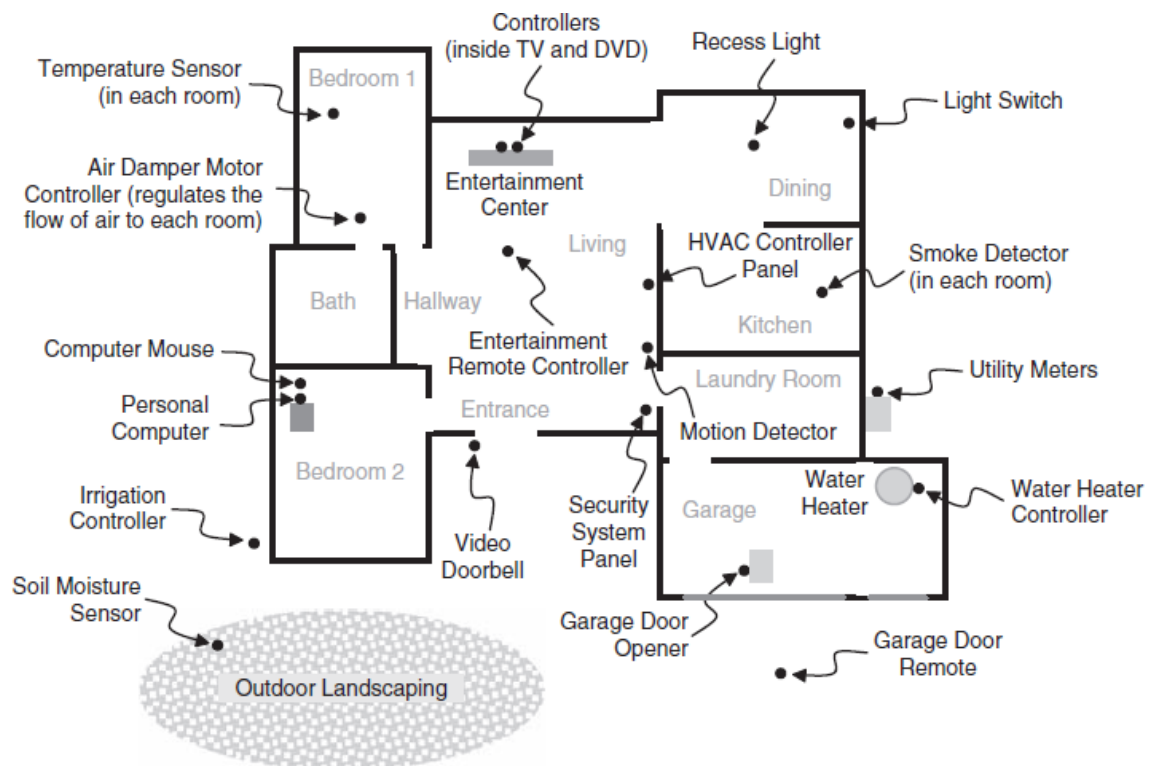


Figura 2.3: Dispositivos ou nós sensores distribuídos em uma rede doméstica [5].

2.1.3 Aplicações Ambientais

Mecanismos para detecção de fogo e inundações estão entre os principais usos das redes de sensores em aplicações ambientais. Inúmeros sensores, integrados por um sistema de rádio frequência e mantido através de baterias ou células solares, podem ser dispostos em uma floresta e durarem por longos períodos. Quando os nós sensores detectam fogo eles enviam informações para o nó central antes que o fogo se espalhe e se torne incontrolável [20]. Um exemplo de uma rede utilizada para detecção de inundações é o sistema ALERT [20], que foi desenvolvido pelo *National Weather Service* nos anos 70 nos Estados Unidos. O sistema ALERT continha sensores que proviam informações em tempo real de tempestades e nível de

água para avaliar a possibilidade de potenciais inundações. Os dados obtidos eram enviados para um nó central para análise de informações.

2.1.4 Aplicações em Sistemas de Saúde

Monitoramento de pacientes é outra aplicação importante das redes de sensores. Sensores são conectados a pacientes que precisam ter informações físicas e biológicas avaliadas constantemente, como a pressão sanguínea ou os batimentos cardíacos. A rede, dessa forma, é usada para coletar dados dos vários sensores anexados ao paciente e transferir essas informações para o médico. A Figura 2.4 ilustra um exemplo do uso de redes de sensores em aplicações utilizadas para monitoramento de paciente. Nesse caso, um paciente veste um dispositivo que é ligado a um sensor. Este irá então coletar informações periodicamente e enviar para um *gateway*, que irá fazer a ligação entre a rede Zigbee e outras redes, como a internet. As informações, então, podem ser enviadas para o computador pessoal do médico.

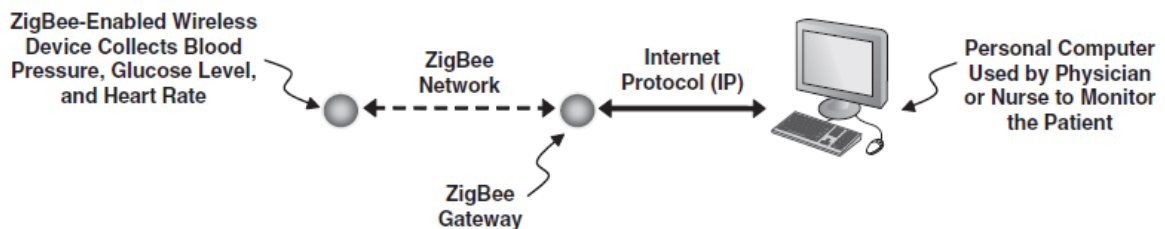


Figura 2.4: Redes Zigbee usada para monitoramento remoto de paciente [5].

2.1.5 Automação Industrial

Em escala industrial, as redes de sensores podem ajudar a coletar informações nas áreas de gerência de energia, controle de luz, controle de estoque, manutenção de equipamentos e muitas outras. Dispondo de tais informações, técnicos podem desenvolver melhores processos para diminuir custos, aumentar a eficiência e conseqüentemente a competitividade.

2.2 CARACTERÍSTICAS DAS REDES DE SENSORES

As redes de sensores sem fio possuem características específicas conforme a área de aplicação em que são utilizadas. Esta seção resume algumas dessas características:

2.2.1 Endereçamento dos nós sensores.

Dependendo do tipo de aplicação, cada sensor pode ser endereçado de forma única ou não. Mais detalhes sobre o endereçamento de redes Zigbee são encontrados na seção 3.5 do próximo capítulo deste trabalho.

2.2.2 Agregação de Dados.

Indica a capacidade de uma rede de sensores sem fio agregar ou sumarizar dados coletados pelos sensores. Se a rede puder realizar essa função, é possível reduzir o número de mensagens transmitidas. [13].

2.2.3 Mobilidade dos Sensores

É a capacidade dos nós sensores poderem se mover ou não em relação ao sistema em que estão coletando dados. Para a maior parte das aplicações, os sensores estarão estáticos em relação ao sistema em que estão monitorando.

2.2.4 Restrições dos dados coletados

Indica se os dados obtidos pelos sensores possuem algum tipo de restrição.

2.2.5 Quantidade de Sensores

Redes de sensores sem fio contendo de 10 a 100 mil nós sensores são previstas para aplicações de monitoramento [13]. A escalabilidade, portanto, é um fator importante em RSSF.

2.2.6 Limitação de Energia

Em muitas aplicações os nós sensores encontram-se em áreas de difícil acesso, o que dificulta o processo de manutenção e reposição desses elementos. Dessa forma, a energia consumida por esses sensores e conseqüentemente o tempo de vida da bateria dos dispositivos são fatores cruciais a serem observados durante a elaboração do projeto de uma RSSF.

2.2.7 Auto-Organização

Sensores de uma RSSF podem ser perdidos devido a falhas, destruição ou perda de energia. Também podem ficar impossibilitados de fazer a comunicação por problemas no canal comunicação sem fio. O oposto também pode ocorrer quando sensores inativos forem ativados ou quando novos sensores passam a fazer parte da rede. Devido a essa característica das redes de sensores sem fio é necessário haver mecanismos de auto-organização para que a rede possa continuar funcionando de forma correta. Essa configuração deve ser feita de forma automática e periódica.

2.3 COMPONENTES DE UMA REDE DE SENSORES

As redes de sensores são formadas por dispositivos computacionais sem fio conhecidos como nós sensores. Estes, por sua vez, são formados por cinco elementos básicos: bateria ou fonte de energia, unidade de armazenamento, transceptor, processador e sensor. A Figura 2.5 ilustra alguns exemplos de nó sensor.

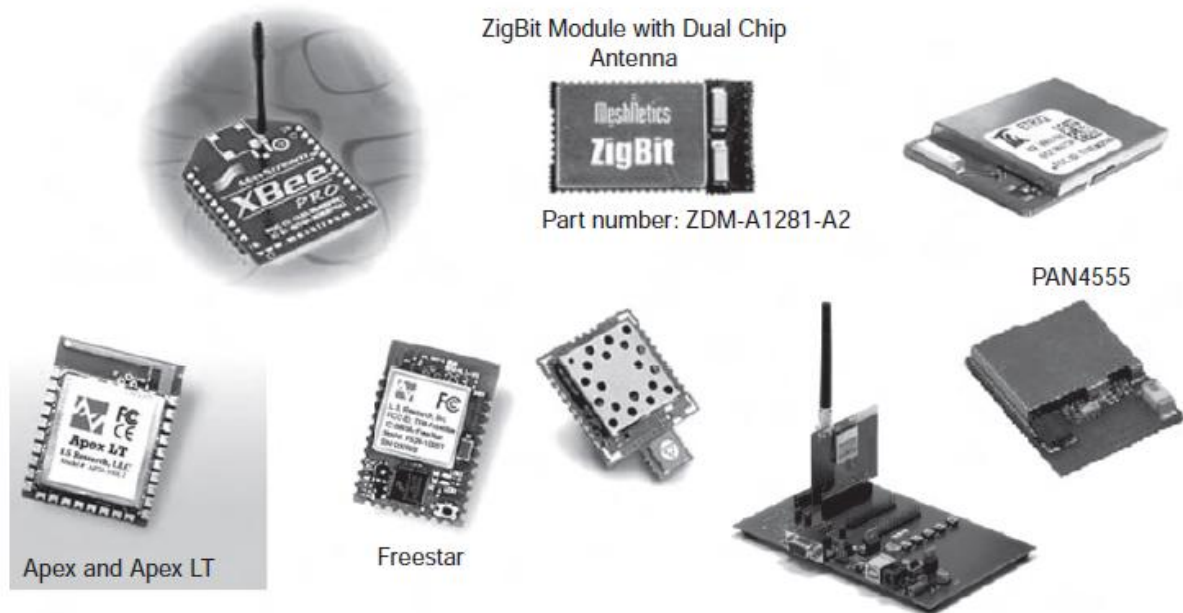


Figura 2.5: Exemplos de nós sensores [6].

2.3.1 Bateria

A Bateria representa a fonte de energia do nó sensor. Ela tem uma capacidade de armazenamento finita e bastante reduzida. O consumo de energia de uma bateria é expresso utilizando-se a métrica unidades de Ampère x Hora (Ah). A bateria é a parte mais sensível de um nó sensor, pois em muitas ocasiões a sua reposição é uma tarefa bastante difícil ou inviável.

2.3.2. Radio

O rádio transceptor representa todo o sistema de transmissão que inclui o transmissor, receptor, amplificador e antena. O transceptor é o elemento computacional responsável pela comunicação nos enlaces sem fio de uma rede de sensores. A comunicação é realizada por sinais de rádio frequência [13].

2.3.3 Unidade de Armazenamento

É o componente responsável pelo armazenamento de dados em um nó sensor, sendo variado de acordo com o protocolo utilizado. A unidade de armazenamento, ou memória, pode possuir capacidade limitada devido a restrições de tamanho do nó sensor.

2.3.4 Unidade de Processamento

A unidade de processamento ou processador é composta por uma memória interna, um micro-controlador e um conversor analógico-digital para receber os sinais advindos dos nós sensores. O processador é um componente de extrema importância e quanto maior a sua capacidade melhor ele será aproveitado pela rede [13].

2.3.5 Sensor

Representa os dispositivos de sensoriamento e monitoramento. O consumo de energia do sensor depende do modo de operação e do tipo de grandeza a ser medida. Em algumas aplicações, por exemplo, os sensores devem acordar apenas em intervalos de tempo regulares. Em situações críticas, por outro lado, os sensores devem permanecer constantemente ligados e nesse caso, obviamente, gastam mais energia. Além das tarefas de monitoramento, o sensor também pode ser utilizado em outras aplicações, o que vai depender do protocolo utilizado na rede [13]. Devido aos efeitos decrescentes do ruído, quanto mais tempo um nó sensor ficar exposto e quanto menor for a sua distância do fenômeno a ser analisado, melhor será a sua contribuição para a rede [13].

3. ZIGBEE

Zigbee é um padrão que define um conjunto de protocolos de comunicações para redes sem fio de curta distância e baixa taxa de dados. [5]. Dispositivos que são baseados em Zigbee podem operar somente nas bandas de 2,4 GHz, 915 MHz e 868 MHz e sua taxa máxima é de apenas 250 Kbits por segundo. O *throughput* típico de redes Zigbee é normalmente igual a 25 Kbits por segundo [6].

O padrão Zigbee foi criado com o objetivo de operar com aplicações que exigissem um baixo consumo de energia, trabalhassem com uma baixa taxa de dados e que tivessem baixo custo. Em muitas das aplicações Zigbee o número total de dispositivos sem fio engajados em uma atividade é bastante limitado. Os dispositivos nessas aplicações gastam a maior parte do tempo em modos de economia de energia, ou *sleep modes*. Dessa forma, dispositivos baseados em Zigbee são capazes de operar por vários anos antes de suas baterias serem repostas. A especificação Zigbee adota a camada física (PHY) e a camada de acesso e controle ao meio (MAC) do padrão IEEE 802.15.4 como seus protocolos. [8]

O padrão Zigbee foi desenvolvido pela Zigbee™ Alliance, um conglomerado que atualmente já é formado por mais de 200 empresas, muitas delas líderes em seu segmento de atuação (vide Figura 3.1). O objetivo do projeto Zigbee era incentivar empresas a produzir produtos de baixo custo para controle e monitoramento de redes sem fio que consumissem pouca energia, fossem eficientes e confiáveis.



Figura 3.1: Zigbee™ Alliance [5]

3.1 CLASSIFICAÇÕES DAS REDES SEM FIO

Redes sem fio de curta distância são divididas em duas categorias principais: *Wireless Local area networks* (WLANs) e *Wireless personal area networks* (WPAN). Esta última categoria é uma extensão das *Wired Local area networks* (LANs) como o padrão IEEE 802.3 (Ethernet). Dispositivos de redes WLAN podem ser integrados a redes LAN e uma vez que esses dispositivos passam a fazer parte da rede LAN, esta passa a tratar os dispositivos da rede sem fio como qualquer outro dispositivo de sua própria rede [5].

As redes WPAN, entretanto, não foram desenvolvidas para repor qualquer outro tipo de redes já existentes. Redes WPAN foram criadas para prover mecanismos de comunicação sem fio que, dentro da sua área operacional, pudessem operar com grande eficiência energética e sem nenhum outro tipo de infra-estrutura auxiliar [5].

As redes WPANs são divididas em três classes: *High-rate* (WR) WPANs, *Medium-rate* (MR) WPANs e *Low-Rate* (LR) WPANs [5]. Redes Zigbee são classificadas como LR-WPAN já que sua máxima taxa de dados atinge 250 Kbits por segundo. Bluetooth, com taxas que variam de 1 a 3 MHz, são redes MR-WPAN. Redes que utilizam o padrão IEEE 802.15.3 são exemplos de redes HR-WPAN. Essas redes podem atingir taxas de dados de até 55Mbits por segundo [4]. A Figura 3.2 faz uma comparação em termos da distância e taxas de dados atingidas por diferentes tecnologias de redes sem fio.

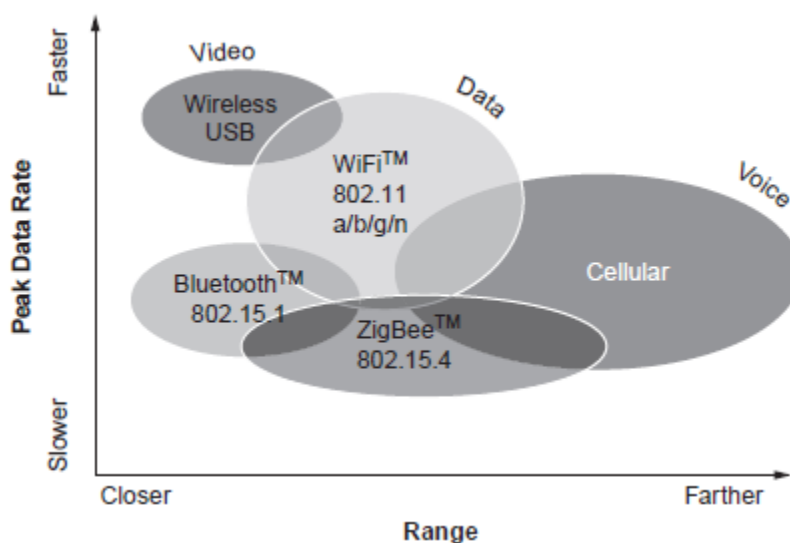


Figura 3.2: Comparação de Tecnologias de redes sem fio[6]

3.2 RELAÇÕES ENTRE OS PADRÕES ZIGBEE E IEEE 802.15.4

Redes de comunicação (com ou sem fio) normalmente são estabelecidas usando o conceito de camadas de protocolo. Cada camada é responsável por determinadas funções da rede e uma camada normalmente só pode passar informações e comandos para as camadas que estão situadas diretamente acima ou abaixo dela.

