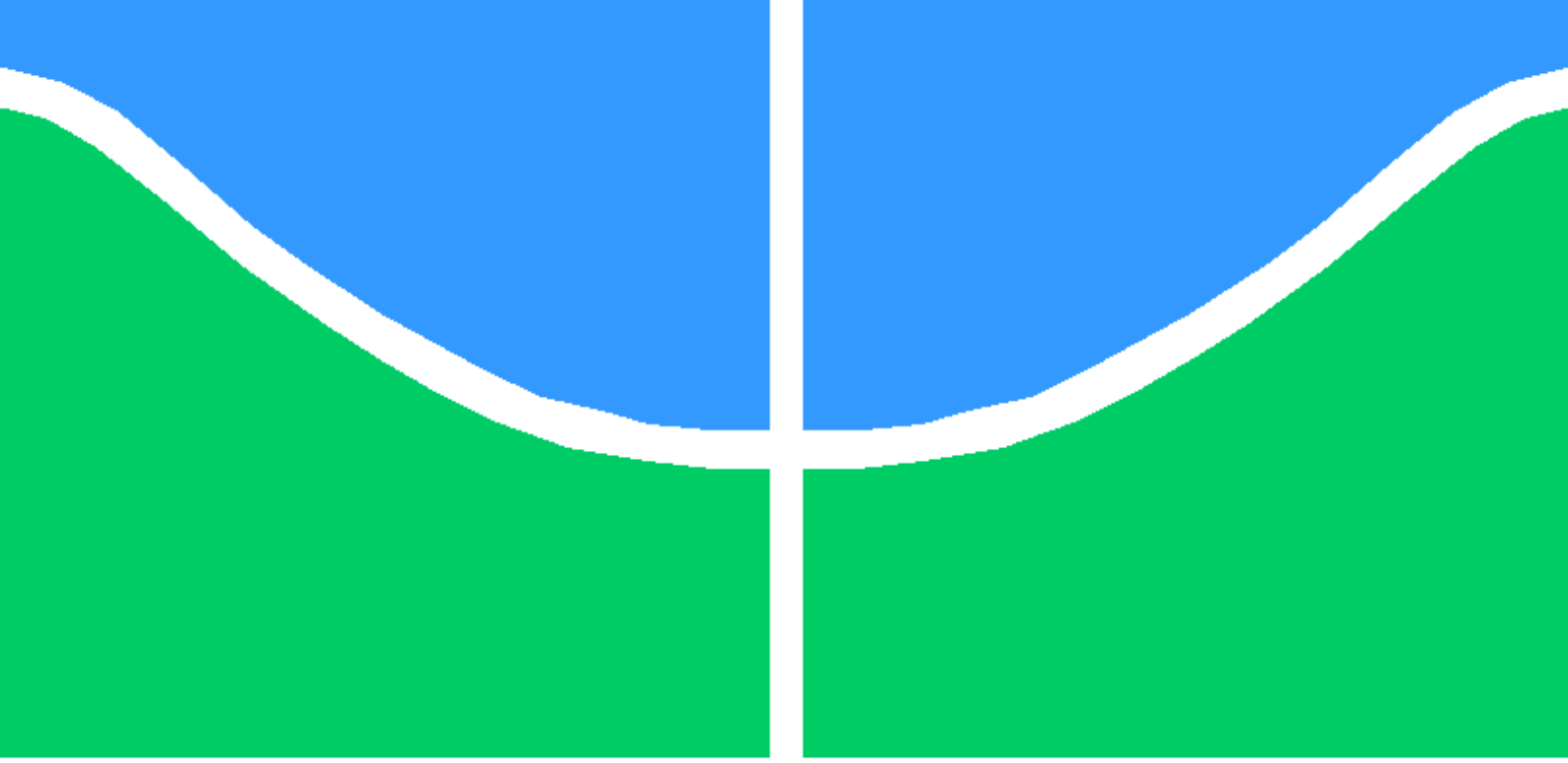

TRABALHO DE GRADUAÇÃO

ANÁLISE E EXPLORAÇÃO DAS VULNERABILIDADES DA TECNOLOGIA ZIGBEE EM AMBIENTES IOT

Bruna de Abreu Silva

Raquel Magalhães de Souza

Brasília, Dezembro de 2018

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

ANÁLISE E EXPLORAÇÃO DAS VULNERABILIDADES DA TECNOLOGIA ZIGBEE EM AMBIENTES IOT

Bruna de Abreu Silva

Raquel Magalhães de Souza

Relatório submetido ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Engenheiras de Redes de Comunicação

Banca Examinadora

Prof. Georges Daniel Amvame Nze, Dr., ENE/UnB _____
Orientador

Fábio Lúcio Lopes de Mendonça, MSc., ENE/UnB _____
Examinador Interno

Prof. Rafael Timóteo de Sousa Jr., Dr., ENE/UnB _____
Examinador Interno

Porque, assim como os céus são mais altos do que a terra, assim são os meus caminhos são mais altos do que os vossos caminhos, e os meus pensamentos, mais altos do que os vossos pensamentos.

Isaías 55:9

Agradecimentos

Agradecemos, primeiramente, a Deus que foi nosso alicerce e suporte para realização deste trabalho, dando-nos, diariamente, a força para seguir em frente.

Ao Professor Georges Daniel Amvame Nze que desempenhou o papel fundamental de orientador e, com palavras sábias e críticas valiosas, transmitiu-nos seus conhecimentos acadêmicos e pessoais. Ele sempre esteve à disposição para apoiar, incentivar e compartilhar experiências que nos engrandeceram.

Agradecemos, imensamente, aos nossos pais, Diandrade Francelino e Maria Hélia, Edimar e Zeneide, e, também, às nossas irmãs, Bruna, Beatriz e Reyslane, que são nossa base familiar, dando-nos o incentivo necessário desde o início da educação até o ensino superior do curso de Engenharia de Redes de Comunicação.

Agradecemos, especialmente, ao Francisco, ao aluno Caio Maia e aos nossos colegas da empresa, Telebrás, que nos ajudaram dando-nos apoio, auxílio e suporte para a realização deste trabalho.

Aos colegas do Laboratório UIoT da Universidade de Brasília que nos disponibilizaram o ambiente e informações necessárias para a realização deste trabalho. O Laboratório UIoT conta com recursos de apoio das Agências brasileiras de pesquisa, desenvolvimento e inovação CAPES (Projeto FORTE 23038.007604/2014-69), CNPq (Projeto INCT em Segurança Cibernética 465741/2014-2) e Fundação de Apoio à Pesquisa do Distrito Federal FAPDF (Projetos UIoT 0193.001366/2016 e SSDDC 0193.001365/2016), bem como do Gabinete de Segurança Institucional da Presidência da República (TED 002/2017) e do Laboratório LATITUDE/UnB (Projeto SDN 23106.099441/2016-43).

Agradecemos, também, aos nossos amigos de curso que nos acompanharam ao longo da caminhada na Universidade de Brasília e foram capazes de transformar o cotidiano de estudo.

Bruna de Abreu Silva e Raquel Magalhães de Souza

RESUMO

Tendo em vista o grande progresso das redes IoT (*Internet of Things*), cada vez mais presentes nos diferentes setores da sociedade, considera-se que faz parte da sua evolução também o desenvolvimento dos métodos de segurança que assegurem sua proteção, garantindo integridade, autenticidade e privacidade. Por esse motivo, o presente trabalho apresenta um estudo realizado no Laboratório UIoT do curso de Engenharia de Redes de Comunicação, localizado na Universidade de Brasília, com o objetivo de analisar as medidas de segurança tomadas. Considerando o aumento da utilização do protocolo ZigBee em redes IoT, devido à sua simplicidade e baixo custo, este projeto se propôs analisar as vulnerabilidades deste protocolo a partir da maneira como foi realizada no ambiente em questão e diante dos cenários apresentados. Além disso, considera-se ainda, a existência de uma rede ZigBee já implementada no Laboratório UIoT. Com o auxílio de um dispositivo *sniffer, USB Dongle CC2531*, em conjunto com um software desenvolvido com a finalidade de gravar as informações do tráfego de redes que usam o protocolo ZigBee, será possível coletar informações de todo o funcionamento da rede. A partir das informações obtidas, o projeto será dividido em quatro cenários, com o intuito de simplificar o estudo. Diante das informações obtidas, propõe-se um método de monitoração, a fim de facilitar a detecção de intrusos na rede e acelerar o processo para solucionar cada caso, auxiliando o gerente na tomada de decisão. Aliado a essa proposta, recomenda-se e implementa-se a utilização da criptografia de dados a fim de assegurar a confidencialidade dos dados transmitidos e evitar a conexão de dispositivos intrusos na rede. Por fim, conclui-se que o novo modelo de monitoramento da rede Zigbee no *Laboratório UIoT* associado à aplicação de criptografia dos dados, permitiu a garantia de uma maior segurança para a rede implementada, tendo em vista que agora, o monitoramento ocorre via camada MAC, e a inserção de um dispositivo externo é dificultada, considerando a obrigatoriedade de se ter uma chave de segurança da rede para conectar-se ao coordenador.