O modelo OSI (*Open System Interconnection Reference Model*), baseado no trabalho inicial feito pela *International Organization for Standardization* (ISO), foi desenvolvido com o objetivo de criar padrões de conectividade para interligar sistemas de computadores locais e remotos. É um modelo que divide as redes de computadores em sete camadas interdependentes em que cada protocolo programa uma funcionalidade assinalada a uma determinada camada [3]. A Figura 3.3 demonstra o modelo OSI.

| | |
|---|------------------------|
| 7 | Camada de Aplicação |
| 6 | Camada de Apresentação |
| 5 | Camada de Sessão |
| 4 | Camada de Transporte |
| 3 | Camada de Rede |
| 2 | Camada de Enlace |
| 1 | Camada Física |

Figura 3.3: Modelo de referência OSI de sete camadas

A Figura 3.4 ilustra as camadas de protocolo de redes Zigbee. As camadas de protocolo Zigbee são baseadas no modelo OSI. A divisão de protocolos de redes em camadas permite modificar ou repor protocolos que mudaram com o tempo de forma mais eficiente já que não será necessário substituir todo o modelo do protocolo e sim apenas as camadas afetadas. Outra vantagem da divisão de protocolos em camadas é o fato de que, ao

desenvolver uma aplicação, apenas mudanças na camada de aplicação são suficientes já que as camadas inferiores são independentes e podem ser desenvolvidas por terceiros [5].

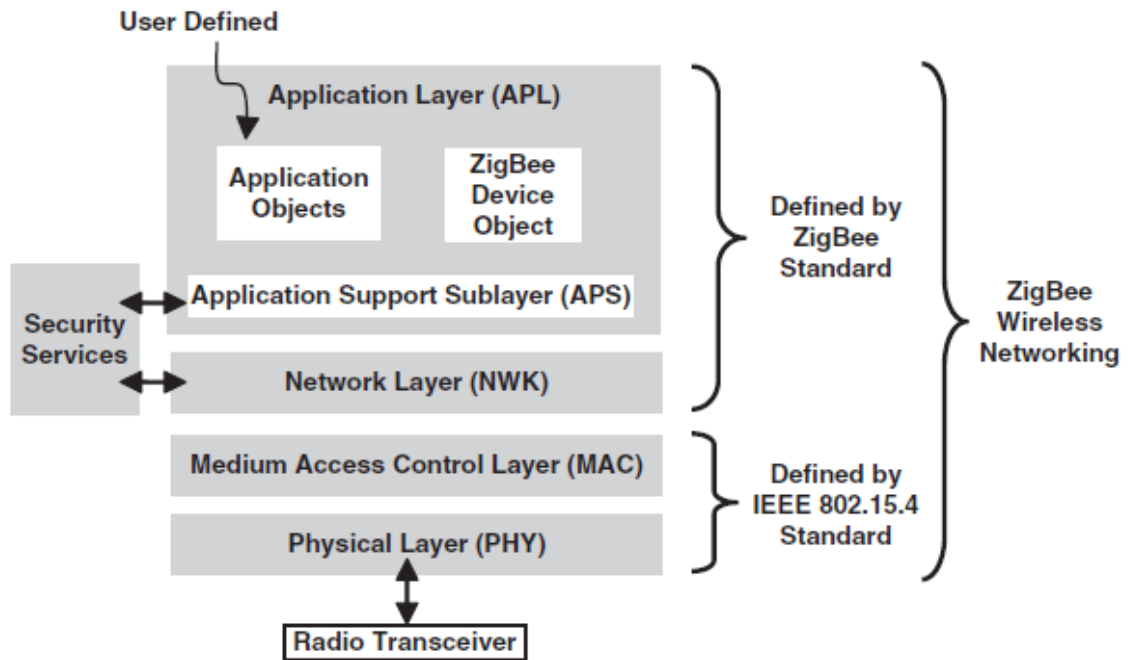


Figura 3.4: Camadas de protocolo do padrão Zigbee [5]

Na Figura 3.4 é possível observar que as duas camadas inferiores são definidas pelo padrão IEEE 802.15.4[8]. O padrão IEEE 802.15.4 foi inicialmente lançado em 2003 e define as especificações para as camadas Física e MAC de redes de comunicações sem fio, porém não especifica os requerimentos para as camadas de redes superiores [5]. O Padrão Zigbee define as camadas de rede, aplicação e segurança do protocolo e adota as camadas física e MAC do padrão IEEE 802.15.4. Este último foi desenvolvido independentemente do padrão Zigbee e é possível construir redes de comunicação sem fio de curta distância baseando – se somente no padrão IEEE 802.15.4. Nesse caso o usuário terá de desenvolver as camadas restantes do protocolo e normalmente essas novas camadas são mais simples e possuem objetivos específicos [5].

3.3 CARACTERÍSTICAS DAS REDES ZIGBEE

3.3.1 Frequências de operação e taxa de dados

Na última versão do padrão IEEE 802.15.4 existiam três bandas de frequências possíveis[8]:

- 868-868.6 MHz (Banda de 868 MHz)
- 902-928 MHz (Banda de 915 MHz)
- 2400 MHz- 2483.5 MHz (Banda de 2.4 GHz)

A banda de 868MHz é usada principalmente para aplicações de redes sem fio de curta distância na Europa[6]. As bandas de 915 MHz e de 2.4 GHz são parte das frequências ISM (Industrial, Scientific and Medical). A banda de 915 MHz é usada principalmente nos Estados Unidos enquanto que a banda de 2.4GHz é mais difundida mundialmente [6].

A Tabela 3.1 fornece detalhes de como essas três bandas de frequências são utilizadas no padrão 802.15.4

Tabela 3.1: Taxas de dados e frequências de operação do padrão 802.15.4 [5]

| | Frequency (MHz) | Number of Channels | Modulation | Chip Rate (Kchip/s) | Bit Rate (Kb/s) | Symbol Rate (Ksymbol/s) | Spreading Method |
|----------|-----------------|--------------------|------------|---------------------|-----------------|-------------------------|---------------------|
| | 868-868.6 | 1 | BPSK | 300 | 20 | 20 | Binary DSSS |
| | 902-928 | 10 | BPSK | 600 | 40 | 40 | Binary DSSS |
| Optional | 868-868.6 | 1 | ASK | 400 | 250 | 12.5 | 20-bit PSSS |
| | 902-928 | 10 | ASK | 1600 | 250 | 50 | 5-bit PSSS |
| Optional | 868-868.6 | 1 | O-QPSK | 400 | 100 | 25 | 16-array orthogonal |
| | 902-928 | 10 | O-QPSK | 1000 | 250 | 62.5 | 16-array orthogonal |
| | 2400-2483.5 | 16 | O-QPSK | 2000 | 250 | 62.5 | 16-array orthogonal |

Como pode ser visto pela tabela anterior, as redes que utilizam o padrão IEEE 802.15.4 possuem uma especificação obrigatória e duas opcionais. A especificação obrigatória é mais simples de implementar, porém trabalha com taxa de dados mais baixas. Também da tabela é possível verificar que o padrão IEEE 802.15.4 trabalha com três tipos diferentes de modulação: *Binary Phase Shift Keying* (BPSK), *Amplitude Shift Keying* (ASK) e *Offset Quadrature Phase Shift Keying* (O-QPSK). Nas modulações do tipo BPSK e O-QPSK, a informação digital está contida na fase do sinal, enquanto que na modulação ASK a informação está contida na amplitude do sinal. Todos os métodos de comunicação sem fio do padrão IEEE 802.15.4 utilizam-se das técnicas de espalhamento espectral do tipo *Direct Sequence Spread Spectrum* (DSSS) ou *Parallel Sequence Spread Spectrum* (PSSS). Essas técnicas são utilizadas para melhorar o desempenho dos receptores em ambientes envolvendo múltiplos caminhos [5].

A camada física é responsável pela especificação das frequências de operação e da taxa de dados. Ela é responsável por todas as características físicas da rede, o que inclui também a escolha dos tipos de dispositivos e os requerimentos de sensibilidade dos mesmos.

3.3.2 Dispositivos

Existem dois tipos de dispositivos em redes que empregam o padrão IEEE 802.15.4: *full function devices* (FFDs) e *reduced function devices* (RFDs) [5]. Um dispositivo FFD é capaz de executar todas as tarefas descritas pelo padrão e estão habilitados para trabalhar em qualquer função na rede. Um dispositivo RFD, ao contrário, pode se comunicar somente com um dispositivo FFD. Dessa forma, os dispositivos RFD são designados para ocupar funções simples enquanto que os dispositivos FFD estão ocupados com as funções mais exigentes. Os dispositivos RFD possuem menos poder de processamento e memória se comparados com os dispositivos FFD [6].

Os dispositivos FFD podem assumir três papéis em uma rede IEEE 802.15.4 [8]: *PAN Coordinator* ou coordenador PAN, *Coordinator* ou coordenador e *Device* ou dispositivo. O padrão Zigbee utiliza uma terminologia diferente. Segundo este padrão, os dispositivos FFD podem assumir as seguintes funções: *ZigBee Coordinator* (ZC) ou coordenador Zigbee, *ZigBee Router* (ZR) ou roteador, e o *ZigBee End Device* (ZED) ou dispositivo final. O *Zigbee Coordinator* é o Coordenador PAN do padrão IEEE 802.15.4. O *Zigbee router* é um coordenador e o *Zigbee End Device* é o *Device* do padrão IEEE 802.15.4. O *Zigbee End*

Device é o dispositivo com menor memória, poder de processamento e capacidade. Obviamente, é também o dispositivo com menor custo. A Figura 3.5 resume as diferentes nomenclaturas para as funções dos dispositivos dos padrões IEEE 802.15.4 e Zigbee.

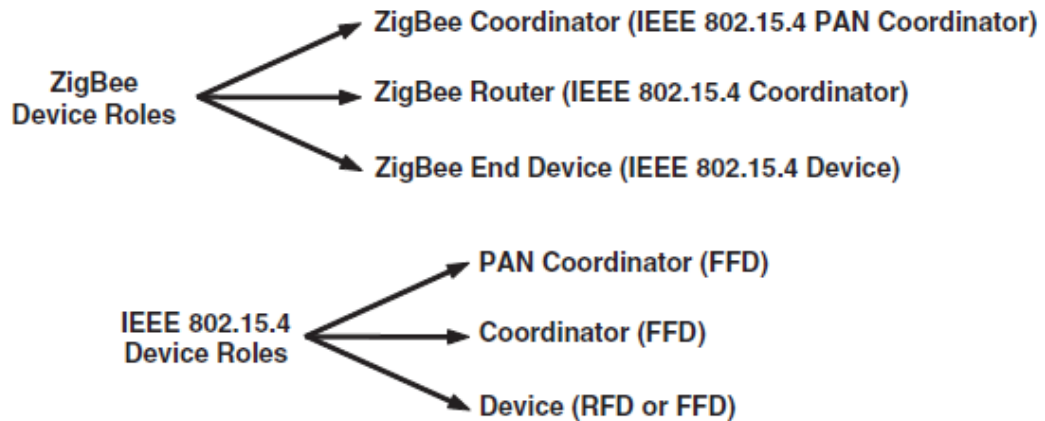


Figura 3.5: Nomenclaturas para as funções dos dispositivos nos padrões Zigbee e IEEE 802.15.4 [5]

3.3.3 Topologias

A formação de uma rede é gerenciada pela camada de rede Zigbee. A rede deve estar organizada em um dos dois tipos possíveis de topologia especificadas pelo padrão IEEE 802.15.4: *star* ou estrela e *peer-to-peer* ou ponto a ponto [5]. Em uma rede com topologia em estrela todo dispositivo na rede pode se comunicar somente com o coordenador PAN. Em redes ponto a ponto cada dispositivo pode se comunicar diretamente com qualquer outro dispositivo da rede desde que estes dispositivos estejam próximos o suficiente para estabelecer um link de comunicação. Em uma rede ponto-a-ponto qualquer dispositivo FFD pode assumir o papel do coordenador PAN. Uma das maneiras de definir quem será o coordenador PAN é escolher o primeiro dispositivo que iniciou a comunicação. Todos os dispositivos que participam da rede e repassam mensagens são dispositivos FFD pois os dispositivos RFD não são capazes de encaminhar mensagens. Um dispositivo RFD pode fazer parte da rede somente se estabelecer comunicação com um único outro dispositivo particular. Em redes ponto a ponto é obrigatória a utilização de dispositivos FFD.

Uma rede ponto a ponto pode assumir diferentes formas de acordo com restrições impostas pelos dispositivos [5]. Se não há restrições de comunicação entre os dispositivos a

topologia da rede é definida como *mesh topology* ou topologia em malha. Outra forma de topologia é a definida como *tree topology* ou topologia em árvore. Nesse caso o coordenador PAN estabelece uma rede inicial. Os roteadores Zigbee formam os galhos e repassam as mensagens, podendo expandir a rede para uma área exterior a área inicial definida pelo Coordenador PAN. Nas Figuras 3.6, 3.7 e 3.8 são ilustradas as topologias discutidas nessa seção.

Uma rede do padrão IEEE 802.15.4, independente do tipo de topologia, é sempre criada pelo coordenador PAN. Este controla a rede e é responsável por importantes funções como: alocação de um endereço único de 16 ou 64 bits (vide seção 3.5) para cada dispositivo na rede; inicialização, término e roteamento de mensagens através da rede; seleção do único identificador PAN para a rede [5]. Existe somente um identificador PAN em toda a rede e este, geralmente, necessita ficar ativo por longos períodos. Assim, normalmente é conectado a uma fonte de energia principal e não a baterias, como a maioria dos outros dispositivos.

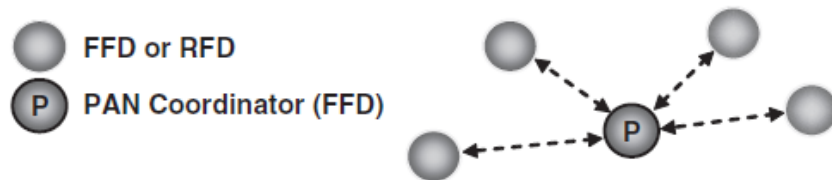


Figura 3.6: Rede com Topologia Estrela [5]

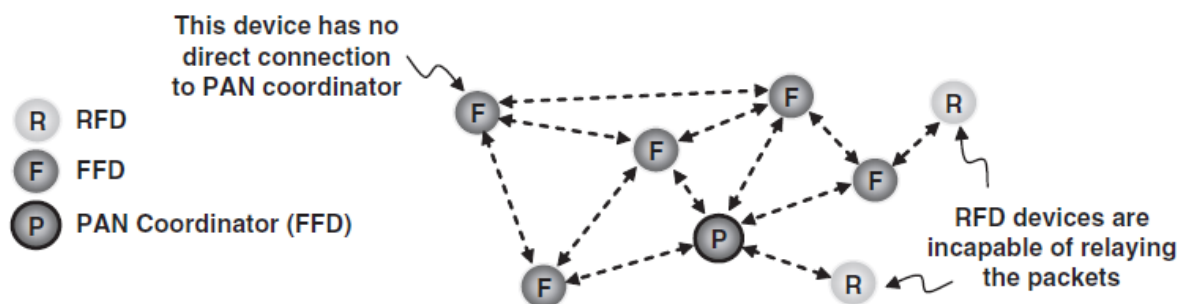


Figura 3.7: Rede com topologia em malha [5]

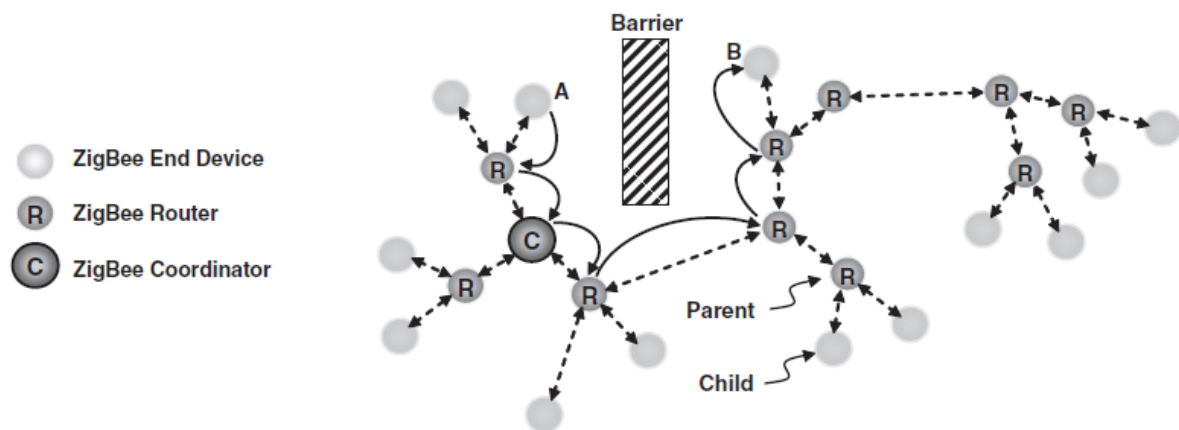


Figura 3.8: Rede com topologia em árvore [5]

Redes Zigbee são consideradas *self-forming networks* e redes Zigbee em topologia *mesh* são também consideradas *self-healing networks*[5]. A primeira definição deriva do fato de que nenhuma supervisão adicional é necessário para a formação de uma rede Zigbee. Em uma rede *mesh*, por exemplo, o primeiro dispositivo FFD que iniciar a comunicação pode se auto estabelecer como o coordenador PAN da rede e os outros dispositivos irão se associar através de uma requisição de associação (mais detalhes na seção 3.5). Em redes *mesh*, se uma rota em uso na transmissão pára de funcionar (pelo esgotamento da bateria, por exemplo) a rede é capaz de selecionar uma rota alternativa e por isso é considerada também uma *self-healing network*.

Redes Zigbee são redes sem fio ad-hoc [11]. Em redes ad-hoc, quando existem nós dispostos a enviar mensagens para outros dispositivos, a rota que irá ser usada durante o encaminhamento dos pacotes, da origem para o destino, é selecionada dinamicamente baseada na conectividade da rede. Se as condições da rede são alteradas, o roteamento deverá ser alterado. É o oposto do que ocorre em redes que já possuem uma estrutura delimitada e em que existem dispositivos atuando sempre como roteadores.

3.4 COMUNICAÇÃO EM REDES ZIGBEE

3.4.1 CSMA-CA

O padrão IEEE 802.15.4 utiliza a técnica *Carrier Sense Multiple Access with Collision Avoidance* (CSMA-CA) para permitir que vários dispositivos usem o mesmo canal de frequência como meio de comunicação [6]. Na técnica CSMA-CA, no momento em que o dispositivo quer transmitir, ele irá executar um processo de *clear channel assessment* (CCA) para assegurar que o canal não está em uso por outro dispositivo. Após verificar que o canal está livre o dispositivo passará a transmitir o sinal. Através da medição da energia espectral do canal de frequência de interesse o dispositivo poderá discernir se o canal está ou não livre.

Quando o dispositivo quiser transmitir ele entrará inicialmente no método de detecção para tentar estimar o nível de energia do canal de frequência desejado. Nesse procedimento, conhecido como *Energy Detection* (ED), o receptor não tenta decodificar o sinal mas apenas avaliar o seu nível de energia[5].

Outra forma de detectar se o canal de frequência está livre é através do método *Carrier Sense* (CS) [5]. Neste método é determinado o tipo de sinal que está ocupando o canal, em oposição ao ED, e se for confirmada a presença de um sinal do padrão IEEE 802.15.4 o dispositivo terá de decidir se o canal está ocupado ou não baseado em algum tipo de limiar predeterminado. Caso o canal não esteja livre, o sistema aguardará um período aleatório, um *backoff time*, para tentar novamente. Esse processo é repetido várias vezes até que o dispositivo atinja um número limite de tentativas que é definido previamente.

3.4.2 Presença de *beacon* em redes Zigbee

Existem dois métodos para acesso de canal em redes Zigbee: com contenção ou sem contenção (*contention-based* e *contention free*) [5]. No método *Contention-based Channel Access* todos os dispositivos irão usar o mecanismo CSMA-CA quando quiserem transmitir dados e o primeiro que encontrar um canal livre irá começar a transmissão. No método livre de contenção o coordenador PAN irá dedicar um intervalo de tempo específico para um determinado dispositivo, esse intervalo é conhecido como *Guaranteed Time Slot* (GTS). Com um slot GTS definido, o dispositivo irá começar a sua transmissão somente quando o seu GTS alocado iniciar e nesse caso o dispositivo não irá usar o método CSMA-CA.

Para garantir a alocação do GTS a rede precisa estar em sincronia, ou seja, o coordenador PAN precisa garantir que todos os dispositivos na rede estejam sincronizados. Para conseguir atingir esse objetivo a rede irá fazer uso do envio de mensagens de sinalização, as mensagens de *beacons*. Esse tipo de rede é chamado de *beacon-enabled networks*. Nessas redes o coordenador PAN terá a opção de enviar ou não sinais de *beacons* para sincronizar cada dispositivo anexado à rede. Toda a estrutura e o formato das mensagens de *beacon* são mostrados na seção 3.8.2.1. Apesar de melhorar o acesso aos canais de transmissão e recepção as redes que utilizam *beacons* apresentam a uma desvantagem que pode ser crucial em redes de sensores, o gasto de energia. Em redes com *beacons* habilitados os dispositivos terão de entrar em modo ativo periodicamente para processos de sincronização, mesmo que não tenham nenhuma outra tarefa enquanto estejam neste modo. Dessa forma, o tempo de vida das baterias em *beacon-enabled networks* é geralmente menor do que em redes que não enviam sinais de sinalização. Estas redes que não enviam *beacons* são chamadas de *nonbeacon networks*. Neste tipo de rede os dispositivos não possuem um *slot* de tempo garantido para envio de dados e nesse caso a rede não terá períodos livres de contenção, já que não há sincronização entre os dispositivos. Como o dispositivo entra no modo ativo menos freqüentemente em *nonbeacon networks*, o tempo de vida de baterias em redes deste tipo são maiores do que em redes com presença de *beacons*.

3.4.3 Métodos de transferência de dados

Em redes do padrão IEEE 802.15.4 existem três tipos de métodos de transferência de informações: transferência de dados do coordenador para o dispositivo; transferência do dispositivo para o coordenador; transferência entre dois dispositivos [14].

Todos os métodos de transferência de dados podem ser utilizados em redes com topologia ponto-a-ponto. Em redes com topologia estrela, somente os métodos de transmissão entre coordenador e dispositivos são permitidos.

3.4.3.1 Transferências do dispositivo para o coordenador

Quando um dispositivo quer transmitir informação para o coordenador, em redes com utilização de mensagens de sinalização, ele precisará sincronizar o seu relógio com o coordenador periodicamente e usará o método CSMA-CA quando a transmissão não ocorrer

durante o período do *slot* GTS[5]. Esse processo de sincronização é representado pela flecha de *beacon* da Figura 3.9(a). Após a sincronização, o dispositivo pode enviar as informações, representado pela flecha *Data* na mesma figura. Se o receptor exigir uma mensagem de confirmação, o coordenador então enviará uma mensagem de *acknowledgment*, como também ilustrado na Figura 3.9 (a). A Figura 3.9 (b) ilustra o caso em que o dispositivo quer transmitir para o coordenador em redes sem a presença de *beacons*. Nesse caso, o dispositivo irá transmitir a informação somente quando o canal estiver limpo. Após detectar que o canal está livre, o dispositivo envia as informações. Essa etapa é representada pela flecha *Data* da Figura 3.9(b). Novamente a transmissão de uma mensagem de confirmação (*acknowledgment*) também é opcional.

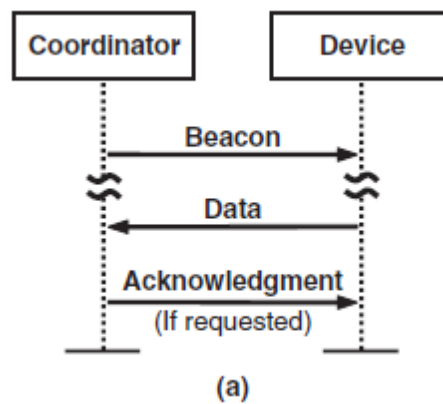


Figura 3.9 (a): Transferência de dados do dispositivo para o coordenador em redes com utilização de *beacons*. [5]

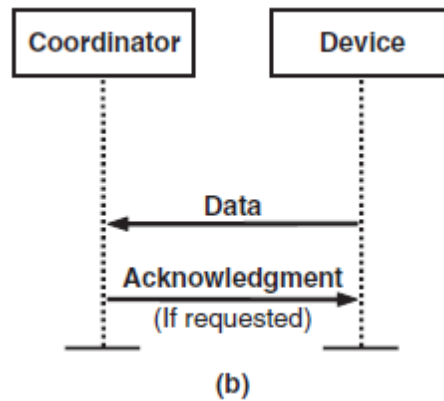


Figura 3.9 (b): Transferência de dados do dispositivo para o coordenador em redes sem *beacons*. [5]

3.4.3.2 Transferências do coordenador para o dispositivo.

Em *beacon-enabled networks*, se o coordenador quer transmitir dados para um dispositivo em particular ele irá indicar para este dispositivo de destino, através de uma mensagem de *beacon*, que existem dados pendentes para ele. O dispositivo então enviará uma mensagem de *data request* para o coordenador indicando que está pronto para receber as informações. O coordenador envia então uma mensagem (obrigatória) para confirmar que recebeu a requisição. Logo em seguida o coordenador envia os dados para o dispositivo. O envio de uma mensagem de confirmação pelo dispositivo é opcional. A Figura 3.10 (a) ilustra esse processo.

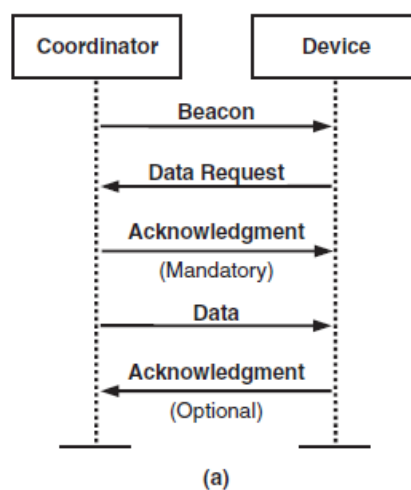


Figura 3.10: Transferência de dados do coordenador para o dispositivo em redes com *beacons*. [5]

Em *nonbeacon networks* o coordenador necessita esperar pelo dispositivo para requisitar informações. Se o dispositivo requisitar dados, como ilustrado através da flecha *Data Request* da Figura 3.10(b), e não houver mensagens de dados para esse dispositivo, o coordenador então enviará uma mensagem de *acknowledgment* com formato específico para indicar que não há informações pendentes para ele. Alternativamente, se houver mensagens de dados, o coordenador enviará essas informações, que estão representadas na Figura 3.10 (b) pela flecha *Data*. Uma mensagem de confirmação enviada pelo dispositivo é também é opcional para esse caso.

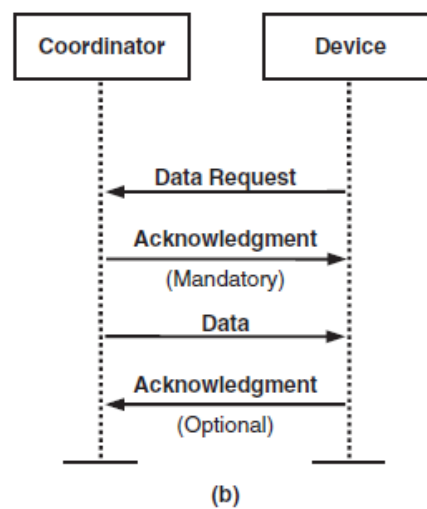


Figura 3.11: Transferência de dados dos dispositivos para o coordenador em redes com *beacons*. [5]

3.4.3.3 Transferências de dados entre dois dispositivos.

Em redes com topologia ponto-a-ponto (ou malha) todos os dispositivos podem se comunicar entre si. Nesse caso, os dispositivos estarão em sincronia na maioria das aplicações.

3.5 ENDEREÇAMENTO EM REDES DO PADRÃO IEEE 802.15.4

Todos os dispositivos em rede necessitam de um endereço único. Redes do padrão IEEE 802.15.4 usam dois métodos de endereçamento: endereçamento curto de 16 bits ou o

endereçamento longo com 64 bits [8]. Em redes com endereçamento curto será permitida comunicação dentro de uma única rede. Nesse caso, o uso de endereços curtos permite uma redução no tamanho das mensagens e um maior espaço de memória estará livre. O endereçamento curto combinado com um identificador de coordenadores PAN (*PAN identifier*) permite a comunicação entre redes independentes.

Uma rede pode definir se vai usar tanto o endereçamento de 16 bits ou de 64 bits. Para este último caso, uma quantidade muito maior de dispositivos poderá ser integrada à rede (2^{64} ou 1.8×10^{19}).

A camada de rede (NWK) do protocolo Zigbee aloca um endereço de rede (NWK) de 16 bits em adição ao endereço IEEE. Uma tabela é usada para mapear cada um dos endereços IEEE de 64 bits a um único endereço NWK. Cada transceptor em uma rede pode ter um único endereço IEEE e um único endereço NWK.

3.6 VERIFICAÇÃO DE DADOS

Os receptores nos processos de comunicação em rede precisam de uma técnica para verificar se os pacotes recebidos estão isentos de erros. Um pacote é formado por certa quantidade de bits que são transmitidos em conjuntos com um formato específico. O padrão IEEE 802.15.4 utiliza um quadro de sequência de reconhecimento de 16 bits (*Frame Check Sequence* ou FCS) baseado na *Cyclic Redundancy Check* (CRC) da *International Telecommunication Union* (ITU) para detectar possíveis erros nas transmissões de pacotes[5].

3.7 PROCESSOS DE ASSOCIAÇÃO E DISSOCIAÇÃO.

O padrão IEEE 802.15.4 fornece serviços que podem ser usados para permitir que um dispositivo se anexe ou deixe uma rede [8]. São os serviços de associação e dissociação. Quando um dispositivo quer se juntar a uma rede ele envia uma mensagem de requisição de associação para o coordenador PAN e este poderá rejeitar ou aceitar o pedido. Da mesma forma, se um dispositivo pretende sair de uma rede ele envia uma mensagem de dissociação para notificar o coordenador PAN da sua intenção.

3.8 CAMADAS DO PROTOCOLO ZIGBEE

As camadas de protocolo do padrão Zigbee estão ilustradas com mais detalhes na Figura 3.12. Como visto anteriormente, as camadas do protocolo Zigbee estão baseadas no modelo OSI, que define sete camadas independentes. Entretanto, o Zigbee implementa somente as camadas responsáveis pelas características das redes sem fio de baixa potência e baixa transferência de dados. As duas camadas inferiores são definidas pelo padrão IEEE 802.15.4 enquanto que as camadas de rede (NWK) e aplicação (APL) são definidas pelo Zigbee. As funcionalidades de segurança são definidas pelos dois padrões. Uma rede que implementa todas as camadas da Figura 3.12 é considerada uma rede Zigbee [5].