Palavras-chave: *Internet das Coisas, ZigBee, Segurança de redes, Sniffer, XBee.*

ABSTRACT

In view of the great progress of the IoT (Internet of Things) networks, which are increasingly present in the different sectors of society, it is considered that their development also includes the development of security methods to ensure their protection, guaranteeing integrity, authenticity and privacy. For this reason, the present work presents a study carried out at the UIoT Laboratory of the Communication Networks Engineering course, located at the University of Brasília, in order to analyze the safety measures taken. Considering the increase in the use of the ZigBee protocol in IoT networks, due to its simplicity and low cost, this project proposed to analyze the vulnerabilities of this protocol based on the way it was done in the environment in question and in the scenarios presented. In addition, it is also considered the existence of a ZigBee network already implemented in the UIoT Laboratory. With the aid of a sniffer device, USB Dongle CC2531, together with software developed for the purpose of recording network traffic information using the ZigBee protocol, it will be possible to collect information about the entire network operation. Based on the information obtained, the project will be divided into four scenarios, in order to simplify the study. Given the information obtained, a monitoring method is proposed in order to facilitate the detection of intruders in the network and accelerate the process to solve each case, assisting the manager in decision-making. In addition to this proposal, the use of data encryption is recommended and implemented in order to ensure the confidentiality of transmitted data and to avoid the connection of intrusive devices in the network. Finally, it is concluded that the new Zigbee network monitoring model in the textit UIoT Lab associated with the application of data encryption, allowed the guarantee of a greater security for the implemented network, since now the monitoring occurs via the MAC layer, and the insertion of an external device is difficult, considering the obligation to have a network security key to connect to the coordinator.

Keywords: *Internet of Things, ZigBee, Network Security, Sniffer, XBee.*

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

Acrônimos

ABNT	Associação Brasileira de Normas Técnicas
ACK	*Acknowledgement*
ACL	*Access Control List*
AES	*Advanced Encryption Standard*
API	*Application Programming Interface*
APL	Camada de aplicação
APP	Camada de aplicação
APS	*Application Support*
BER	*Bit Error Rate*
BI	*Business Intelligence*
BSN	*Beacon Sequence Number*
CCM = CBC-MAC	*Counter with Cipher Block Chaining Message Authentication Code*
CRC	*Cyclic Redundancy Check*
CSMA-CA	*Carrier Sense Multiple Access with Collision Avoidance*
DDoS	*Distributed Denial of Service*
DH	*Destination Address High*
DHT11	Sensor de umidade e temperatura
DL	*Destination Address Low*
DNS	*Domain Name System*
DoS	*Denial of Service*
DSN	*Sequence Number*
DSSS	*Direct Sequence Spread Spectrum*
EE	*Encyption Enable*
FCF	*Frame Control*
FFD	*Full Function Device*
FIPS	*Federal Information Processing Standard*
FPGA	*Field Programmable Gate Array*
FSC	*Frame Check Sum*
GHz	*Gigahertz*

Acrônimos

GPRS	*General Packet Radio Services*
HDMI	*High-Definition Multimedia Interface*
IBSG	*Internet Business Solutions Group*
IDE	*Integrated Development Environment*
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IoT	Internet das Coisas
IrDA	*Infrared Data Association*
ISM	*Industrial, Scientific and Medical*
ITU-T	*International Telecommunication Union for Telecommunications*
KY	*Encryption Key*
MAC	*Media Access Control*
MHz	*Megahertz*
MIC	*Message Integrity Code*
MPDU	*Media Access Control Protocol Data Unit*
MTR	Meteorologia
NI	*Node Identifier*
NTP	*Network Time Protocol*
NWK	Camada de rede
O-QPSK	*Offset Quadrature Phase-shift Keying*
PAN ID	*Personal Area Network Identifier*
PCB	*Printed Circuit Board*
PCs	*Personal Computers*
PDAs	Assistentes Digitais Pessoais
PDU	*Protocol Data Unit*
PHY	Camada Física
PSD	*Power Spectral Density*
PY	*Python*
QoS	Qualidade de Serviço
RAM	*Random Access Memory*
RFD	*Reduced Function Device*
RPSMA	*Reverse-Polarity SubMiniature version A*
RX	Recepção
SC	*Scan Channels*
SD	*Secure Digital*
SH	*Serial Number High*
SKKE	*Symmetric-Key Key Establishment Protocol*

Acrônimos

SKY	*Symmetric-Key*
SL	*Serial Number Low*
SoC	*System-on-chip*
TC	Centro de confiança
TEMP	Temperatura
TX	Transmissão
UIoT	*Universal Internet of Things*
UnB	Universidade de Brasília
USB	*Universal Serial Bus*
USB HID e CDC	*USB Human Interface Devices e Communication Device Class*
UWB	*Ultrawideband*
VBA	*Visual Basic for Applications*
VDC	Tensão em Corrente Contínua
WPAN	*Wireless Personal Area Networks*
ZB	*ZigBee*
ZC	*ZigBee* Coordenador
ZDO	*ZigBee Device Object*
ZED	*ZigBee* Dispositivo Final
ZR	*ZigBee* Roteador

Capítulo 1

Introdução

As redes sem fio estão evoluindo de forma cada vez mais rápida, um termo que tem se tornado muito utilizado quando se fala em tecnologia é a sigla IoT (*Internet of Things*), que significa Internet das Coisas. Trata-se de uma rede de dispositivos que possuem tecnologia embarcada, de forma que podem coletar e transmitir dados, mesmo que essa não seja sua principal função, como os veículos, por exemplo. A IoT é considerada a próxima evolução da Internet (EVANS, 2011), considerando o impacto que pode ter assim como a Internet, na educação, nos negócios, na comunicação.

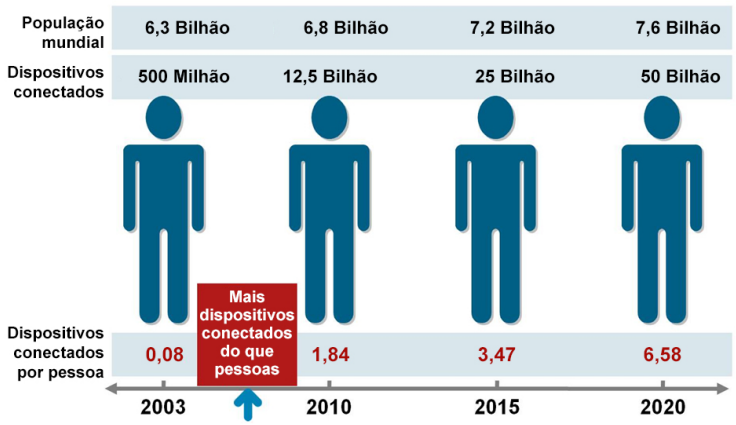

Figura 1.1: Estimativa da quantidade de dispositivos conectados por pessoa no decorrer dos anos. Fonte: (EVANS, 2011).

A partir de um estudo relacionando a população mundial com a quantidade de dispositivos conectados por pessoa, a Cisco IBSG estimou o ano em que teria iniciado a IoT, indicando que seria entre 2008 e 2009, pois foi a partir desse ano que calculou-se a existência de mais dispositivos conectados do que pessoas. Conforme pode ser visualizado na Figura 1.1, estima-se que o número de dispositivos conectados por pessoa nos próximos anos deve aumentar significativamente, esse é um motivo pelo qual é importante o estudo sobre esse tema, principalmente em relação aos seus mecanismos de segurança.

Dentre os protocolos utilizados nesse tipo de dispositivos está o ZigBee, um protocolo definido pelo padrão IEEE 802.15.4, muito útil para uma rede de sensores pois possui baixa potência, o que implica em baixo custo, e por isso é uma boa opção em situações mais simples, isto é, aplicações que não exigem processamento de dados ou cálculos complexos, por exemplo, rede de sensores e automação residencial.