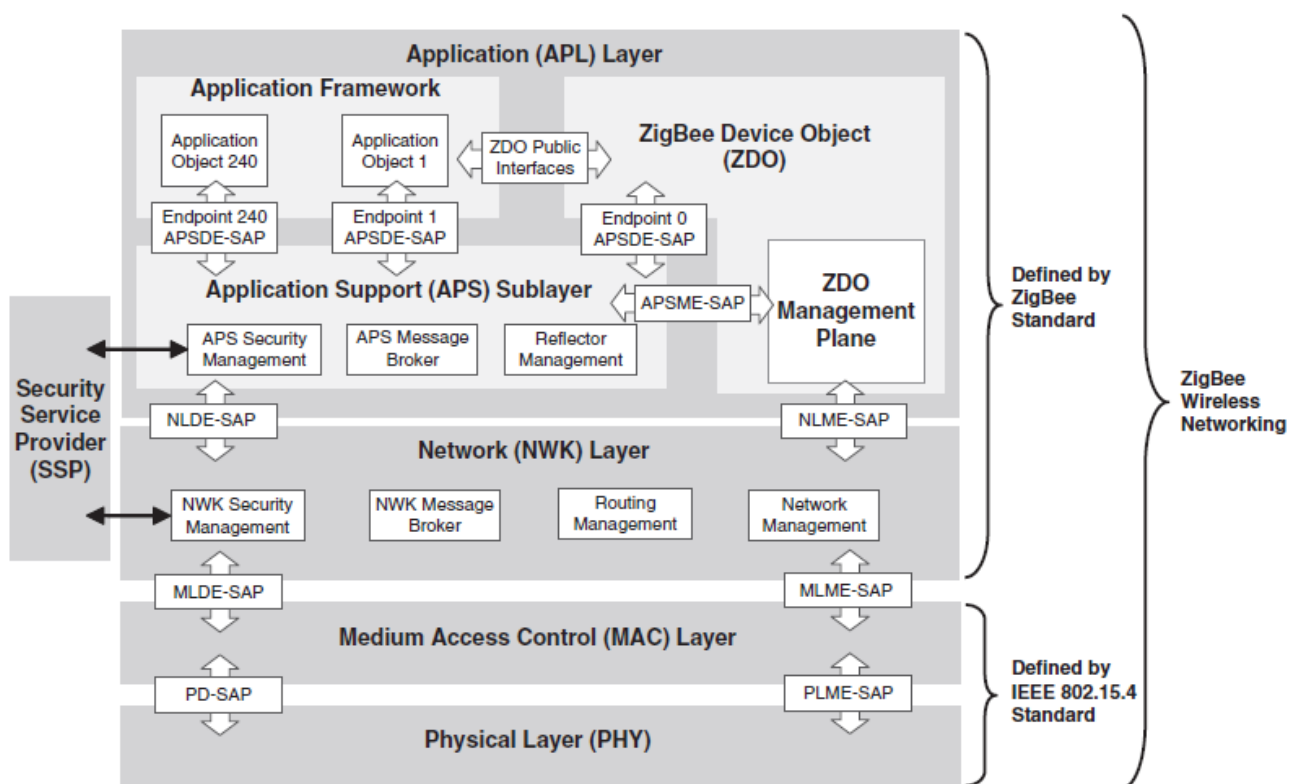


Figura 3.12: Camadas de protocolo do padrão Zigbee [5]

3.8.1 Camada física (PHY) do padrão IEEE 802.15.4

Em redes Zigbee a primeira camada da pilha de protocolos é a camada física do padrão IEEE 802.15.4 [5]. É a camada mais próxima do Hardware do sistema, controlando diretamente o transceptor de rádio. A camada física é responsável pelas seguintes tarefas: ativação do rádio que irá transmitir ou receber pacotes; detecção de energia, ou *Energy Detection* (ED) dentro do canal atual; *Clear Channel Assessment* (CCA) ou Verificação de Ocupação do Canal; Indicação de Qualidade do *Link* (LQI) para a recepção de pacotes e Transmissão e Recepção dos Dados. A seleção da frequência do canal, que irá certificar que nenhum outro dispositivo esteja usando o canal atual, também é outra das responsabilidades da camada física [26].

Comandos e informações são trocados entre vários dispositivos da rede na forma de pacotes. A estrutura geral de um pacote está ilustrada na Figura 3.13. Os pacotes da camada física são compostos por três componentes: o SHR, ou *Synchronization Header*, o PHR, ou *PHY reader* e o PHY *payload*. O primeiro desses componentes habilita o receptor a iniciar a sincronização, o segundo contém informações sobre o tamanho do *frame* e o último componente é fornecido pelas camadas superiores e contém informações e comandos para os dispositivos receptores.

O MAC *frame* é dividido em três partes e é transmitido para os outros dispositivos como o *payload* da camada física. É composto pelo MAC *header* (MHR) que contém informações de endereçamento e segurança, pelo MAC *payload*, que possui tamanho variável e contém informações e comandos, e pelo MAC *footer* (MFR), uma sequência de quadros de 16 bits para verificação de dados.

O NWK *frame* é formado por duas partes: o NHK *reader* (NHR) e o NHK *payload*. O primeiro tem informações de endereçamento ao nível de rede e informações de controle. O segundo é fornecido pela subcamada APS. A Figura 3.13 ilustra a relação entre os tipos de pacotes do padrão Zigbee e mostra que o primeiro bit transmitido é o bit menos significativo (LSB) do *Synchronization Reader* (SHR).

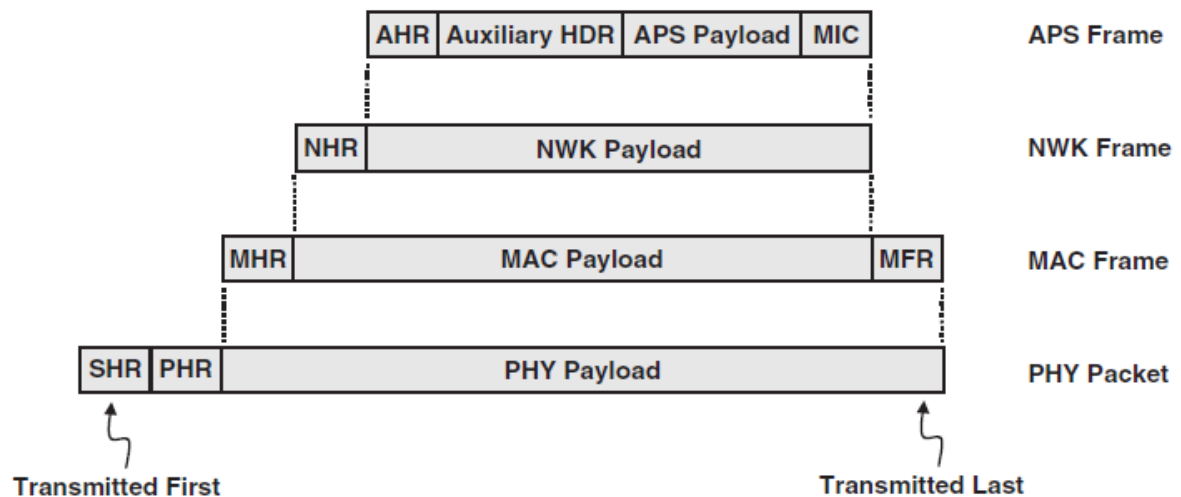


Figura 3.13: Estrutura de pacotes de redes Zigbee [5]

3.8.2 Camada MAC do padrão IEEE 802.15.4

A camada MAC (*Medium Access Control*) fornece a interface entre a camada física e a camada NWK. Ela é responsável, para o caso de *beacon enabled networks*, pela geração de *beacons* e pela sincronização dos dispositivos para o recebimento e envio das mensagens de *beacons*. A camada MAC também fornece serviços de associação e dissociação. Outras de suas responsabilidades são: utilizar a técnica *Carrier Sense Multiple Access with Collision Avoidance* ou Portadora de Múltiplo Acesso com Prevenção de Colisão (CSMA-CA); utilização e manutenção do mecanismo *Guaranteed Time Slot* (GTS) para realizar a transmissão direta, indireta e reserva dos dados e, finalmente, por proporcionar um *link* confiável entre duas entidades MAC pares [8] [26].

O padrão IEEE 802.15.4 define quatro estruturas MAC Frames: *Beacon frame*, *Data frame*, *Acknowledgement frame* e *MAC command frame*.

3.8.2.1 Beacon Frame.

O *Beacon frame* é usado pelo nó coordenador PAN para enviar *beacons* que são usados para sincronização de todos os dispositivos presentes na rede [5] [8]. A estrutura do *beacon frame* é mostrada na Figura 3.14. Todo o MAC frame é usado como *payload* em um pacote da camada física. Todo o conteúdo do *payload* PHY é referido como *PHY Service*

Data Unit (PSDU). Em um pacote da camada física o campo de preâmbulo é usado pelo receptor para sincronização. O campo SFD (*Start-of-frame delimiter*) indica o fim do SHR e o começo do PHR. O campo *frame length* especifica o número total de octetos no *payload* físico.

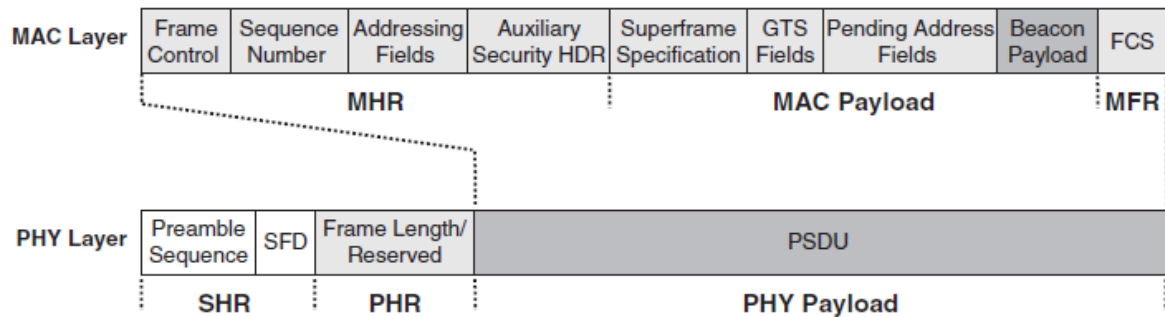


Figura 3.14: Estrutura do MAC *beacon frame* [5].

O *payload* MAC é fornecido pela camada de rede (NWK) e o *superframe* é um quadro que faz fronteira entre dois *beacon frames*. O *superframe* é opcional e é usado somente em redes com transmissão de *beacons*, as *beacon-enabled networks*.

Em redes no modo *beacon* é possível utilizar o quadro do *superframe*, cuja estrutura está ilustrada na Figura 3.15. O quadro do *superframe*, que é de uso opcional no padrão IEEE 802.15.4, faz fronteira com dois quadros de *beacon*. A estrutura do *superframe* determina o período entre dois *beacons* consecutivos. Dentro do *superframe* pode haver três tipos de períodos: *contention access period* (CAP), *contention free period* (CFP) e o período inativo.

Durante o período CAP, todos os dispositivos que querem transmitir devem usar o mecanismo CSMA-CA para ganhar acesso ao canal. O canal de frequência, neste caso, estará igualmente disponível para todos os dispositivos na rede. O primeiro dispositivo que começa a transmitir usando um canal disponível irá permanecer no canal até que sua transmissão esteja completa. Se o dispositivo encontrar o canal ocupado, ele irá aguardar um certo período aleatório (*backoff time*) e tentar novamente.

Não há garantia de que durante o período CAP o canal esteja livre quando um dispositivo necessitar. Entretanto, o período CFP irá garantir um *slot* de tempo, o GTS, para um dispositivo específico e para esse dispositivo não será necessário o uso do mecanismo CSMA-CA para acesso do canal. A combinação dos períodos CFP e CAP é chamada de período ativo. O período ativo é dividido em 16 *slots* de tamanho iguais e o quadro de *beacon*

sempre começa no início do primeiro *slot*. Podem haver até sete *slots* GTS no período CFP e cada GTS pode ocupar mais de um *slot*. Opcionalmente, a estrutura do *superframe* também pode ter um período inativo que permite um dispositivo entrar em modo de economia de energia.

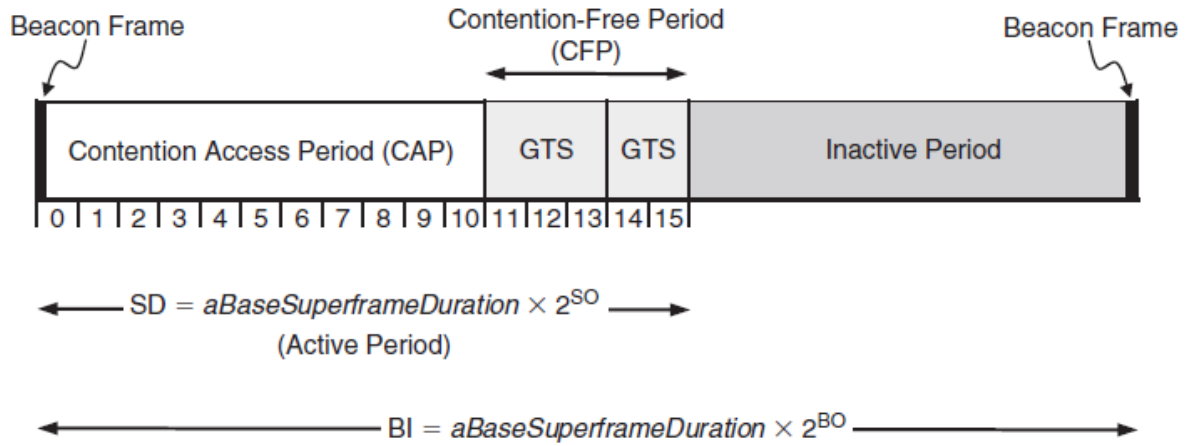


Figura 3.15: Estrutura do *Superframe* [5]

A estrutura do *superframe* é definida pelo coordenado PAN e configurada pela camada de rede. O *Beacon Interval* (BI) é o tempo de duração entre dois *beacons* consecutivos e é determinado pelo valor do parâmetro *macBeaconOrder* (BO) e pela constante *aBaseSuperframeDuration*, como pode ser visto na Equação 3.1:

$$BI = aBaseSuperframeDuration \times 2^{\text{macBeaconOrder}} \quad (\text{símbolos}) \quad (3.1)$$

Da mesma forma, o tamanho do período ativo, conhecido como *Superframe Duration* é calculado pela Equação 3.2, onde o atributo *macSuperframeOrder* (SO) substitui o BO. Todas as constantes e atributos são fornecidos na documentação do padrão IEEE 802.15.4 [8][25].

$$SD = aBaseSuperframeDuration \times 2^{\text{macSuperframeOrder}} \quad (\text{símbolos}) \quad (3.2)$$

A duração do *superframe* não pode exceder o tamanho do *beacon interval* e, dessa forma, o valor de SO é sempre menor ou igual ao valor de BO. Em *nonbeacon networks* o atributo BO é definido com o valor 15 e nesse caso a rede não transmitirá *beacons* a não ser que receba uma requisição para isso.

O *beacon frame* não é só usado para sincronização dos dispositivos da rede mas também é usado pelo coordenador para avisar a um dispositivo específico na rede que há alguma informação pendente para aquele dispositivo no coordenador. Esse dispositivo, então, irá contatar o coordenador e fazer a requisição para que este envie as informações para o dispositivo.

3.8.2.2 Data frame.

O quadro *Data frame*, cujo *payload* é fornecido pela camada NWK, é utilizado para transmitir dados[5] [8]. A informação contida no MAC *payload* é referida como MAC *service Data Unit* (MSDU) e o MAC *data frame*, que irá se tornar PHY *payload*, é definido como MAC *Protocol Data Unit* (MPDU). A estrutura do *Data frame* é ilustrada na Figura 3.16.

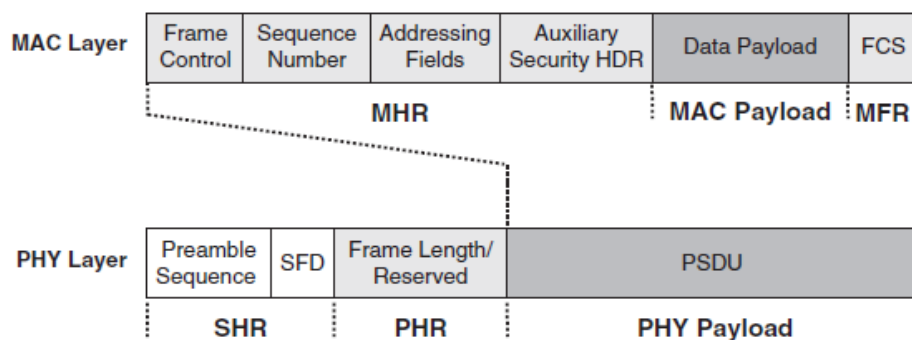


Figura 3.16: Estrutura do MAC *data frame* [5].

3.8.2.3 Acknowledgment frame.

Este *frame*, que está ilustrado na Figura 3.17, é o tipo mais simples encontrado entre os formatos de MAC *frames* e não carrega consigo qualquer MAC *payload*. O *Acknowledgment frame* é enviado de um dispositivo para outro a fim de confirmar uma recepção de pacotes ocorrida com sucesso.

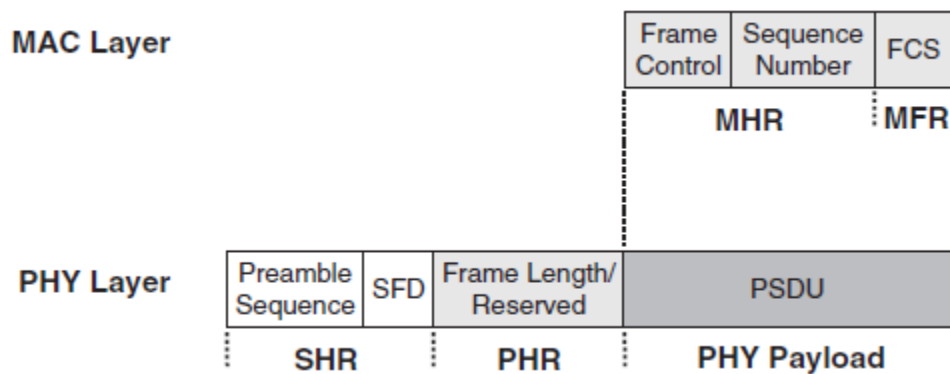


Figura 3.17: Estrutura do MAC *Acknowledgment frame* [5].

3.8.2.4 *Command frame*.

Comandos MAC, como o requerimento de associações e dissociações com a rede são transmitidos usando o quadro *Command Frame*, ilustrado na Figura 3.18. O campo *command type* determina o tipo de comando (requisição de dados ou de associação) e o *command payload* contém o comando propriamente dito. Toda a estrutura do *Command Frame* é colocada no *payload* da camada física e é denominada PSDU ou *PHY Service Data Unit*.

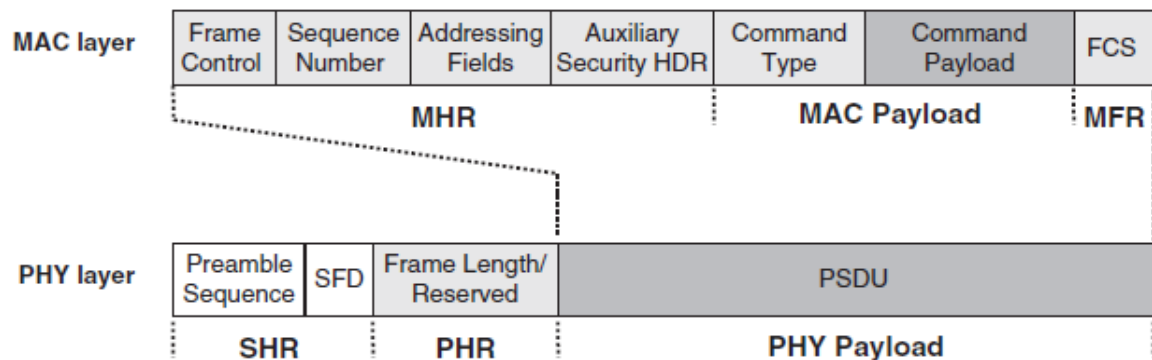


Figura 3.18: Estrutura do MAC *command frame* [5].

3.8.3 Camada de rede (NWK) do padrão zigbee

A camada de rede faz a interligação entre a camada MAC e a camada de aplicação (APL) [25]. É responsável pelo gerenciamento, pela formação e roteamento da rede. Em redes

Zigbee o coordenador e os roteadores são responsáveis pelo descobrimento e manutenção das rotas na rede. Um dispositivo final Zigbee, ou *Zigbee end device*, não pode executar o procedimento de descobrimento de rota e dessa forma essa tarefa ficará a cargo do coordenador, que executará o processo pelo dispositivo final *Zigbee*. A camada de rede, dessa forma, é a responsável por selecionar o tipo de topologia e pelo estabelecimento de uma nova rede.

3.8.4 Camada de aplicação (APL) do padrão zigbee

É a camada superior das redes sem fio Zigbee [25]. A camada de aplicação (APL) provê o gerenciamento e suporte ao *ZigBee Device Object* (ZDO), que é onde está definido a função do dispositivo ZigBee na rede. Cada fabricante irá desenvolver objetos de aplicações que irão customizar dispositivos para várias aplicações. Objetos de aplicações controlam e gerenciam as camadas de protocolo em um dispositivo Zigbee. A camada APL é a que também oferece serviços de segurança para a rede.

4. SIMULAÇÃO

Para a simulação e análise do objetivo proposto foi utilizada a versão 2.33 do software *Network Simulator 2* (NS2), rodando no sistema operacional Linux, distribuição Ubuntu 9.04. O NS2 é um simulador de eventos discretos que foi desenvolvido na linguagem orientada a objetos C++ por um grupo formado pela UC Berkeley, USC/ISI, LBL e Xerox PARC [10] [24]. Para simulações o NS utiliza duas linguagens de programação, C++ e OTcl. O uso da linguagem C++ oferece mais praticidade e velocidade na implementação de protocolos, melhorando o desempenho de memória. Por outro lado, a interface com o usuário, o estabelecimento de parâmetros e a manipulação de objetos são feitas em modo texto, através da linguagem OTcl [10].

O NS, ao utilizar esses dois tipos de linguagem, satisfaz as suas maiores necessidades. Para tarefas que envolvem simulações detalhadas de protocolo, que requerem uma linguagem de programação que possa manipular bytes de forma eficiente e implementação de algoritmos que rodem por longos períodos, o programa utiliza a linguagem C++. Entretanto, para tarefas que envolvam pequenas variações de parâmetros ou configurações e a exploração de um grande número de cenários, casos em que o tempo de iteração é mais importante, utiliza-se a linguagem OTcl. Dessa maneira, o NS preenche todos os requisitos para a maior parte das pesquisas envolvendo redes de comunicações.

O NS2 foi escolhido como ferramenta de simulação desse trabalho pois, além das características já citadas, é um software livre bastante difundido no meio acadêmico e com ampla documentação disponível. Todos os detalhes do programa, bem como um manual de instalação e configuração podem ser encontrados na referência [10].

4.1 PARÂMETROS

Neste trabalho, para realizar as simulações utilizou-se os algoritmos implementados por Vaddina Prakash Rao [14] [16]. Esses algoritmos são adições e modificações feitas aos códigos desenvolvidos por Zheng et Lee [26] [27]. É importante destacar que a partir do código criado por Zheng é que foi originalmente incluído o suporte ao padrão IEEE 802.15.4 no programa de simulação NS2 (a partir da versão 2.28). Na estrutura de arquivos do NS2

esse suporte é encontrado no arquivo denominado *wpan.tcl*. A modificação feita por Rao para suporte à banda de 868 MHz e redes em topologia estrela denomina-se *wpan868*.

Em todos os cenários alguns parâmetros foram mantidos inalterados. Esses parâmetros escolhidos estão definidos a seguir:

4.1.1 Parâmetros de tráfego e nós

A topologia da rede está no formato estrela e é composta por 20 nós em todos os casos. Nesta rede existe somente um nó coordenador PAN central que irá receber os dados dos nós circundantes, distantes 10m do nó central, com um tráfego do tipo CBR ou *Constant Bit Rate*. O tráfego ocorre somente em um sentido, na direção do coordenador PAN e não há movimento entre os nós. O tamanho dos pacotes é igual a 70 bytes e a taxa de transmissão é igual a 1 pacote por segundo. O tempo total de simulação foi definido como 1000s. Todos esses dados são mostrados na Tabela 4.1. A Figura 4.1 ilustra o formato da rede.

Tabela 4.1: Parâmetros de Tráfego e Nós

| Parâmetro | Valor |
|-------------------------|-----------------------|
| Topologia da Rede | Formato Estrela |
| Número de Nós | 20 |
| Número de Coordenadores | 1 |
| Tipo de Tráfego | CBR |
| Direção do Tráfego | Nó para Coordenador |
| Tamanho dos Pacotes | 70 Bytes |
| Distância entre nós | 10m |
| Tempo de Simulação | 1000s |
| Taxa de Transmissão | 1 pacotes por segundo |

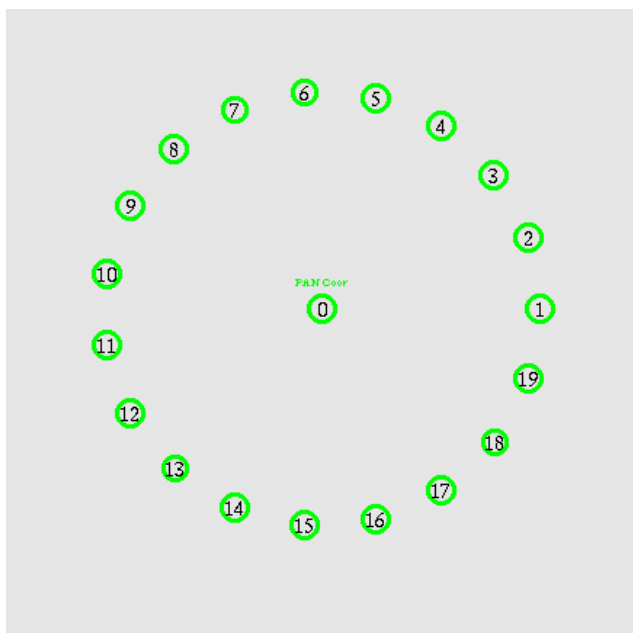


Figura 4.1: Topologia da Rede

Os parâmetros de tráfego e nós foram escolhidos para possibilitar a análise de uma rede simples, pequena, econômica no consumo de energia e banda. Essa rede trabalha com uma baixa taxa de dados e mesmo quando todos os seus nós estão ativos a taxa total de transferência não ultrapassa 10 Kbps, que é um valor suficiente para a maioria das aplicações domésticas de RSSF.

4.1.2 Parâmetros físicos

O NS2 suporta três tipos de modelos de propagação: *Free-Space Model*, *Two-Ray Ground Model* e o *Shadowing Model*. Os dois primeiros são mais direcionados para comunicações de curta distância, enquanto o *Shadowing Model* é mais utilizado para redes em que os nós apresentam maiores distâncias, levando em conta os efeitos de atenuação ou *fading*. Para esta simulação foi escolhido o modelo *Two-Ray Ground*, que considera tanto o caminho de comunicação direto como as reflexões do solo e dessa forma dá uma resposta mais acurada que o *Free-Space Model*. Neste modelo, a potência recebida pelo nó a uma distância d é dada por:

$$Pr(d) = (Pt \times Gr \times Gt \times hr^2 \times ht^2) \div (d^4 \times L) \quad (4.1)$$

em que P_t é a potência de transmissão, G_r é o ganho de recepção, G_t é o ganho de transmissão, h_t é o tamanho da antena transmissora, h_r é o tamanho da antena receptora, d é a distância do transmissor e L é a atenuação (*Pathloss*) [5]. Antenas do tipo Omni direcional foram usadas já que transmitem a mesma potência para todas as direções. Os ganhos de transmissão e recepção foram definidos como 1 (em dB). A Tabela 4.2 resume esses dados:

Tabela 4.2: Parâmetros Físicos

| Parâmetro | Valor |
|----------------------|---------------------|
| Modelo de Propagação | Two-Ray-Ground |
| Tipo de Antena | Omni direcional |
| Ganho de Transmissão | 1 dB |
| Ganho de Recepção | 1 dB |
| Direção do Tráfego | Nó para Coordenador |

4.1.3 Parâmetros de roteamento.

Filas do tipo *First in First Out*, ou *Droptail*, foram usadas em todas as simulações e o tamanho da fila foi escolhido como 200. Após atingir esse limite todos os pacotes são perdidos. A Tabela 4.3 resume esses dados:

Tabela 4.3 Parâmetros de Roteamento

| Parâmetro | Valor |
|-----------------|-----------|
| Tipo de Fila | Drop Tail |
| Tamanho da Fila | 200 |

4.1.4. Parâmetros de potência e energia.

A potência de transmissão é a potência em que o sinal é transmitido. Os dispositivos baseados no padrão IEEE 802.15.4 são produzidos para trabalhar com uma potência de transmissão de 0dBm ou 1 mW e esse valor foi o escolhido para a simulação. A potência consumida por um transceptor para transmitir um pacote foi definida como 0.075W. Esse

valor foi estimado como sendo a transmissão de um pacote de tamanho médio e suprido por uma fonte de 30mA e 2.5V. A potência de recepção é o valor limite mínimo a partir do qual uma transmissão pode ser detectada com sucesso. Foi definido como -92 dBm ($6,3096 \times 10^{-13}$ W) o valor para a potência de recepção para as simulações. A potência consumida por um transceptor ao receber um pacote foi definida como 0.0625W, calculado como sendo a potência consumida por uma descarga de 25mA a 2.5V. A potência consumida por um nó durante o período de inatividade foi calculada como uma descarga de $2.5 \mu\text{A}$ a 2.5V, ou seja, 0.00000625W . O tamanho da antena foi definido como 1 metro. Todos esses parâmetros são repetidos na Tabela 4.4.

Os dispositivos que seguem o padrão IEEE 802.15.4 são produzidos para trabalhar com um consumo muito baixo de energia. Para a simulação assumiu-se que esses dispositivos trabalham com duas pilhas *Duracell* AA simples, cujo gráfico da característica de descarga de corrente é apresentado a na Figura 4.2. Essas baterias fornecem isoladamente uma tensão de 1.5V. Utilizando-se o gráfico da Figura 4.2 como referência, observa-se que com uma descarga constante de corrente entre 10 e 20mA obtém-se um tempo de operação da bateria entre 100 a 200h. Utilizando – se o valor de 180h para uma descarga constante de 15mA e sabendo que $Potência (W) = Tensão (V) \times Corrente (A)$ e que $Energia (J) = Potência (W) \times Tempo (s)$ obtemos uma energia de 14580 Joules:

$$Potência (W) = 1.5V \times 15mA = 0.0225 W$$

$$Energia (J) = 0.0225W \times 180 \times 60 \times 60 = 14580 J$$

Esse valor será considerado a energia inicial dos dispositivos nas simulações com o NS2.

Tabela 4.4 Parâmetros de potência e energia

| Parâmetro | Valor |
|--------------------------------------|-------------|
| Potência de Transmissão | 0 dBm |
| Potência para transceptor transmitir | 0.075W |
| Potência de Recepção | -92 dBm |
| Potência para transceptor receber | 0.0625W |
| Potência consumida na inatividade | 0.00000625W |

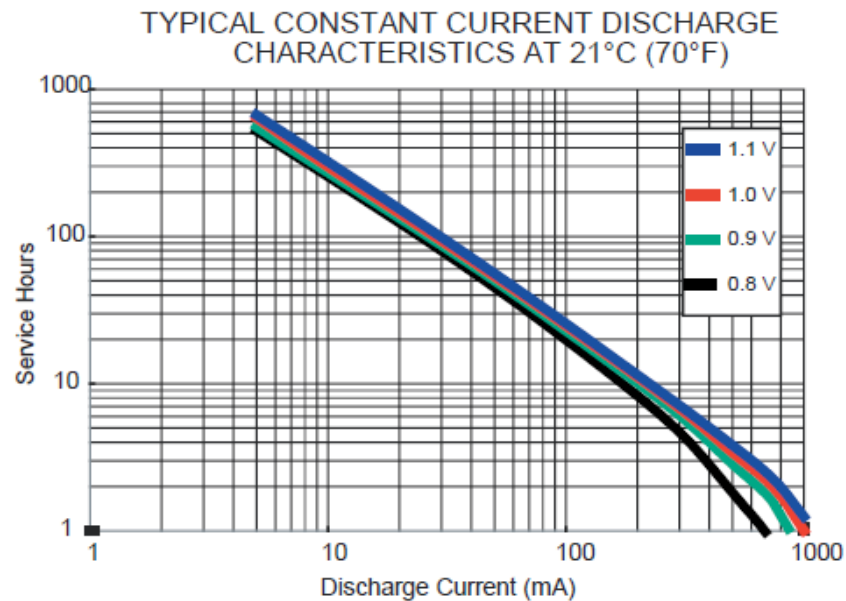


Figura 4.2: Descarga de corrente constante. Fonte: [28]

4.2 CENÁRIOS

Foram elaborados quatro cenários distintos para a simulação. Esses cenários foram então utilizados para analisar a vazão, ou *Throughput*, a *Delivery Ratio* e a quantidade de energia consumida pelos nós sensores de uma rede WPAN com 20 nós em topologia estrela. A rede foi simulada de acordo com as definições do protocolo IEEE 802.15.4 implementadas na ferramenta *Network Simulator 2* e utilizando os parâmetros definidos na seção 4.1.

Em todos os casos variou-se a quantidade de nós ativos na rede. No primeiro cenário a rede está atuando sem que o coordenador PAN esteja transmitindo *beacons*, ou seja, é uma *Nonbeacon Network*. Nos outros três cenários a rede está atuando no modo *Beacon Enabled* e cada um destes cenários representa um diferente tamanho de *Superframe*.

Para a realização das simulações foram definidas como métricas o *Throughput* da rede, o consumo de energia e a *Delivery Ratio*. Essas métricas foram então utilizadas para analisar o comportamento de uma rede de sensores ZIGBEE.