1.1 Definição do Problema

É analiticamente evidente e comprovado que ataques causam grandes prejuízos ao seu alvo, onde as vulnerabilidades são um extenso problema recorrente atualmente, em que os usuários que necessitam dos aparelhos sem fio, buscam cada vez mais medidas para erradicar e diminuir os ataques pertinentes e aumentar a segurança no tráfego dos dados.

Há ainda, a ocorrência de vazamento de informações pessoais em larga escala, obtendo um significativo aumento nos últimos anos, o que põe em risco a privacidade e a segurança dos indivíduos afetados.

Naturalmente, vulnerabilidades em diferentes protocolos estarão sempre presentes, no entanto o ZigBee foi escolhido para ser trabalhado nesse estudo pois é um dos protocolos mais presentes em redes IoT, além disso, alguns estudos e pesquisas apontam um crescimento constante em seu uso para os próximos anos. A figura 1.2 apresenta uma estimativa desse crescimento, calculada pela *ABI Research* e *BI Intelligence* em 2014.

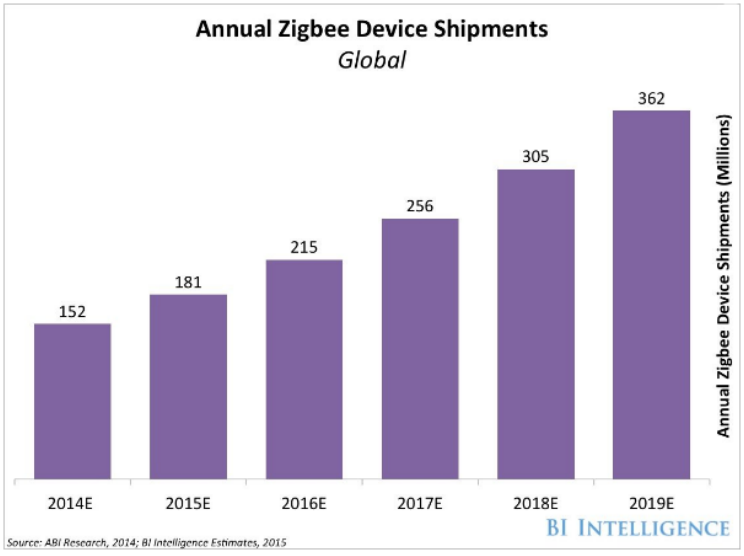

Figura 1.2: Estimativa da quantidade de dispositivos embarcados utilizando ZigBee no decorrer dos anos. Fonte: (CAMHI, 2016).

Dentre os dados mencionados acima, alguns questionamentos podem ser feitos, como: A quais riscos um usuário desta rede pode estar exposto? Como evitá-los? Ou como impedi-los? Como eliminar essas vulnerabilidades? Estas são algumas das perguntas que este trabalho se propõe a responder.

1.2 Objetivo

Este projeto tem por objetivo geral o estudo e a análise detalhada da tecnologia Zigbee, dando ênfase às suas vulnerabilidades e ataques. Devido às limitações decorrentes deste simples protocolo, alguns requisitos como segurança e privacidade são fatores relevantes a serem discutidos e abordados dentro do escopo deste projeto, uma vez que a garantia da integridade dos dados é um fator primordial que assegura a correta entrega das informações entre a origem e o destino evitando assim, que estes sofram quaisquer modificações.

1.2.1 Objetivos específicos

Para alcançar este objetivo, os seguintes Objetivos Específicos são propostos: Dentre os objetivos específicos, tem-se:

- Análise detalhada da tecnologia estudada;
- Simular e identificar ataques pertinentes na aplicação do protocolo ZigBee;
- Propor soluções para minimizar os ataques e consequentemente aumentar a segurança do mesmo;
- Detectar e avaliar a eficiência do modelo de análise dos ataques propostos, com intuito de garantir a correta utilização do protocolo afim de manter seu modelo simplista.
- Propor um método de monitoração, a fim de facilitar a detecção de intrusos na rede e acelerar o processo de solução para cada caso.

1.3 Revisão da Literatura

A base da bibliografia referenciada neste trabalho levou em conta a busca por artigos teses, monografias e livro em diversas fontes, especialmente, a UnB (Universidade de Brasília) e IEEE (*Institute of Electrical and Electronic Engineers*). Além disso foram realizadas pesquisas em bases de dados dos organismos de normatização ISO (*Internacional Organization for Standardization*) e ABNT (Associação Brasileira de Normas Técnicas). Foram realizadas pesquisas na base bibliográfica da UnB, que é uma instituição de renome. Para esta busca, foram utilizados os trabalhos do aluno Caio Maia (MAIA, 2017) e das alunas Catharina Daher e Mariana Clarim (TEIXEIRA; LACERDA CLARIM, 2017), como referência e continuação dos projetos e pesquisas feitas por estes alunos.

A ideia deste projeto tem como base a utilização da topologia Zigbee construída pelo aluno Caio Maia no laboratório de UIoT, localizado na Universidade de Brasília. A junção da topologia criada juntamente com, pesquisas, análises e aplicação de conceitos empregados neste trabalho em questão, deu-se continuidade ao projeto das alunas Catharina e Mariana, que visa enfatizar as vulnerabilidades das redes sem fio.

Apesar das pesquisas dos alunos citados acima apresentarem trabalhos relevantes, nenhum deles aborda a análise proposta neste projeto. No entanto, foi encontrado uma tese com o tema e proposta bastante semelhante ao que será evidenciado neste trabalho. A partir da tese feita por Charbel Azzi (AZZI, 2016), da *University of Nevada, Las Vegas* pôde-se obter um estudo aprofundado sendo fundamental para a realização deste trabalho.

1.4 Estrutura do Trabalho

Este trabalho será organizado entre os próximos capítulos em quatro partes principais: fundamentação teórica, proposta, análise de resultados e conclusões.

O Capítulo 2 apresenta os principais conceitos e características que compõem este trabalho, bem como o funcionamento detalhado do protocolo em estudo, como níveis de segurança, ataques, topologia da rede utilizada, entre outros.

O Capítulo 3 apresenta a arquitetura, softwares e metodologia dos estudos conduzidos. Essa seção tem como finalidade promover uma visão detalhada da estrutura de rede utilizada, de forma que a mesma possa ser reproduzida posteriormente em outros trabalhos, a fim de atender a outras propostas de estudo.

O Capítulo 4 mostra de forma detalhada o desenvolvimento do trabalho e uma análise dos resultados obtidos em cada cenário considerado.

O Capítulo 5 contém a discussão final do estudo e conclui o trabalho, apresentando possíveis trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo visa, explicar o funcionamento do protocolo em estudo, assim como todos os conceitos mencionados na realização prática das simulações.

2.1 Internet das Coisas

A Internet das Coisas emergiu a partir dos avanços de diferentes áreas como sistemas embarcados, microeletrônica, comunicação e sensoriamento. Por se tratar de uma tecnologia emergente, ainda não há um conceito único que possa definir de forma completa o termo Internet das Coisas, no entanto, considerando seus aspectos qualitativos é possível descrever como uma extensão da Internet atual (SANTOS et al., 2016), pois permite que objetos que fazem parte do cotidiano possam conectar-se à Internet e se comunicarem entre si, a fim de facilitar e beneficiar as pessoas em suas atividades diárias. Por outro lado, considerando uma abordagem mais objetiva em função de aspectos quantitativos, de acordo com o *Cisco Internet Business Solutions Group* - IBSG, a IoT começou no momento exato em que foram conectados à Internet mais objetos do que pessoas (EVANS, 2011).

Apesar das divergências entre as definições já expostas, o que todas têm em comum é a ideia de que a primeira versão da Internet era sobre dados criados por pessoas, enquanto a próxima versão é sobre dados criados por coisas. Desta forma, a melhor definição para a Internet das Coisas seria:

"Uma rede aberta e abrangente de objetos inteligentes que têm a capacidade de se auto-organizar, compartilhar informações, dados e recursos, reagindo e agindo diante de situações e mudanças no ambiente" (MADAKAM; RAMASWAMY; TRIPATHI, 2015)

Diante desses conceitos, é possível destacar sua importância em uma sociedade de constante desenvolvimento, pelo fato de cumprir com a função de facilitar o dia-a-dia das pessoas, a IoT possui aplicações em diferentes áreas de atuação, na área médica, de segurança, em redes domésticas e empresariais. Nesse sentido, é possível fazer uma pequena análise segundo o Gartner Hype Cycle.

O Gartner Hype Cycles fornece uma representação gráfica da maturidade e adoção de tecnologias e aplicativos, e como eles são potencialmente relevantes para resolver problemas reais de

negócios e explorar novas oportunidades. A metodologia Gartner Hype Cycle oferece uma visão de como uma tecnologia ou aplicativo evoluirá ao longo do tempo, fornecendo uma fonte de informações para gerenciar sua implantação dentro do contexto de suas metas de negócios específicas e proporcionando assim um critério de decisão importante na respetiva aplicabilidade das mesmas.

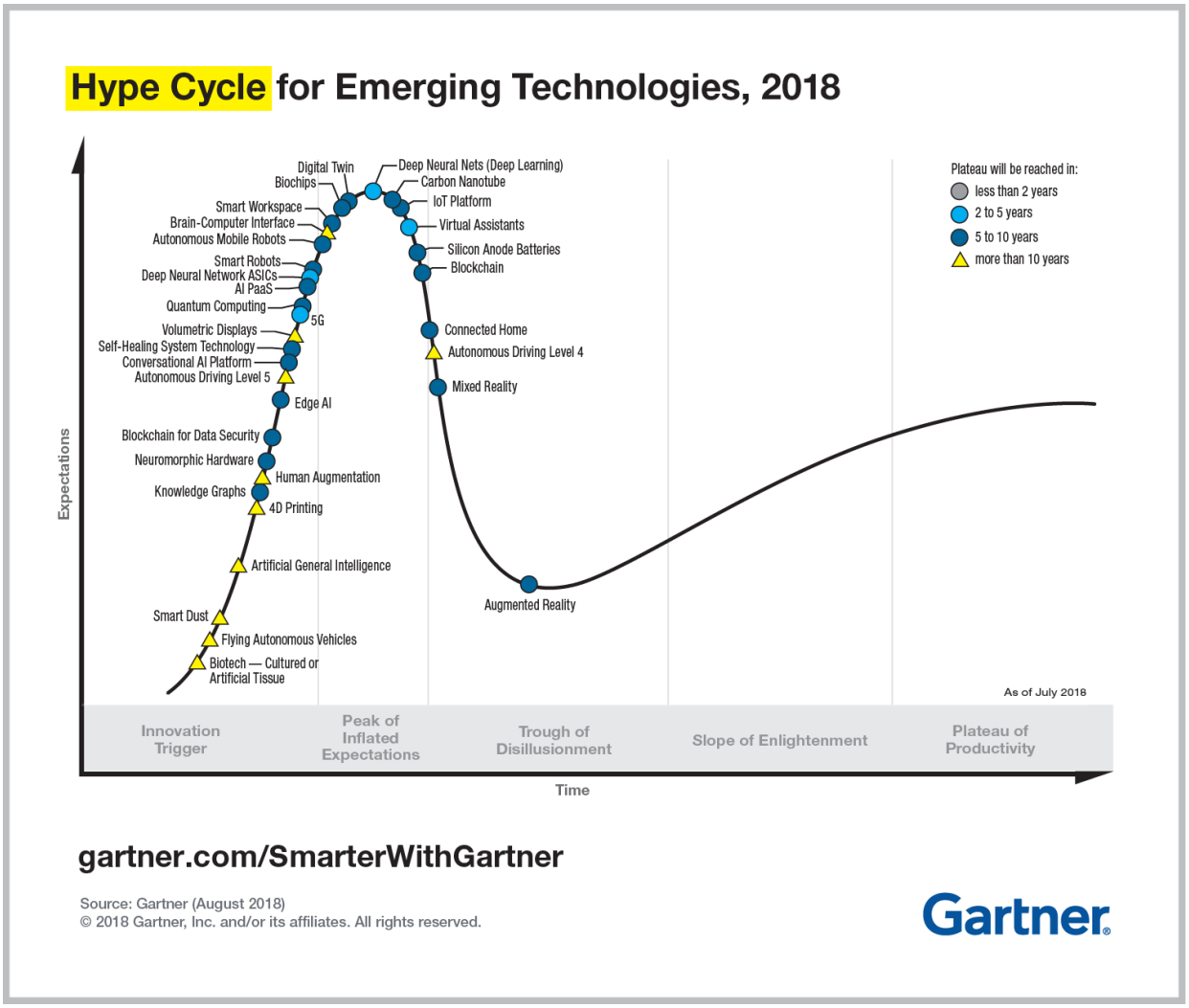

Figura 2.1: Posição das tecnologias emergentes no ano de 2018 segundo a metodologia Gartner Hype Cycle. Fonte: (PANETTA, 2018)

Devido à ampla variedade de aplicações que pode se empregar a IoT, este termo tornou-se em poucos anos um assunto muito falado na área de tecnologia da informação e segundo o diagrama Gartner Hype Cycle 2018 da Figura 2.1, a IoT está ainda na fase *Peak of Inflated Expectations*, que é a fase em que surgem notícias de algumas histórias de sucesso em sua utilização, mas também aparecem casos de falhas. Começa então a experimentação por parte de algumas organizações chamados "*early adopters*", pois ainda é uma fase de testes, verificação das funcionalidades dessa nova tecnologia (GARTNER, 2018).

De acordo com esta metodologia, no ano de 2018, o conceito de plataforma IoT deve chegar à fase de produção (*Plateau of Productivity*) nos próximos 5 a 10 anos. Esta é a fase em que a tecnologia deixa de ser apenas nova ideia que está na moda. Nessa momento, os critérios para

avaliar a viabilidade do provedor são mais claramente definidos e a ampla aplicabilidade e relevância do mercado da tecnologia estão trazendo lucros (GARTNER, 2018).

A IoT pode ser vista como a combinação de várias tecnologias, as quais são complementares quando se trata de viabilizar a integração dos objetos no ambiente físico ao mundo virtual. A Figura 2.2 apresenta os blocos básicos de construção da IoT sendo eles (SANTOS et al., 2016):

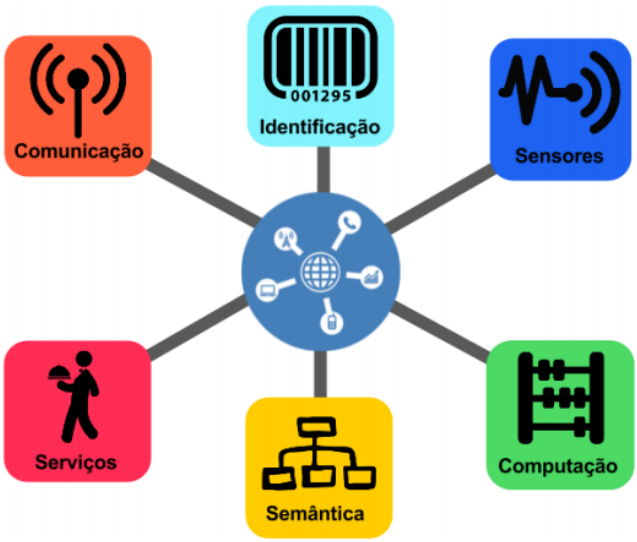

Figura 2.2: Blocos básicos da Internet das Coisas. Fonte: (SANTOS et al., 2016)

- **Identificação**: cada objeto deve ser identificado unicamente para conectar-se à Internet.
- **Sensores/atuadores**: sensores servem para coletar informações sobre o contexto onde estão inseridos. Atuadores, por sua vez, podem modificar o ambiente com suas ações, eles recebem comandos e reagem de acordo com os dados lidos.
- **Comunicação**: com o objetivo de conectar objetos inteligentes, diferentes técnicas podem ser utilizadas.
- **Computação**: compreende a unidade de processamento como microcontroladores, processadores e FPGAs, responsáveis por executar algoritmos locais nos objetos inteligentes.
- **Serviços**: a IoT pode oferecer serviços em diferentes classes, entres as principais, estão os serviços de identificação, agregação de dados e colaboração e inteligência.
- **Semântica**: diz respeito à capacidade de extração de conhecimento dos objetos na IoT. Refere-se ao uso eficiente dos recursos existentes na IoT, a partir dos dados existentes, com o objetivo de prover determinado serviço.

Portanto, a Internet das Coisas é considerada hoje como uma tecnologia emergente e que está sendo testada e explorada em tudo o que pode oferecer, por esse motivo é ainda uma área muito ampla e de constante desenvolvimento.