O *Throughput* ou vazão é a medida da quantidade de informação transmitida da fonte para o destino em um período de tempo, que é igual a 1 segundo nesta simulação. Ou seja, é a quantidade total de bits recebidos por segundo. Para calcular o *Throughput* de um nó primeiro é medido o número de pacotes recebidos com sucesso no nó. Em seguida, é computado o número de bits recebidos nesse nó. Esse número é então dividido pelo tempo total de simulação. O *Throughput* da rede é então definido como a média do *Throughput* de todos os nós envolvidos na transmissão [14]. Dessa forma, o *Throughput* pode ser enunciado como:

$$\textit{Throughput de um nó} = (\textit{Total de bits recebidos} / \textit{tempo total de simulação}) \quad (4.2)$$

Da mesma maneira, o *Throughput* da rede é definido como:

$$\textit{Throughput da rede} = (\textit{Soma do Throughput de todos os nós envolvidos na transmissão} / \textit{número de nós}) \quad (4.3)$$

Outra métrica utilizada para a avaliação da rede é a *Delivery Ratio*, ou taxa de entrega. A taxa de entrega indica a porcentagem de pacotes de informação transmitidos que são recebidos com sucesso. É uma métrica importante que pode ser usada como um indicador da congestão na rede [14][16]. Para o cálculo dessa taxa, inicialmente é calculado o número total de pacotes transmitidos, seguido pelo cálculo do número total de pacotes recebidos e pelo a quantidade de pacotes perdidos [14][16]. A taxa de entrega ou *Delivery Ratio* pode ser enunciada como:

$$\textit{Delivery Ratio} = (\textit{Número de pacotes recebidos} / \textit{Número de Pacotes transmitidos}) \times 100. \quad (4.4)$$

A última métrica utilizada neste trabalho é a medida da porcentagem do consumo de energia de um nó com respeito a sua energia inicial. Para esta simulação foi definida como 14580 J a energia inicial de um nó sensor (Seção 4.1.4). Depois de a simulação ter sido completada é medida a energia final presente em cada nó. A porcentagem de energia consumida por um nó é então calculada como a razão da energia consumida pela energia inicial. Finalmente, a energia consumida por todos os nós é calculada como a média da energia consumida pelos nós individuais. A porcentagem de energia consumida e a energia média consumida podem ser definidas respectivamente como:

$$\text{Porcentagem da energia consumida por um nó} = [(Energia\ inicial\ do\ nó - Energia\ final\ presente\ no\ nó) / Energia\ inicial\ do\ nó] \times 100. \quad (4.5)$$

$$Energia\ média\ consumida = Soma\ da\ energia\ consumida\ por\ todos\ os\ nós / Número\ de\ Nós. \quad (4.6)$$

4.2.1 Cenário 1: Nonbeacon network

No primeiro cenário analisado o coordenador PAN da rede não está transmitindo beacons. Para simular essa situação foi definido como 15 o valor do parâmetro *macBeaconOrder* (BO) no algoritmo usado para a simulação. Neste cenário, assim como nos três cenários seguintes, há somente dois nós ativos inicialmente. Esse número é variado até que todos os nós da rede estejam ativados. As Figuras 4.3 e 4.4 ilustram respectivamente o comportamento dos valores do *Throughput* e da energia média consumida para a situação em que a rede não transmite *beacons* com a variação da quantidade de nós ativos.

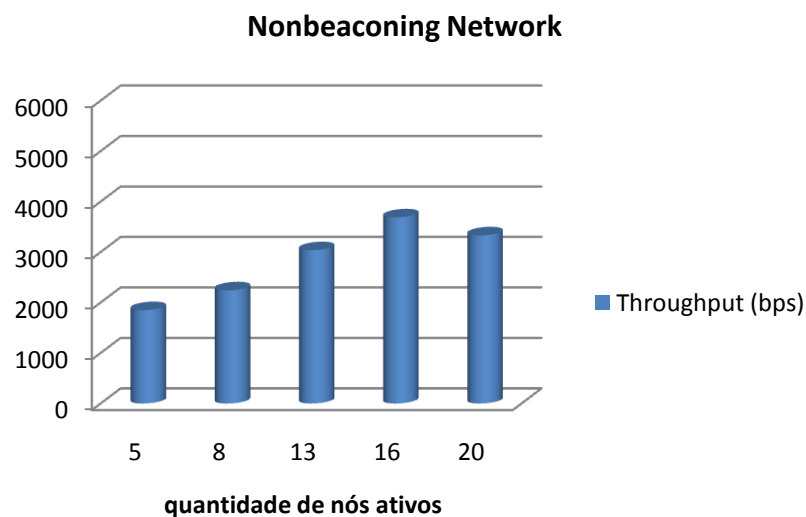


Figura 4.3: Variação do Throughput (em bits por segundo) com o aumento da quantidade de nós ativos para a rede sem beacons.

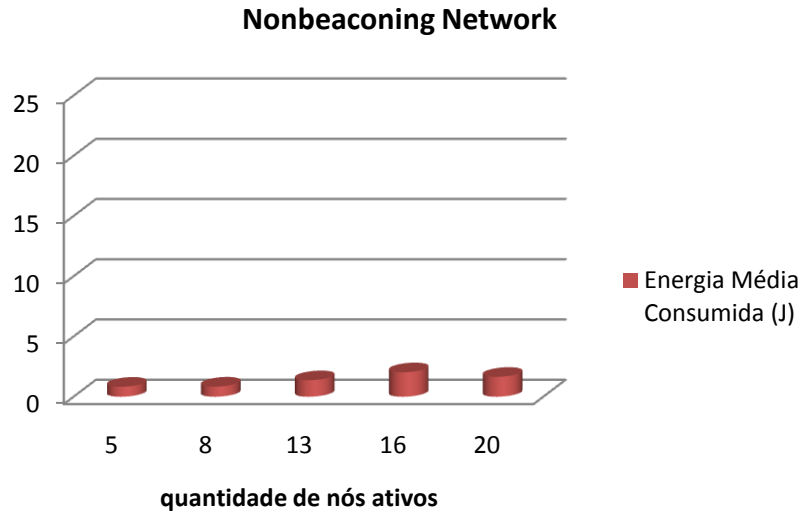


Figura 4.4: Variação do consumo médio de energia (em Joule) com o aumento da quantidade de nós ativos para a rede sem beacons.

4.2.2 Cenário 2: *beacon enabled network* com parâmetro $SO=BO=1$.

Neste cenário o coordenador PAN da rede está transmitindo sinais de beacons para sincronização dos nós da rede. Para este caso foi definido o valor 1 para os parâmetros *macBeaconOrder* (BO) e *macSuperframeOrder* (SO), que definem um período de 1920 para transmitir toda a estrutura do *superframe*:

$$SD = aBaseSuperframeDuration \times 2^{\text{macSuperframeOrder}}$$

$$BI = aBaseSuperframeDuration \times 2^{\text{macBeaconOrder}}$$

Neste caso, $SD = BI = 960 \text{ símbolos} \times 2^1 = 1920 \text{ símbolos}$.

Os gráficos 4.5 e 4.6 respectivamente ilustram como se comportam os valores do *Throughput*, e a energia média consumida para o caso em que a rede está operando no modo *beacon enabled* com *superframe* de duração de 1920 símbolos.

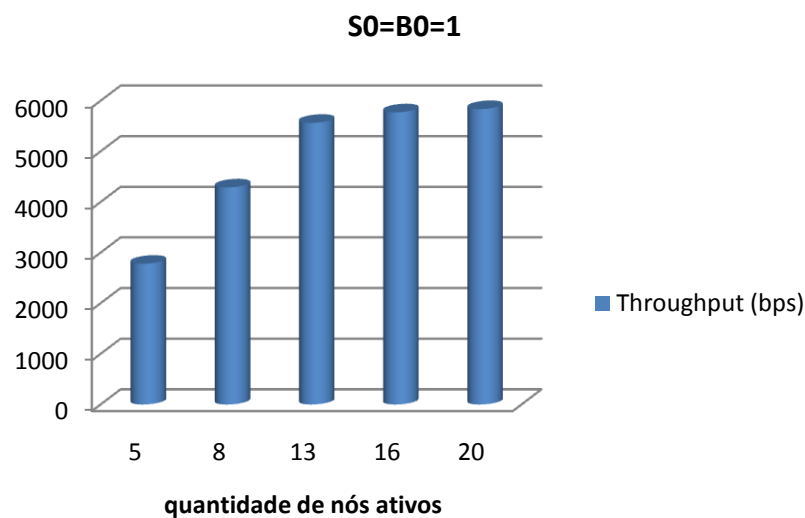


Figura 4.5: Variação do Throughput (em bits por segundo) com o aumento da quantidade de nós ativos para a rede com $BO=SO=1$.

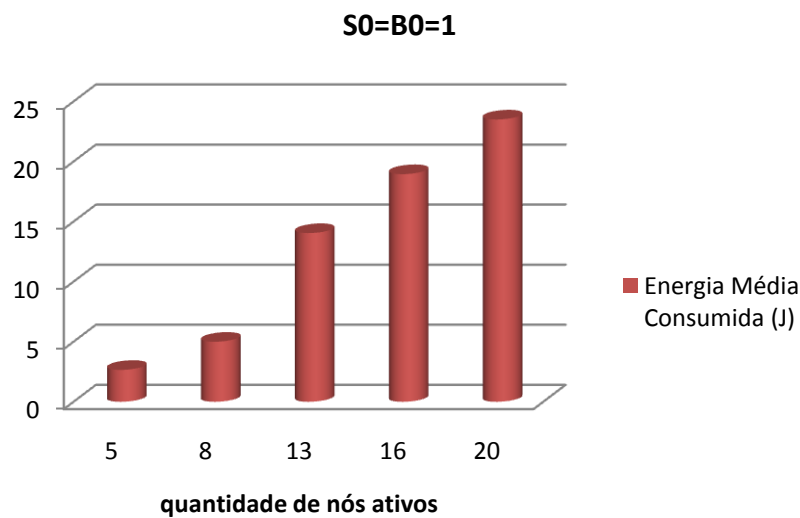


Figura 4.6: Variação do consumo médio de energia (em Joule) com o aumento da quantidade de nós ativos para a rede com $BO=SO=1$.

4.2.3 Cenário 3: *beacon enabled network* com parâmetro $SO=BO=3$

Neste cenário a rede também está operando em modo *beacon enabled*. Porém, para este caso foi definido um valor igual a 3 para os parâmetros *macBeaconOrder* (BO) e *macSuperframeOrder* (SO), que definem um período de 7680 símbolos para transmitir toda a estrutura do *superframe*:

$$SD = aBaseSuperframeDuration \times 2^{\text{macSuperframeOrder}}$$

$$BI = aBaseSuperframeDuration \times 2^{\text{macBeaconOrder}}$$

Neste caso, $SD = BI = 960 \text{ símbolos} \times 2^3 = 7680 \text{ símbolos}$.

Os gráficos 4.7, e 4.8 respectivamente ilustram como se comportam os valores do *Throughput*, *Delivery Ratio* e a energia média consumida para o caso em que a rede está operando no modo *beacon enabled* com *superframe* de duração de 7680 símbolos.

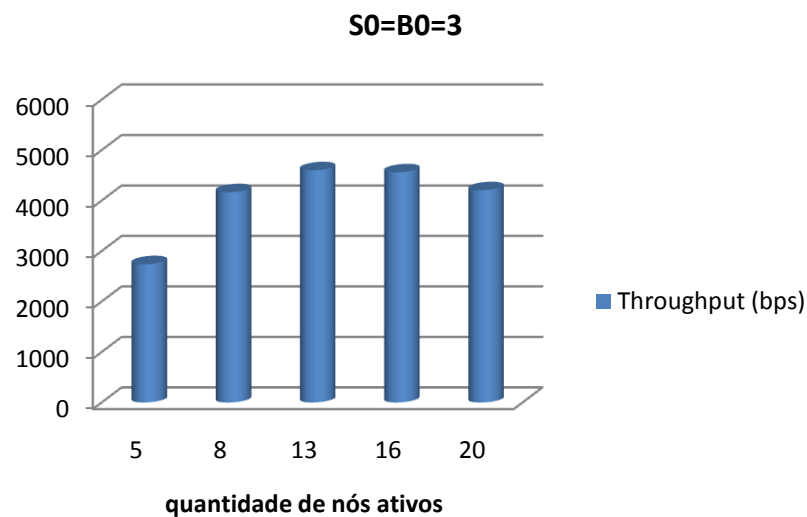


Figura 4.7: Variação do Throughput (em bits por segundo) com o aumento da quantidade de nós ativos para a rede com $BO=SO=3$.

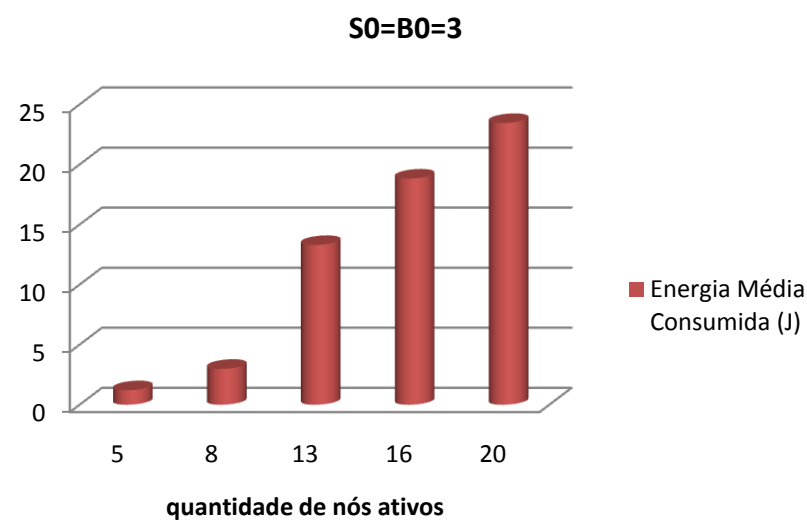


Figura 4.8: Variação do consumo médio de energia (em Joule) com o aumento da quantidade de nós ativos para a rede com $BO=SO=3$.

4.2.4 cenário 4: *beacon enabled network* com parâmetro $SO=BO=5$

Novamente, a rede também está operando em modo *beacon enabled*. Porém, nesse cenário, foi definido um valor igual a 5 para os parâmetros *macBeaconOrder* (BO) e *macSuperframeOrder* (SO), que definem um período de 30720 símbolos para transmitir toda a estrutura do *superframe*:

$$SD = aBaseSuperframeDuration \times 2^{\text{macSuperframeOrder}}$$

$$BI = aBaseSuperframeDuration \times 2^{\text{macBeaconOrder}}$$

Neste caso, $SD = BI = 960 \text{ símbolos} \times 2^5 = 30720 \text{ símbolos}$.

Os gráficos 4.9 e 4.10 respectivamente ilustram como se comportam os valores do *Throughput*, *Delivery Ratio* e a energia média consumida para o caso em que a rede está operando no modo *beacon enabled* com *superframe* de duração de 30720 símbolos.

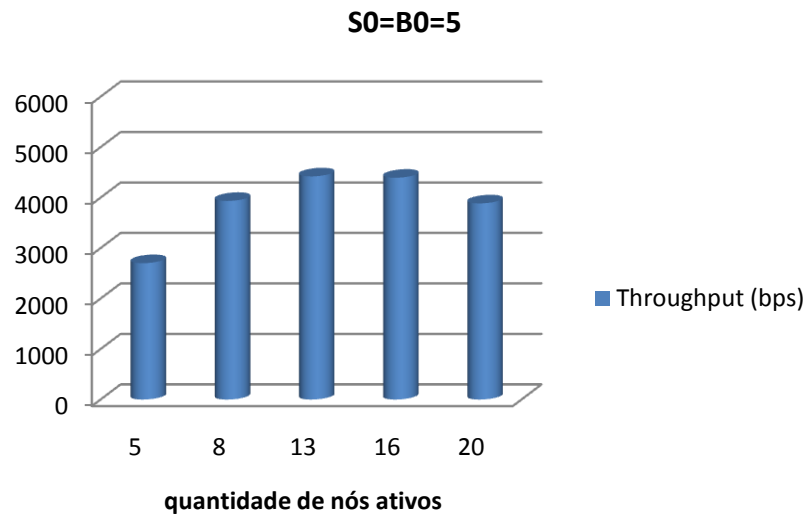


Figura 4.9: Variação do Throughput (em bits por segundo) com o aumento da quantidade de nós ativos para a rede com $BO=SO=5$.

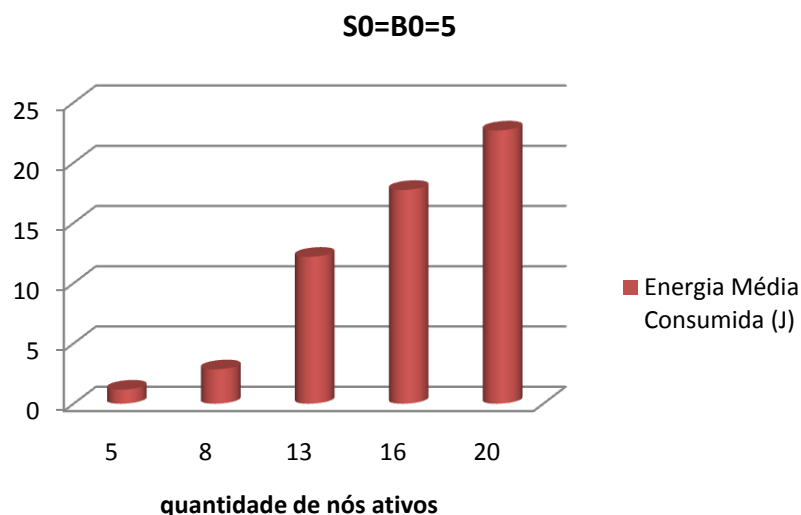


Figura 4.10: Variação do consumo médio de energia (em Joule) com o aumento da quantidade de nós ativos para a rede com $BO=SO=5$.