2.2 ZigBee

2.2.1 Conceito

O ZigBee (ZIGBEE ALLIANCE, 2012) é um padrão de comunicação wireless de baixo custo e baixo consumo de energia, desenvolvido pela ZigBee Alliance para redes pessoais sem fio, WPANs (Wireless Personal Area Networks). As soluções que adotam o padrão ZigBee são geralmente automação residencial e predial, controles industriais, periféricos de PC, aplicativos de sensores médicos, brinquedos e jogos. A ZigBee Alliance consiste em uma união de diversas empresas incluindo multinacionais, universidades, empresas públicas e startups.

A primeira versão da especificação ZigBee foi lançada em 2004: hoje em dia essa versão pode ser considerada obsoleta e, portanto, não é mais suportada em novos dispositivos ZigBee. A segunda versão da especificação ZigBee foi lançada em 2006: esta versão é usada quando a rede ZigBee deve ser a mais barata possível. A versão ZigBee Pro, também conhecida como ZigBee 2007, foi lançada em 2007 e é usada quando o tamanho da rede ZigBee é muito grande e recursos de segurança aprimorados são necessários para proteger a rede. A versão mais recente deste protocolo, o ZigBee 3.0 (ZIGBEE ALLIANCE, 2018), é feita a partir do ZigBeePro que aprimora o padrão IEEE 802.15.4 adicionando camadas de rede e segurança em malha junto com um framework de aplicação e é uma solução Zigbee interoperável, full stack, com baixo consumo de energia(ZIGBEE ALLIANCE, 2012)(VIDGREN et al., 2013).

O principal propósito do ZigBee é a automação residencial, como sistemas de sensores de chuva, luz, fumaça, fechaduras e janelas. ZigBee opera na frequência de 2,4 GHz suportando taxa de dados teórica de 250 kbps, na frequência de 915 MHz suportando taxas de dados teóricas de 40 kbps e 250 kbps, e na frequência de 868 MHz suportando taxas de dados teóricas de 20 kbps, 100 kbps e 250 kbps. O ZigBee é baseado na técnica DSSS (Direct Sequence Spread Spectrum - Espectro de Espalhamento de Sequência Direta) e na modulação O-QPSK (Offset Quadrature Phase-shift Keying)(ZIGBEE ALLIANCE, 2012)(VIDGREN et al., 2013).

2.2.2 Características

O esforço do 802.15 WPAN se concentra no desenvolvimento de padrões consensuais para redes de área pessoal ou redes sem fio de curta distância. Esses WPANs abordam a rede sem fio de dispositivos de computação portáteis e móveis, como PCs, Assistentes Digitais Pessoais (PDAs), periféricos, telefones celulares e eletrônicos de consumo; permitindo que esses dispositivos se comuniquem e interajam uns com os outros (IEEE 802.15, 2018).

Os dispositivos baseados na tecnologia ZigBee operam na faixa ISM que não requer licença para funcionamento, incluindo as faixas de 2,4GHz (Global), 915Mhz (EUA e Austrália) e 868Mhz (Europa) e com taxas de transferência de dados de 250kbps em 2,4GHz, 40kbps em 915Mhz e 20kbps em 868Mhz (PINHEIRO, 2004).

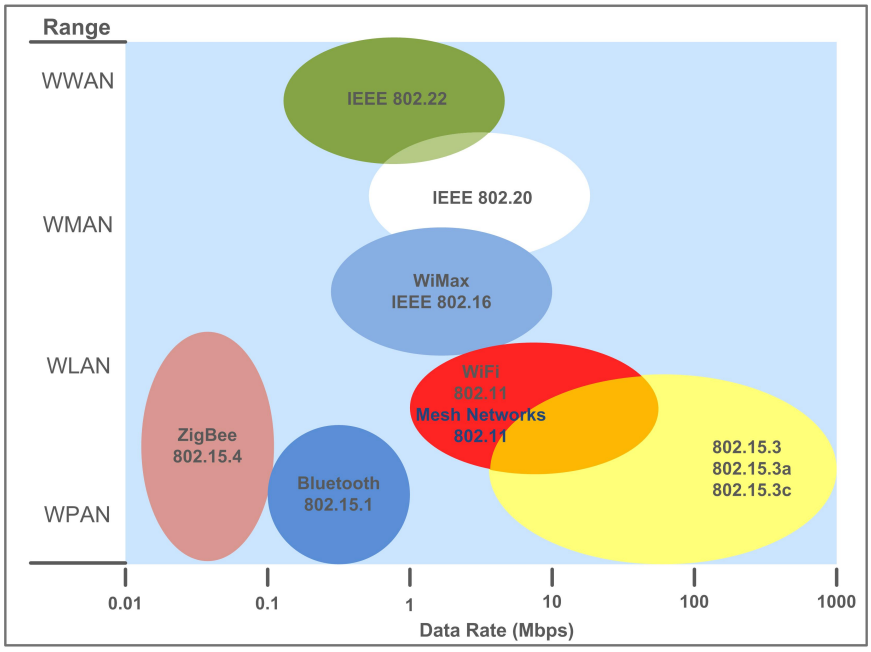

Figura 2.3: Comparação do ZigBee com outras tecnologias wireless. Fonte: (ROWEBOTS, 2015).

O padrão atualmente oferece velocidades de conexão entre 10kbps e 115kbps e uma transferência de transmissão entre 10m e 100m, dependendo das características de energia e ambientais do equipamento (barreiras físicas, interferência eletromagnética, etc). No que diz respeito ao problema dos dispositivos de alimentação, os módulos de controle dotados com esta nova tecnologia podem ser alimentados até mesmo por baterias (pilhas) comuns, sendo que sua vida útil está relacionada diretamente com a capacidade da bateria e a a implementação pretendida. A este respeito, o protocolo ZigBee é projetado para suportar aplicativos com consumo mínimo (para baterias comuns, um dispositivo pode durar até 6 meses) (PINHEIRO, 2004).

O padrão ZigBee (IEEE 802.15.4) foi projetado com as seguintes características (PINHEIRO, 2004):

- Baixo consumo de energia e fácil operação com interfaces de baixo custo;
- Dois principais estados de funcionamento: "active"para transmissão e recepção e "sleep", quando não está transmitindo;
- Simplicidade de configuração e redundância de dispositivos (operação segura);
- Alta densidade de nós por rede. As camdas PHY e MAC permitem que as redes trabalhem com um grande número de dispositivos ativos. Esse recurso é crucial para aplicativos com sensores e redes de controle;
- Protocolo simples que permite a transferência de dados confiáveis para níveis de segurança apropriados. Velocidades de dados de 250 kbps, 40 kbps e 20 kbps.
- Dois modos de endereçamento; Endereçamento IEEE de 64 bits e de 16 bits.

- Acesso ao canal CSMA-CA.
- Estabelecimento automático de rede pelo coordenador.
- Protocolo totalmente "apertado"para confiabilidade de transferência.
- Gerenciamento de energia para garantir baixo consumo de energia.
- 16 canais na banda ISM de 2,4 GHz, 10 canais no 915MHz e um canal na banda 868MHz.

2.2.3 Aplicações

Existem vários aplicativos que podem usar os recursos do ZigBee. O que vale ressaltar é, que qualquer aplicativo que precise de uma bateria de longa duração, pequenos circuitos, e baixo custo de taxa de transferência, podem ser implementados nesta tecnologia. Ou seja, de forma prática, o ZigBee é de grande valia em ambientes não cabeados, situações em que a rede for utilizada para experimentos temporários, e quando se deseja ter um baixo custo de manutenção para o projeto requerido.

Figura 2.4: Setores de aplicação da tecnologia ZigBee. Fonte: (FRARE; ARAKI; XAVIER, 2009).

Alguns exemplos de aplicações são descritos abaixo, de maneira a facilitar o entendimento das vantagens a se utilizar o ZigBee (VASQUES et al., 2010).

Automação de Controles Domésticos

Devido ao baixo preço, o ZigBee se apresenta como uma promissora tecnologia para automação e gerenciamento do sistema residencial a um preço acessível para o cliente. O controle de múltiplos sistemas operacionais pode ser feito através de uma única rede, com os recursos robustos da topologia de malha, que também permite a centralização desses sistemas em um controle remoto.

Existem vários sistemas domésticos que podem ser integrados com este sistema de automação, como: luz, calor, ventilação, ar condicionado, computadores, televisão, sensores de água, sensores elétricos, sensores de gás, detectores de fumaça, painéis de alarme, detectores de movimento, e outros dispositivos eletrônicos. A implementação de tal sistema pode garantir dados precisos sobre o consumo de água, gás e eletricidade e pode permitir a implementação de inteligência incorporada no sistema para otimizar o uso desses recursos.

Aplicações na Área de Saúde

Um dos usos da tecnologia ZigBee no campo da saúde atualmente em estudo é o monitoramento do paciente. Esse monitoramento pode ser dividido em:

Episódio: monitorização periódica, usada para pacientes não críticos, para rastrear indicadores específicos (sinais vitais, pressão arterial, níveis de glicose) para identificar o andamento da doença. Nesse caso, um registro de data e hora (timestamp) é adicionado às informações coletadas e enviadas para um banco de dados, onde elas serão armazenadas para consultas futuras.

Contínuo: usado em pacientes que requerem análise constante das condições de saúde. As formas de ondas dos sinais vitais são transferidos para uma unidade de coleta de dados no corpo do paciente para ser feita a junção de dados e/ou armazenamento sequencial. Esta informação é então enviada para um computador onde os sensores são configurados, armazenados e analisados.

Alarme: usado para sinalizar se o paciente atinge determinadas condições predeterminadas. Nesse cenário, as informações são armazenadas da mesma forma que o monitoramento episódico,porém, garantem a frequência mínima de erro de bit (BER) e a latência mínima já que a monitoração é contínua e, se as condições especificadas forem atendidas, um alarme será acionado.

Manutenção da Rede Elétrica

Atualmente, a manutenção de postes de luz e a revisão dos relógios medidores residenciais são feitos por um técnico que vai ao local para realizar as medições. No caso do poste, o técnico precisa subir para fazer a avaliação do problema. Essa tarefa é contrária à política de segurança do trabalho, cada vez mais presente em grandes empresas e com leis cada vez mais exigentes. No caso dos relógios, é indesejável que os técnicos precisem de permissão para acessar a casa e fazer a medição. É notável que ocorra um gasto de tempo desnecessário, para que o acesso à propriedade seja negociado. O objetivo do projeto é a instalação do aparelho Zigbee em tal equipamento. Os dispositivos do Zigbee Router seriam instalados em cada poste da rede, que se comunicaria com a central elétrica por meio de saltos, e os terminais Zigbee seriam instalados no medidor residencial.

Utilização nos postes

O uso de um projeto em uma rede elétrica é necessário quando um ponto específico da rede elétrica para de funcionar ou por algum motivo não estivesse funcionando como deveria. Neste caso, o painel de controle envia um sinal para descobrir em qual ponto exato ele sai da comunicação. Se esse ponto for detectado, o técnico será enviado para verificar o problema e fazer o reparo.

Durante este projeto, não será necessário que o técnico suba no poste para verificar o defeito. Através de um dispositivo PDA ou outro dispositivo de avaliação, após a instalação do hardware Zigbee, o técnico sincroniza com a rede e executa os testes necessários para detectar anomalias.

Utilização nos relógios residenciais

No caso dos relógios medidores residenciais, os técnicos podem acessar o relógio do consumidor e fazer a checagem do mesmo para a verificação a pedido do consumidor ou da própria central.

A instalação dessa tecnologia também aumenta a segurança do consumidor, pois este não corre o risco de que um indivíduo mal intencionado se faça passar por um técnico da companhia de energia. Mesmo para uma empresa de eletricidade, há uma vantagem, pois não há o risco de que os técnicos cheguem ao local e não possam fazer medições devido à ausência de qualquer ocupante.

2.3 Arquitetura ZigBee

2.3.1 Pilha ZigBee

No padrão 802.15.4, a estrutura geral da PDU do nível MAC (MPDU) foi projetada para ser flexível o bastante para acomodar as necessidades de diferentes aplicações e topologias de rede e, ao mesmo tempo, permitir a definição de um protocolo de nível MAC relativamente simples. O MPDU é composto por um cabeçalho (*MAC Header*), por uma Unidade de Serviço de Dados (MAC Service Data Unit) também referenciada de *MAC Payload*, e pelo *Frame Check Sum* (FILHO, 2015).

O cabeçalho contém informações sobre o tipo de quadro (a norma define quatro diferentes tipos de quadro, todos eles baseados no formato genérico), o número de sequência (que varia de acordo com o tipo de pacote enviado), as identificações de endereços (da rede PAN destinatária e remetente, e o endereço do dispositivo destinatário e remetente) e um campo que informa as opções de segurança utilizadas no quadro. O *payload* contém os dados provindos da camada acima e o *Check Sum* contém a sequência de verificação de erros do quadro (FILHO, 2015).

Para melhor entendimento sobre a estrutura do frame 802.15.4, é válido explicitar a função de cada campo (FILHO, 2015).

Campo Frame Control (FCF): o primeiro campo do cabeçalho é o campo de controle de quadro. Este campo indica o tipo de quadro MAC sendo transmitido, define o formato do campo de endereço, e controla o reconhecimento (acknowledgment) de quadros. Em suma, o campo de controle determina como é o restante do frame e o que ele contém.

Campo Sequence Number(DSN): este campo especifica a sequência de identificação do quadro. Para quadros beacon, especifica um BSN. Para quadros do tipo data, acknowledgment ou command, o número de sequência DSN é usado para fazer o casamento ("match") de um quadro de acknowledgment com um quadro de dados ou um quadro de comando.

Campo Destination PAN Identifier: especifica o identificador único da PAN do receptor do quadro. Um valor de 0xFFFF representa o identificador de broadcast de PAN, que deve ser aceito

como um identificador de PAN válido por todos os dispositivos correntemente ouvindo naquele canal.

Campo Destination Address: define o endereço do receptor. Um valor de 0xFFFF representa o endereço curto de broadcast, que deve ser aceito como um endereço válido por todos os dispositivos correntemente ouvindo naquele canal.

Campo Source PAN Identifier: especifica o identificador único da PAN do dispositivo transmissor do quadro. Este campo deve ser incluído no quadro apenas se o campo Source Addressing Mode é diferente de zero. O identificador da PAN de um dispositivo é inicialmente determinado durante o processo de associação do dispositivo à PAN mas pode mudar em decorrência de uma resolução de um conflito de identificadores de PAN.

Campo Source Address: especifica o endereço do dispositivo transmissor do quadro.

Campo Frame Payload: contém informação específica do tipo de quadro. Se o cabeçalho do bit "Security Enabled" estiver ligado, o payload pode estar protegido por criptografia.

Campo Frame Check Sum (FSC):contém o resultado da aplicação do CRC de 16 bits do ITU-T sobre o cabeçalho e o payload.

O tamanho do campo de endereço pode variar entre 0 e 20 bytes. Por exemplo, um quadro de dados pode conter a informação de endereço fonte e de destino, enquanto que um quadro de *acknowledgment* não contém qualquer informação de endereço. Por outro lado, um quadro *beacon* pode conter apenas informação de endereço de origem. Além disso, endereços curtos ou endereços IEEE de 64 bits podem ser usados. Esta flexibilidade exibida pelo formato do quadro MAC ajuda a aumentar a eficiência do protocolo, mantendo os pacotes curtos. O cabeçalho MAC também trata das opções auxiliares de segurança usadas na transmissão do quadro. Para tanto é utilizado o padrão de criptografia avançado (AES), que descreve rotinas de segurança utilizando chaves com comprimento de 128, 192 ou 256 bits. O campo de *payload* possui tamanho variável; entretanto, observa-se que o tamanho total de quadro MAC não pode exceder 127 bytes de comprimento. Os dados contidos no campo de payload são dependentes do tipo de quadro. Outros campos do quadro MAC são o número de sequência e a sequência de verificação de quadro (FCS - Frame Sequence Check). Numa rede 802.15.4 uma transação só é considerada um sucesso quando o quadro de reconhecimento (ack) contém o mesmo número de sequência do quadro recebido previamente. O FCS ajuda a verificar a integridade do quadro MAC. Ele é implementado como uma verificação de redundância cíclica (CRC – Cyclic Redundancy Check) de 16-bit padronizado pelo do ITU-T (FILHO, 2015).