4.3 ANÁLISE E RESULTADOS

Aplicações de monitoramento e controle são umas das classes de aplicações mais utilizadas pelas redes de sensores sem fio que utilizam a pilha de protocolos do padrão IEEE 802.15.4. Nos cenários demonstrados nessa seção procurou-se analisar o desempenho de uma rede em que os sensores estão na borda de uma topologia estrela cujos dispositivos estão enviando os dados coletados para o nó coordenador central.

Os cenários e parâmetros apresentados foram escolhidos a fim de analisar de que forma a utilização ou não de *beacons* afeta o desempenho de uma rede de sensores que trabalha com um número pequeno de nós no que se refere ao gasto de energia, taxa de transmissão e congestionamento. Ou seja, a análise desses resultados procurou responder até que ponto a falta de sincronização afeta a performance de uma rede e, mais ainda, como a frequência com que as mensagens de sincronização são enviadas em redes que oferecem essa opção também afetam o funcionamento dessa rede.

Como pode ser observado no gráfico ilustrado na Figura 4.11, o aumento do número de nós ativos, em todos os cenários, provocou inicialmente uma elevação no *throughput* da rede. Este fato já era esperado pois mais nós estão enviando pacotes e conseqüentemente ocorre um aumento da taxa total de transmissão de dados e aumento do número de pacotes

recebidos por unidade de tempo. Porém, após um certo número de nós ativados, o valor do *throughput* começa a baixar. Isso ocorre devido ao aumento do tráfego na rede e conseqüentemente ao aumento do número de colisões que provocam perdas de pacotes. Pelo gráfico também pode se observar uma grande diferença no *throughput* entre redes no modo *beacon enabled* e redes que não utilizam esse mecanismo para sincronização. O *throughput* é muito mais elevado quando a rede está no modo *beacon enabled*, fato que também já era esperado. Sincronização é um fator chave para o *throughput* da rede. Todo dispositivo que quer transmitir deve competir pelo canal e para isso ele deve saber exatamente quando o período de contenção começa. Essa informação é dada justamente pela transmissão de mensagens de *beacons*. O mesmo fato ocorre quando o dispositivo quer transmitir de forma exclusiva ao ter acesso a um GTS ou *Guaranteed Time Slots*. Com a sincronização, o número de colisões e a perda de pacotes são reduzidos. Quanto menor a duração do *superframe*, ou seja, menor o intervalo entre *beacons*, maior é o controle sobre a rede e maior será o valor do *throughput*.

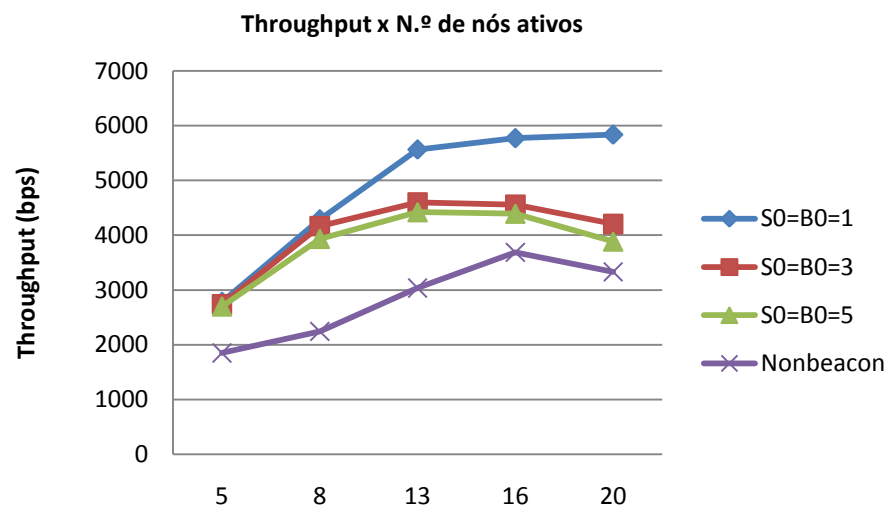


Figura 4.11: Comparação da variação do Throughput (em bits por segundo) para a rede operando nos Modo Beacon com diferentes tamanhos de superframe e sem Beacon.

A Figura 4.12 mostra que o aumento no número de nós ativos provocou uma redução no número médio de pacotes recebidos com sucesso, ou seja, provocou uma queda na taxa de entrega ou *delivery ratio*. Esse fato já era esperado pois quanto maior a quantidade de nós transmitindo dados maior será o número de colisões e a perda de pacotes. Observa-se também pelo gráfico uma grande diferença na taxa de entrega entre redes no modo *beacon enabled* e

nonbeacon networks. Nesta última, a falta de sincronização reduziu bruscamente a taxa de entrega. Em uma *Nonbeacon network*, os dispositivos não têm informações de sincronização nem sabem em que momento devem começar a transmitir para competir pelo canal. Dessa forma, essas redes que não transmitem *beacons* não podem ter acesso a um *slot* de transmissão exclusivo. Nesse caso, a rede não poderá ter períodos livres de contenção e sempre usará o mecanismo de CSMA-CA. Obviamente haverá um maior número de colisões na rede e maior será o número de pacotes perdidos. A taxa de entrega é maior em redes em modo *beacon enabled* e, assim como na análise feita para o *throughput*, quanto menor o intervalo entre *beacons*, ou seja, quanto menor o tamanho do *superframe*, maior será o valor da taxa de entrega. *Throughput* e *delivery ratio* estão intimamente relacionadas.

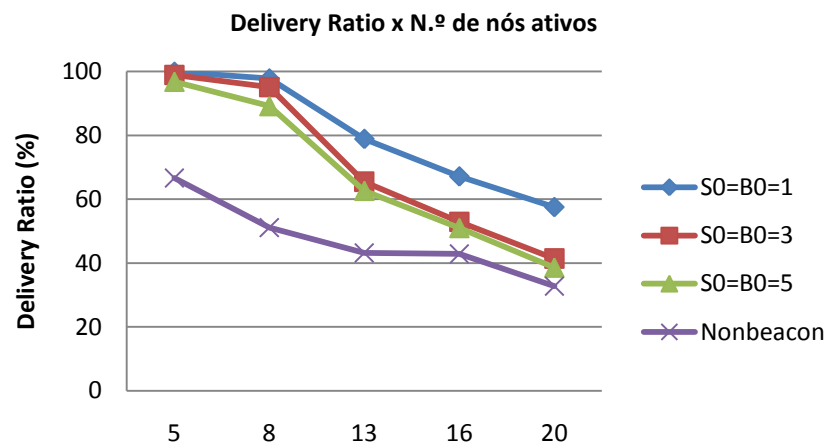


Figura 4.12: Comparação da variação da taxa de entrega (em porcentagem) para a rede operando nos Modos Beacon com diferentes tamanhos de superframe e sem Beacon.

As Figuras 4.11 e 4.12 ilustraram como a sincronização é importante para melhorar o *throughput* e a taxa de entrega da rede. Entretanto, essa vantagem é balanceada pelo aumento do consumo de energia. A Figura 4.13 mostra que a utilização da rede sem mecanismos de sincronização reduziu bruscamente a energia consumida pelos nós se comparado com a rede no modo *beacon enabled*. Nesse último, dispositivos na rede necessitam acordar freqüentemente para escutar as mensagens de *beacon* e fazer a sincronização, mesmo que não tenham que executar nenhuma tarefa enquanto estejam ativos. Ou seja, os dispositivos vão precisar estar ativos em alguns momentos apenas para sincronização. A figura também mostra

que a rede em modo *beacon enabled* com *superframes* com duração de símbolos reduzidas gastam mais energia. Isso deriva do fato de que a rede estará enviando *beacons* com uma frequência maior.

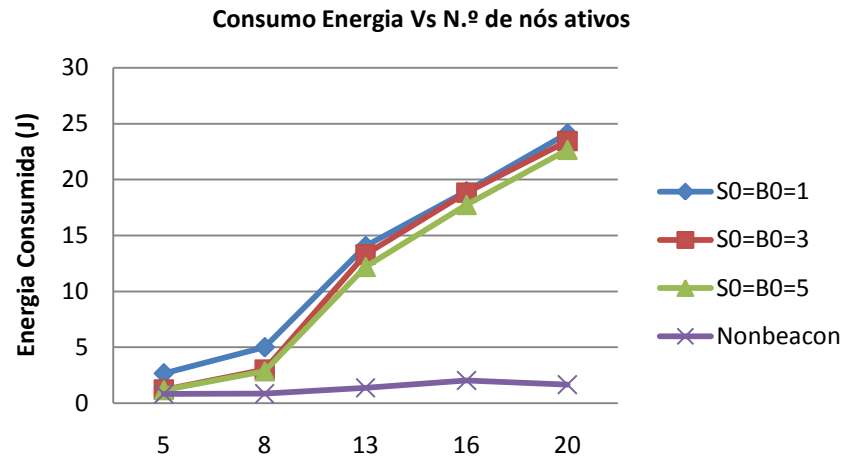


Figura 4.13: Comparação do consumo de energia médio (em Joule) para a rede operando nos Modos Beacon com diferentes tamanhos de superframe e sem Beacon.

5. CONCLUSÃO.

De acordo com os resultados obtidos nas simulações, foi possível observar que as Redes em modo *beacon-enabled* apresentam melhores valores de *throughput*, porém com a desvantagem de consumirem mais energia. Ao habilitar o modo *beacon-enabled* se está dando um maior controle sobre a rede, já que o coordenador PAN estará enviando mensagens de *beacons* constantemente a fim de sincronização. Entretanto, isso também significa que os dispositivos finais precisam entrar em modo ativo mais freqüentemente, mesmo que não tenham nenhuma outra tarefa a ser realizada além do processo de sincronização, daí o aumento do consumo de energia.

Uma forma de permitir maior controle sobre a rede e melhorias nas taxas de transferência, porém também atendendo a restrições de energia, é variar o tamanho do *superframe* em redes no modo *beacon-enabled*. Nesse caso, ao variar o tamanho do quadro de *superframe*, altera-se o intervalo de tempo em que as mensagens de *beacons* estão sendo enviadas e dessa forma pode-se fazer um balanceamento entre o gasto de energia e controle na rede. O processo de sincronização é um ponto chave para a melhoria da eficiência da rede. Em *nonbeacon networks*, em que não ocorre o processo de sincronização, observa-se uma degradação nas taxas de transferência e um aumento no número de colisões. Mesmo com o gasto de energia reduzido para este caso, pode não ser uma estratégia inteligente o uso dessas redes para muitas aplicações.

Outra forma de se reduzir a energia consumida pelas redes de sensores sem fio é através de melhorias nos projetos de hardware, diminuindo a potência consumida por esses dispositivos ou melhorando o armazenamento de energia. Felizmente, avanços significativos estão ocorrendo nessa área. Melhorias no projeto de sensores, através da implementação de novos conceitos e materiais está possibilitando a produção de dispositivos com custos e tamanhos reduzidos. As RSSF já são amplamente utilizadas em uma vasta gama de aplicações e segundo projeções espera-se que mais de meio bilhão de dispositivos serão entregues em 2010 para utilização em redes de sensores [19].

Apesar de todos esses avanços, não existe uma vasta bibliografia disponível para pesquisa nessa área. Ainda é limitado o número de livros que abordam as redes de sensores Zigbee e o padrão IEEE 802.15.4, e a maioria das obras encontradas são escritas com foco em um assunto específico relacionado com o tema. Além disso, também são poucos os códigos

ou algoritmos disponíveis para uso e simulações de redes de sensores do padrão IEEE 802.15.4. Essas falhas podem ser supridas por trabalhos futuros no campo das redes de sensores.

É válido destacar também que neste trabalho foram simuladas redes de sensores bastante limitadas, com um reduzido número de nós e baixa taxa de dados. Simulações futuras que expandam essas fronteiras e testem os limites das redes de sensores sem fio podem render trabalhos interessantes.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ARAUJO, Ingrid Palma. **Uma análise comparativa sobre o consumo de energia em redes de sensores sem fio utilizando o padrão IEEE 802.15.4 no modo de rede *beacon***. Salvador, 2009, 77 p. (Trabalho de Conclusão de Curso apresentado ao Centro Universitário da Bahia para graduação no curso de Sistemas de Informação).
- [2] CALLAWAY, Edgar H. **Wireless Sensor Networks: Architectures and Protocols**. Auerbach Publications. CRC Press. 2004
- [3] COMER, Douglas E.. **Interligação em Rede com TCP/IP: Princípios, Protocolos e Arquitetura**. 2. ed. Rio de Janeiro: Editora Campus, 1998.DAVIDSON, Jonathan;
- [4] DOWNARD, Ian T. **Simulating Sensor Networks in NS-2**. 2004. Disponível em <http://cs.itd.nrl.navy.mil/pubs/docs/nrlsensorsim04.pdf>. Acesso em: Outubro de 2009.
- [5] FARAHAANI, Shahin. **Zigbee Wireless Networking and Tranceivers**. Boston: Newnes, 2008.
- [6] GISLASON, Drewm. **Zigbee Wireless Networking**. Boston: Newnes, 2008.
- [7] HAENSELMANN, Thomas. **Sensor Networks**. 5th. [S.I.: s.n.], Apr. 2006. cap 1. HEILE, Bob. **The ZigBee Resource Guide is a publication of Webcom Communications Corporation**. Set. 2008.
- [8] IEEE, Standards - Electrical and Electronics Engineers. **802.15.4™- Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)**. Oct. 2003

- [9] IEEE, Standards - Electrical and Electronics Engineers. **IEEE 802.15 WPAN™ Task Group 4 (TG4)**. Sep. 2006. Disponível em: <http://www.ieee802.org/15/pub/TG4.html>, Acesso em: Setembro de 2009.
- [10] INFORMATION Science Institute, USC Viterbi School of Engineering. **The network Simulator -ns-2**. Disponível em: <http://www.isi.edu/nsnam/ns/>. Acessado em: Agosto de 2009.
- [11] JURDAK, R. **Wireless Ad-Hoc and sensor Networks**. New York: Springer, 2007.
- [12] KALR, Holger; WILLING Andreas. **Protocols and Architectures for Wireless Sensor Networks**. Chichester: John Wiley & Sons, 2005
- [13] LOUREIRO, Antonio A.F. **Redes de Sensores Sem Fio**. Departamento de Ciência da Computação. Universidade Federal de Minas Gerais, Belo Horizonte. 2006.
- [14] MARANDIN, Dimitri; RAO, Vaddina Prakash. **Simulation of IEEE 802.15.4/ZigBee with Network Simulator-2 (ns-2)**. Desdren. Disponível em: <http://www.ifn.et.tu-dresden.de/~marandin/ZigBee/ZigBeeSimulation.html>. Acesso em: Outubro de 2009.
- [15] PEREIRA, Marluce R; AMORIM, Claudio R; CASTRO, Maria C Stelling de. **Tutorial sobre redes de sensores**. Junho 2003. Disponível em: <http://www.ime.uerj.br/cadernos/cadinf/>. Acesso em: Outubro de 2009.
- [16] RAO, Vaddina Prakash. **Simulative Investigation of Zigbee/IEEE 802.15.4**. Desdren, 2006. Disponível em:
http://www.vaddina.com/pages/01_01_00_00_Zigbee/snav_1_0.php. Acesso em: Outubro de 2009.
- [17] RAPPAPORT, Theodore S. **Comunicações sem fio**. Tradução de Daniel Vieira. 2.ed. São Paulo: Prentice Hall, 2008.

- [18] SANTOS, Sergio Torres dos. **Redes de sensores sem fio em monitoramento e controle**. Departamento de Engenharia Elétrica. Universidade Federal do Rio de Janeiro. Junho. 2007. Dissertação de mestrado.
- [19] SOHRABY, Kazen; MINOLI, Daniel; ZNATI, Taieb. **Wireless sensor networks Technology, Protocols, and Applications**. Hoboken: John Wiley & Sons, 2007.
- [20] SRIPORAMANONT, Thammakit; LIMING, Gu. **Wireless Sensor Network Simulator**. School of Information Science, Computer and Electrical Engineering Halmstad University. Janeiro de 2006. Dissertação de Mestrado.
- [21] SWAMI, Ananthram et al. **Wireless Sensor Networks: Signal Processing and communications perspectives**. Chichester: John Wiley & Sons, 2007.
- [22] TEIXEIRA, Ingrid. **Roteamento com Balanceamento de Consumo de Energia para Redes de Sensores Sem Fio**. Departamento de Engenharia Elétrica. Universidade Federal do Rio de Janeiro. Abr. 2005. Dissertação de mestrado.
- [23] TILAK, S; ABU-GHAZALEH, N.B; HEINZELMAN, W. **A taxonomy of wireless micro-sensor network models**. ACM Workshop on Wireless Security, ACM Press, 2002.
- [24] THE VINT Project. **The NS Manual**. Disponível em: <http://www.isi.edu/nsnam/ns/ns-documentation.html>. Acesso em: Agosto de 2009.
- [25] ZIGBEE Alliance. **ZigBee Specification - Document 053474r17**. ZigBee Standards Organization. Jan. 2008.
- [26] ZHENG, Jianliang; LEE, Myung. J. **A Comprehensive Performance Study of IEEE 802.15.4**. Sensor Network Operations, IEEE Press Book, *Wiley InterScience*, cap. 4. 2006.

[27] ZHENG, Jianliang; LEE, Mying J. **Will IEEE 802.15.4 make ubiquitous networking a reality? - a discussion on a potential low power, low bit rate standard.** *IEEE Communications Magazine*, Vol. 42, No. 6, pp. 140-146, June 2004.

[28] http://www.duracell.com/oem/Pdf/new/1500_US_CT.pdf. Acesso em: Novembro de 2009.

[29] <http://www.rogercom.com/ZigBee/ZigBee.htm>. Acesso em: Novembro de 2009.

APÊNDICE

CÓDIGOS PARA O NS2

Os códigos e módulos Zigbee utilizados nesse projeto foram implementados por Vaddina Prakash Rao e podem ser encontrados para download nas bibliografias [14] e [16]. Esses algoritmos de Rao são adições e modificações feitas aos códigos desenvolvidos por Jianliang Zheng, da *The City University of New York* e do *Samsung Advanced Institute of Technology*, nos Estados Unidos. Os códigos de Zheng podem ser encontrados para download nas bibliografias [26] e [27].

Para alcançar os objetivos deste projeto, foram feitas mínimas modificações nos códigos, apenas para atender todos os parâmetros requeridos. Os códigos utilizados na simulação são expostos a seguir. Todos os códigos originais de Zheng e Rao, bem como as instruções de como eles podem ser utilizados se encontram nas bibliografias [14], [16], [26] e [27].

O primeiro código ilustrado, denominado *wpan868.tcl* contém as variáveis e parâmetros que ajudam a mudar o cenário da rede, bem como as ferramentas de controle da simulação. O utilitário *scen_gen.cpp* automaticamente gera as coordenadas dos nós e os aloca em volta de um círculo imaginário tendo o coordenador PAN como centro. O código *cbrgen_star.tcl* gera o tráfego aleatório entre os nós. Ele foi desenvolvido para uma rede em topologia estrela.

```
Star over 802.15.4 (beacon enabled)
Copyright (c) 2003 Samsung/CUNY
Prepared by Jianliang Zheng (zheng@ee.ccny.cuny.edu)
Modified By: Vaddina Prakash Rao
Chair of Telecommunications, TU Dresden
```

wpan868.tcl

```
=====
# Define options
#
=====
set val(chan)           Channel/WirelessChannel      ;# Channel Type
set val(prop)           Propagation/TwoRayGround     ;# radio-propagation
model
set val(netif)          Phy/WirelessPhy/802_15_4
set val(mac)            Mac/802_15_4
set val(ifq)            Queue/DropTail/PriQueue     ;# interface queue
type
```

```

set val(ll)                LL                ;# link layer type
set val(ant)               Antenna/OmniAntenna ;# antenna model
set val(ifqlen)            150              ;# max packet in ifq
set val(rp)                AODV             ;# routing protocol
set val(x)                 50
set val(y)                 50
set val(cp)                "traffic"        ;# point to the traffic characteristic
file
set val(dst)               3                ;# device start time(in
secs). The time at which the first device will be started.
set val(dstit)             1                ;# device start time
increment time (in secs). The time value after which the next device
will be started.

# Parameters used for traffic generation
# Do not use more than 25 nodes. Else modify the awk script file
analyzing the trace file, accordingly
# so that it can convert higher hex values to corresponding decimal
values.
set val(nn)                20               ;# nodechange:
number of mobilenodes
set val(starttime)         20.0
set stopTime               1000            ;# Non_Simulation
time
set BO                     15              ;# Non_Simulation
time
set SO                     15              ;# Non_Simulation
time

set infile [open "current_seed.tmp" r]
while {[gets $infile line] >= 0} {
set inputseed $line
}

set temp1 $val(nn)
incr temp1 -1
exec ./scen_gen $temp1 25 25 10

global defaultRNG
$defaultRNG seed $inputseed

set val(nam)               wpan.nam
#set val(traffic)          ftp              ;# cbr/poisson/ftp
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 0.0864
Antenna/OmniAntenna set Gt_ 1.0
Antenna/OmniAntenna set Gr_ 1.0

# Initialize Global Variables
set ns_                     [new Simulator]
set tracefd                [open ./wpan.tr w]
$ns_ trace-all $tracefd

```

```

# A few statements to indicate the simulator, about the NS animator.
if { "$val(nam)" == "wpan.nam" } {
    set namtrace [open ./$val(nam) w]
    $ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
}

$ns_ puts-nam-traceall {# nam4wpan #} ;# inform nam that this is
a trace file for wpan (special handling needed)

Mac/802_15_4 wpanCmd verbose on
Mac/802_15_4 wpanNam namStatus on ;# default = off (should be
turned on before other 'wpanNam' commands can work)
Mac/802_15_4 wpanNam ColFlashClr black ;# default = gold

# For model 'TwoRayGround'
Phy/WirelessPhy set CStresh_ 6.3096e-13
Phy/WirelessPhy set RXThresh_ 6.3096e-13
Phy/WirelessPhy set CPThresh_ 10

# The threshold values for the transmitter (CStresh_) and
receiver(RXThresh_). Values can be obtained from the "threshold"
utility.
Phy/WirelessPhy set Pt_ 0.001 ;# Transmitter power =
0.0456
Phy/WirelessPhy set freq_ 8.68e+08 ;# frequency of operation =
8.68e+08
Phy/WirelessPhy set L_ 1.0 ;# Path loss = 1.0

# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

# Create God
set god_ [create-god $val(nn)]

set chan_1_ [new $val(chan)]

# configure node

$ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace ON \
    -movementTrace ON \
    -energyModel "EnergyModel" \
    -initialEnergy 14580 \
    -rxPower 0.0625 \
    -txPower 0.0750 \

```

```

        -idlePower 0.00000625 \
        -channel $chan_1_

for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0           ;# disable random motion
}

source wpan.scn

# nodechange
$ns_ at 0.0 "$node_(0) NodeLabel PAN Coord"
$ns_ at 0.0 "$node_(0) sscs startPANCoord 1 $BO $SO"           ;#
startPANCoord <txBeacon=1> <BO=3> <SO=3>

# Loop to start all the other nodes, with the first node starting at
val(dst) and the following nodes to
# start at times, incremented by val(dstit), to the starting time of
the previous node.
set i 1
while {$i < $val(nn)} {
    $ns_ at $val(dst) "$node_($i) sscs startDevice 0 0 0 $BO $SO" ;#
    startDevice <isFFD=1> <assoPermit=1> <txBeacon=0> <BO=3> <SO=3>
    incr val(dst) $val(dstit)
    incr i 1
}

Mac/802_15_4 wpanNam PlaybackRate 3ms

$ns_ at $val(starttime) "puts \"\\nTransmitting data ...\\n\""

if { $val(cp) == "" } {
    puts "*** NOTE: no connection pattern specified."
    set val(cp) "none"
} else {
    puts "Loading connection pattern..."
    source $val(cp)
}

# defines the node size in nam
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 2
}

# Tell nodes when the simulation ends
for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $stopTime "$node_($i) reset";
}

$ns_ at $stopTime "stop"
$ns_ at $stopTime "puts \"NS EXITING...\\n\""
$ns_ at $stopTime "$ns_ halt"

proc stop {} {
    global ns_ tracefd val(starttime) val env
    $ns_ flush-trace

```

```

close $tracefd
set hasDISPLAY 0
foreach index [array names env] {
    #puts "$index: $env($index)"
    if { (" $index" == "DISPLAY") && (" $env($index)" != "") } {
        set hasDISPLAY 1
    }
}
# if { (" $val(nam)" == "wpan.nam") && (" $hasDISPLAY" == "1") } {
#     exec /root/ns-2/ns-allinone-2.28/nam-1.11/nam wpan.nam &
#     exec kate ~/work/ns-allinone-2.33/ns-2.33/wpan_test/wpan.tr &
#     exec cp wpan.tr /root/downloads/scripts/awk/ &
#     exec awk -f avg_throughput.awk wpan.tr
# }
# }

puts "\nStarting Simulation..."
$ns_ run

```

The copyright for the original script lies with the following author.
copyright (c) 1999 by the University of Southern California
All rights reserved.

Modified by:
Vaddina Prakash Rao
Chair of Telecommunications
TU Dresden

cbrgen_star.tcl

```

=====
# Default Script Options
#
=====
# The following default values are used when not provided

set opt(nn)      0          ;# Number of Nodes
set opt(seed)    0.0        ;# Random number generator seed value
set opt(mc)      0          ;# Number of sources
set opt(pktsize) 70         ;# Packet size

set opt(rate)    0          ;# Datarate
set opt(interval) 0.0       ;# inverse of rate
set opt(type)    ""         ;# Traffic type
set opt(starttime) 20.0     ;#Traffic start time
set opt(timegap) 5.0        ;# Default traffic gap time
#array set sources {}
#
=====

```

```

proc usage {} {
    global argv0

    puts "\nusage: $argv0 \[-type cbr|tcp\] \[-nn nodes\] \[-seed
seed\] \[-mc connections\] \[-rate rate\] \[-starttime st\] \[-timegap
tg\]\n"
}

proc getopt {argc argv} {
    global opt
    lappend optlist nn seed mc rate type starttime timegap

    for {set i 0} {$i < $argc} {incr i} {
        set arg [lindex $argv $i]
        if {[string range $arg 0 0] != "-"} continue

        set name [string range $arg 1 end]
        set opt($name) [lindex $argv [expr $i+1]]
    }
}

proc create-cbr-connection { src dst stime } {
    global rng cbr_cnt opt

    # The following line specifies the interval within which the flows will
    start.
    #set stime [$rng uniform 0.0 10.0]

    puts "#\n# $src connecting to $dst at time $stime\n#"

    ##puts "set cbr_($cbr_cnt) \[\$ns_ create-connection \
    ##CBR \$node_($src) CBR \$node_($dst) 0\]";
    puts "set udp_($cbr_cnt) \[new Agent/UDP\]"
    puts "\$ns_ attach-agent \$node_($src) \$udp_($cbr_cnt)"
    puts "set null_($cbr_cnt) \[new Agent/Null\]"
    puts "\$ns_ attach-agent \$node_($dst) \$null_($cbr_cnt)"
    puts "set cbr_($cbr_cnt) \[new Application/Traffic/CBR\]"
    puts "\$cbr_($cbr_cnt) set packetSize_ $opt(pktsize)"
    puts "\$cbr_($cbr_cnt) set interval_ $opt(interval)"
    puts "\$cbr_($cbr_cnt) set random_ 1"
    puts "\$cbr_($cbr_cnt) set maxpkts_ 10000"
    puts "\$cbr_($cbr_cnt) attach-agent \$udp_($cbr_cnt)"
    puts "\$ns_ connect \$udp_($cbr_cnt) \$null_($cbr_cnt)"

    puts "\$ns_ at $stime \"\$cbr_($cbr_cnt) start\""

    incr cbr_cnt
}

proc create-tcp-connection { src dst stime } {
    global rng cbr_cnt opt

    #set stime [$rng uniform 0.0 180.0]

    puts "#\n# $src connecting to $dst at time $stime\n#"

    puts "set tcp_($cbr_cnt) \[\$ns_ create-connection \

```

```

        TCP \${node_}($src) TCPSink \${node_}($dst) 0\];
puts "\${tcp_}($cbr_cnt) set window_ 32"
puts "\${tcp_}($cbr_cnt) set packetSize_ $opt(pktsize)"

puts "set ftp_($cbr_cnt) \[\${tcp_}($cbr_cnt) attach-source FTP\]"

puts "\${ns_} at $stime \[\${ftp_}($cbr_cnt) start\]"

incr cbr_cnt
}

#
=====

getopt $argc $argv

if { $opt(type) == "" } {
    usage
    exit
} elseif { $opt(type) == "cbr" } {
    if { $opt(nn) == 0 || $opt(seed) == 0.0 || $opt(mc) == 0 ||
$opt(rate) == 0 } {
        usage
        exit
    }

    set opt(interval) [expr 1 / $opt(rate)]
    if { $opt(interval) <= 0.0 } {
        puts "\ninvalid sending rate $opt(rate)\n"
        exit
    }
}

puts "#\n# nodes: $opt(nn), max conn: $opt(mc), send rate:
$opt(interval), seed: $opt(seed)\n#"

set rng [new RNG]
$rng seed $opt(seed)

set u [new RandomVariable/Uniform]
$u set min_ 0
$u set max_ $opt(nn)
$u use-rng $rng

set cbr_cnt 0
set src_cnt 0
set flag 0
set loopcount $opt(nn)

for {set i 0} {$i < $loopcount} {incr i} {

    while {1} {
        set x [$u value]
        set x [expr $x/0.01]

```

```

    if { $x < $opt(nn) } {
        set src [expr round($x)]
        break
    }
}

if { $src >= $opt(nn) } {
    set src [expr $opt(nn)-1]
}

# Add the generated source to the list unless we already had the
source before
if {[array size sources] == 0} {
    set sources(0) $src
} else {
    # Comparing each source in the list to the currently generated
source, to see if there is a match
    for {set p 0} {$p < [array size sources]} {incr p 1} {
        if {$src == $sources($p)} {
            set flag 1
            break
        }
    }

    if {$src == 0} {
        set flag 1
    }

    # If there is a match, break from the current loop (discard
the source)
    if {$flag == 1} {
        set flag 0
        incr loopcount 1
        continue
    } else {
        set size [array size sources]
        set sources($size) $src
    }
}

# We always want the destination to be node-0.
set dst 0

if { $opt(type) == "cbr" } {
    incr src_cnt 1
    create-cbr-connection $src $dst $opt(starttime)
} else {
    create-tcp-connection $src $dst $opt(starttime)
}

if { $src_cnt == $opt(mc) } {
    # You have created the required number of sources. So break now !!
    break
} else {
    # The required number of sources have not been reached yet, even
though it is highly unlikely.
    set temp $loopcount

```



```

        incr temp -1
        if {$i == $temp} {
            incr loopcount 1
        }
    }

    set opt(starttime) [expr $opt(starttime)+$opt(timegap)]
}

puts "#\n#Total sources: $src_cnt\n#"

```

scen_gen.cpp

```

#include <iostream>
#include <fstream>
#include <math.h>
#include <stdlib.h>
using namespace std;

#ifndef EXIT_FAILURE
#define EXIT_FAILURE 1
#endif

#define RADIANS 6.285714286

double add(int, double);

int main(int argc, char *argv[])
{
    ofstream out("wpan.scn", ios::trunc);
    int ql=0, NUM_OF_NODES, X, Y, POS;
    double x_pos, y_pos, sector_ang, dPOS;

    if (!out)
    {
        cout << "Error Opening file, wpan.scn. The program will now
terminate" << endl;
        exit(EXIT_FAILURE);
    }

    if (argc<5 || argc>5)
    {
        cout << "Error !!! Insufficient or excess parameters." <<
endl;
        cout << "Usage:" << endl;
        cout << "./scen_gen <number-of-nodes> <X-Position-of-Coord>
<Y-Position-of-Coord> <Personal-Operating-Space>" << endl;
        cout << "\tnumber-of-nodes: Number of nodes to be arranged
around the coordinator. Currently the program only works when the
number of nodes to be arranged around the coordinator is divisible by
4." << endl;
        cout << "\tX-Position-of-Coord: The x-ordinate of the
coordinator" << endl;
    }
}

```

```

        cout << "\tY-Position-of-Coord: The y-ordinate of the
coordinator" << endl;
        cout << "\tPersonal-Operating-Space: The POS of a node." <<
endl;

        cout << "Example: ./scen_gen 11 25 25 10" << endl;
        exit(EXIT_FAILURE);
    }
    else
    {
        NUM_OF_NODES = atoi(*(argv+1));
        X = atoi(*(argv+2));
        Y = atoi(*(argv+3));
        POS = atoi(*(argv+4));
        dPOS = POS;
        sector_ang = RADIANS/NUM_OF_NODES;

        // Entering the position for node-0.
        out << "$node_(0) set X_ " << X << endl;
        out << "$node_(0) set Y_ " << Y << endl;
        out << "$node_(0) set Z_ 0" << endl << endl;
    }

    //cout << cos(45.00) << sin(45.00) << endl;

    for (int i=0; i<NUM_OF_NODES; i++)
    {
        x_pos = dPOS * cos(sector_ang*i);
        y_pos = dPOS * sin(sector_ang*i);

        out << "$node_(" << i+1 << ") set X_ " << add(X, x_pos) <<
endl;
        out << "$node_(" << i+1 << ") set Y_ " << add(Y, y_pos) <<
endl;
        out << "$node_(" << i+1 << ") set Z_ 0" << endl << endl;
    }
    out.close();
    return 0;
}

// Function to add an integer and a double and return the double value.
double add(int a1, double b1)
{
    return a1+b1;
}

```