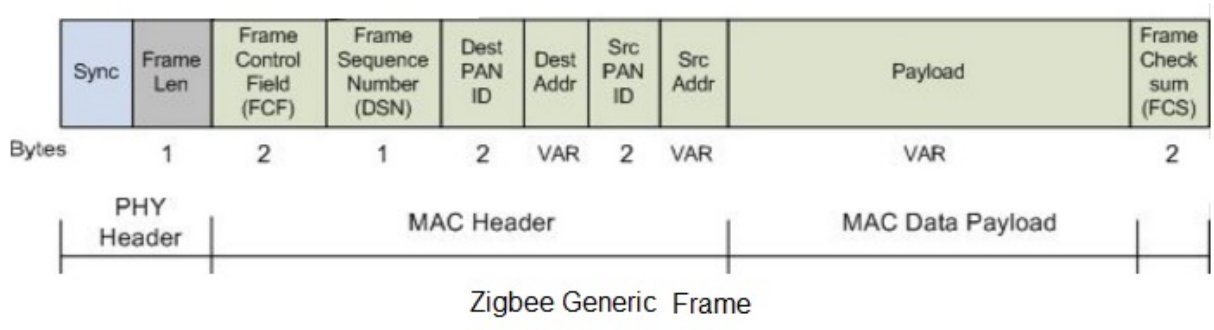

Figura 2.5: Estrutura do pacote ZigBee. Fonte: (RF WIRELESS WORLD, 2012).

Nenhuma das tecnologias sem fio Zigbee foi projetada para operar em taxas de bits típicas para dispositivos simples, como sensores, luzes e outros dispositivos para automação industrial e automação residencial. Um problema com o design do Bluetooth e do UWB é a ausência de rotinas de roteamento. Suas redes suportam apenas a hierarquia mestre/escravo e topologia em estrela. Assim, a pilha ZigBee foi criada com o protocolo de roteamento de alta ordem, responsável pela expansão de rede, aceitando topologia, estrela, malha e árvore. O ZigBee, definido pelo padrão IEEE 802.15.4, surgiu para preencher uma lacuna deixada pela rede sem fio. Ao observar as redes sem fio já existentes, como Wi-Fi, WiMAX, Bluetooth, telefones móveis (GPRS -EDGE), percebe-se que nenhuma delas é dedicada a uma rede de equipamentos simples que não requisite grande complexidade e onde a economia de baterias e o baixo custo sejam essenciais, como ocorre numa rede de sensores, iluminação, refrigeração e afins. As prioridades padrão do ZigBee são o consumo de energia (baterias), custos e complexidade reduzidos. Esses princípios levaram seu desenvolvimento. Para comparação, enquanto um nó Bluetooth operado por bateria funciona por cerca de 1 semana, o nó ZigBee, que é alimentado pela mesma bateria, funciona por aproximadamente um ano. Este é o resultado do longo tempo em que os dispositivos mais simples ficam em modo sleep, acordando de tempos em tempos e perguntando por mensagens que lhe foram enviadas enquanto estavam dormindo (ZUCATO, 2009).

A estrutura da pilha ZigBee, dividida em camadas, pode ser vista na figura 2.6. O ZigBee é definido nas camadas intermediárias. Aí estão inclusos os algoritmos de roteamento, as regras para formação e integração da rede, dentre outros.

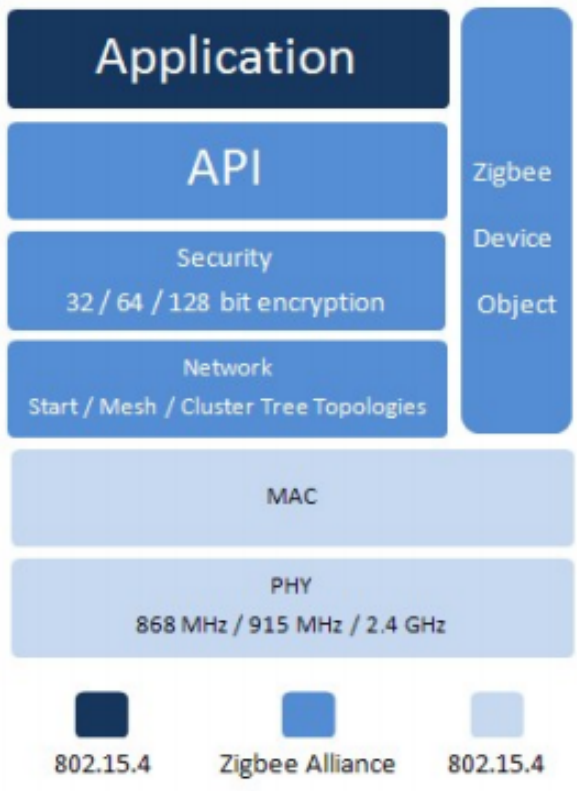

Figura 2.6: Arquitetura da pilha ZigBee. Fonte: (PIRES; MIANI; SOUZA MENDES, 2009).

Camada física - PHY

A camada física foi projetada para acomodar a necessidade de um baixo custo permitindo ainda altos níveis de integração. O uso da sequência direta permite que os circuitos analógicos possam ser muito simples e muito tolerantes a implementações baratas. Esta camada oferece a interface com o meio físico onde as comunicações realmente ocorrem, e é responsável pelas seguintes tarefas (KINNEY, 2003):

- Ativação e desativação do transmissor/receptor de rádio;
- Detecção de energia dentro do canal atual;
- Indicação de qualidade do link para os pacotes recebidos;
- Avaliação de canal livre;
- Seleção da frequência do canal;
- Transmissão e recepção do pacote de mensagem de dados.

Neste modelo, três frequências espectrais são definidas e variam de acordo com a região. Na Europa, somente o canal 0 é usado na frequência de 868,3 MHz. Nos Estados Unidos, é usada a faixa de 902 MHz a 928 MHz, sendo os canais espaçados 2 MHz. O restante do mundo usa a

frequência de 2,4 GHz, com 16 canais espaçados a 5 MHz estabelecido pelo IEEE 802.15 .4 (SILVA, 2014).

Camada de enlace - MAC

Essa camada é responsável por gerenciar o acesso ao meio, enviando quadros de caracteres (beacon frames) para sincronização e transferência de informações de maneira confiável. Existem métodos de acesso de dois canais: a comunicação com beacon habilitado e a comunicação com beacon não habilitado. No primeiro, o coordenador da rede envia quadros por broadcast para sincronizar os dispositivos. Neste modo, você não precisa usar o protocolo CSMA/CA porque é garantido um intervalo de tempo para a transmissão de cada dispositivo. No segundo, é necessário verificar se o canal está disponível antes de iniciar a transmissão, para isso usa-se o protocolo CSMA/CA (SILVA, 2014).

Esses dois modos refletem diretamente a configuração dos módulos XBee. Graças a isso, o dispositivo pode estar no modo "sleep"para economizar energia, enquanto o outro não permite essa função e essa é a razão pela qual ele é usado apenas no coordenador e nos módulos do roteador (SILVA, 2014).

Essa camada também fornece o algoritmo Advanced Encryption Standard (AES) para proteger quadros transmitidos. Além disso, o ZigBee ainda possui as camadas mais altas para garantir a segurança das informações (SILVA, 2014).

Camada de rede - NWK

A camada de rede utiliza endereços de rede e roteamento de pacotes. Os deveres desta camada ZigBee incluem: (KINNEY, 2003):

- Detectar e estabelecer com sucesso uma nova rede;
- Associar ou desassociar dispositivos de uma rede;
- Configurar um novo dispositivo;
- Endereçar os novos dispositivos a partir do coordenador;
- Sincronização os dispositivos através de beacons de varredura ou por pooling.
- Assegurar a segurança dos quadros transmitidos e recebidos.

Essa camada visa permitir que a rede tenha um crescimento espacial sem a necessidade de transmissores de alta potência além de lidar com um grande número de nós com baixa latência. A adição de um roteador entre os nós de interesse aumenta o alcance da comunicação sem exigir grandes antenas e alta potência de transmissão.

Esta camada foi projetada para permitir que a rede tenha um crescimento espacial sem a necessidade de transmissores de alta potência, além de poder manipular grande quantidade de

nós com baixa latência. A adição de um roteador entre os nós de interesse aumenta o alcance da comunicação sem a necessidade de grandes antenas e alta potência de transmissão (MAIA, 2017).

Camada de aplicação - APL

A camada de aplicação ZigBee contém uma subcamada Application Support (APS), do ZigBee Device Object (ZDO) e dos objetos de aplicação definidos pelo fabricante (Application Framework). A subcamada do APS é responsável por fazer a manutenção de tabelas para efetuar o *binding*, que é a capacidade de conectar dois dispositivos com base em seus serviços, requisitos e mensagens de roteamento entre dispositivos conectados. Outra responsabilidade da subcamada APS é o *discovery*, que é a capacidade de determinar quais outros dispositivos funcionam na área de trabalho pessoal do dispositivo. A tarefa da ZDO é identificar recursos de dispositivos na rede, iniciar ou responder a solicitações de conexão e estabelecer um relacionamento seguro entre dispositivos de rede, selecionando um dos métodos de segurança do ZigBee, como uma chave pública e uma chave simétrica. Os objetos de aplicação definidos pelo fabricante usam aplicações em tempo real de acordo com as definições de aplicativos definidos para o ZigBee (KINNEY, 2003).

2.3.2 Tipos de dispositivos ZigBee

O padrão IEEE 802.15.4 define dois tipos de dispositivos de rede ZigBee: dispositivos de função completa ((Full Function Device - FFD)) e dispositivos com funcionalidade reduzida (Reduced Function Device - RFD). Os FFDs são aqueles que podem funcionar em qualquer modo padrão, coordenador, roteador ou dispositivo final. Eles podem se comunicar com outros dispositivos FFD e RFD. Os dispositivos RFD se comunicam apenas com dispositivos FFD. Dessa forma, fica claro que os dispositivos de RFD podem funcionar apenas como dispositivos finais (End Devices) na rede. Eles são dispositivos mais simples e baratos com recursos e capacidade de memória mínimos (IEEE, 2006).É por isso que ele é dedicado para aplicações simples, como detecção ou ligar/desligar luzes remotas. O RFD não tem capacidade de se tornar um coordenador ou roteador e não pode se comunicar com outros RFDs, por isso ele se comunica diretamente com o FFD (BRONZATTI, 2013).

Tabela 2.1: Tabela de funcionalidades dos dispositivos ZigBee. Fonte: Traduzido de (PINHEIRO, 2004).

Coordenador da Rede - FFD	Nó da Rede - RFD
Ajustes de parâmetros da rede	Função passiva na rede
Transmite como informações pela rede	Efetua buscas por redes disponíveis
Gerencia os nós da rede	Transferência de dados da aplicação
Armazena informações dos nós de rede	Determina o status dos dados
Distribuição entre nós de rede	Solicitar dados ao coordenador da rede
Ópera tipicamente no estado "ativo"	Pode permanecer no estado "dormir"por longos períodos

2.3.3 Funções lógicas dos dispositivos

Para compreender a estrutura de uma rede ZigBee, deve-se descrever os três tipos de dispositivos existentes: Coordenador, Roteadores e End Devices (dispositivo final).

Coordenador

O ZigBee Coordinator é o nó central e o mais importante da rede. Apresenta maior complexidade e normalmente tem interface de configuração do usuário com a rede. É o responsável pela formação e gerenciamento da mesma e age também como roteador das mensagens, além de determinar o PAN ID a ser utilizado por todos os dispositivos que fazem parte da mesma rede (SILVA, 2014). Vale ressaltar, que somente um dispositivo Zigbee Cordinator é suportado na rede, exceto em topologia do tipo malha, onde estes atuam como FFD.

Roteador

O ZigBee Router tem a função de rotear as mensagens pela rede, conectando o coordenador aos dispositivos finais. Este componente, apesar de opcional é de grande importância, uma vez que estende a rede e amplia o alcance da mesma (SILVA, 2014).Estes dispositivos também são FFD.

End Device

O ZigBee End Device é o nó final da rede. São dispositivos mais simples e de menor custo para aquisição dessa infraestrutura. Tem permissão somente para conectar-se a outros Routers ou diretamente ao Coordinator para troca de mensagens. Possuem capacidade de transmitir e receber informações e por ter baixo custo de aquisição, são recomendados para pequenas redes em topologia estrela. Ele pode ser configurado para consumir pouca energia por meio do modo sleep, sendo o único dos três que possui essa opção. Os outros dois possuem buffers para guardar as mensagens que devem ser repassadas aos dispositivos finais caso estes estejam em modo sleep no momento do envio. Por este motivo, eles devem estar sempre ativos (SILVA, 2014). O dispositivos End Devices são os únicos do tipo RFD (Reduced Function Device).

Outro componente de grande importante na estrutura da rede Zigbee é o **ZigBee Trust Center(TC)**. Um dispositivo ZigBee totalmente funcional (FFD) responsável pela autenticação de dispositivos que se juntam à rede ZigBee. Ou seja, quando um dispositivo tenta entrar na rede, o roteador mais próximo notifica o TC de que um dispositivo juntou-se a rede, daí o TC instrui o roteador a autenticar ou encerrar o novo nó de conexão.

2.3.4 Topologia da rede

De acordo com a norma IEEE 802.15.4, as topologias compatíveis com o padrão são ponto a ponto e estrela. Devido a essa limitação, o protocolo ZigBee adicionou novas camadas que permitem a topologia do tipo árvore e rede (SILVA, 2014).

Estrela

A topologia em estrela mostrada na Figura 2.7 possui um coordenador e outros dispositivos que se comunicam diretamente com ela. Portanto, todas as mensagens são enviadas ao coordenador

responsável por aplicá-las a outros nós da rede. Uma das características dessa topologia é que possui baixa flexibilidade e, portanto, é recomendada para projetos de baixa complexidade (SILVA, 2014).

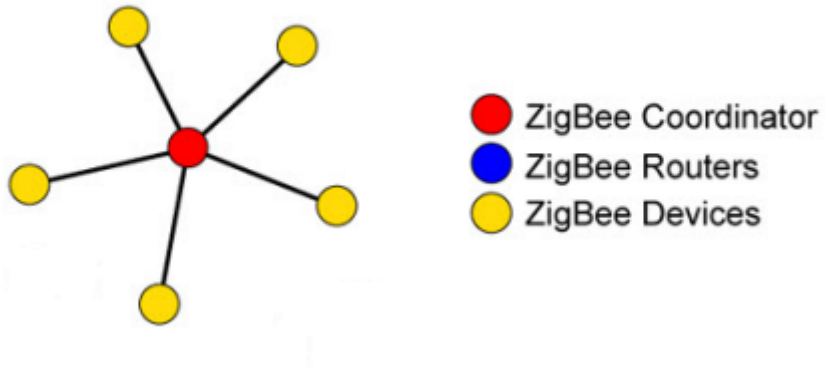

Figura 2.7: Topologia estrela. Fonte: (YAHIA, 2016).

Árvore

A topologia ponto-a-ponto permite dois sistemas: uma árvore e uma grade.Na topologia em árvore existem limitações na comunicação entre os dispositivos, ou seja, nem todos os roteadores se comunicam diretamente (SILVA, 2014). A Figura 2.8 apresenta a topologia em árvore.

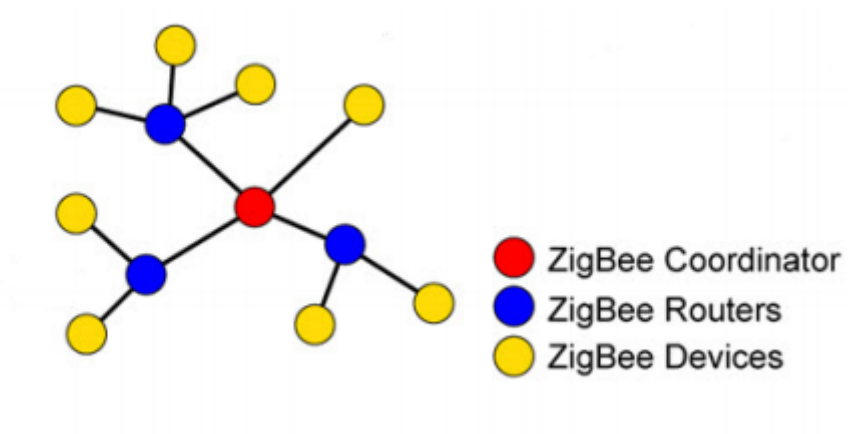

Figura 2.8: Topologia em árvore. Fonte: (YAHIA, 2016).

Malha

Em contraste, a rede em malha permite que todos os roteadores se comuniquem diretamente, permitindo definir novos caminhos quando qualquer dispositivo é desconectado da rede ou entra na rede, tornando-o muito flexível (SILVA, 2014). A Figura 2.9 apresenta a topologia em malha.

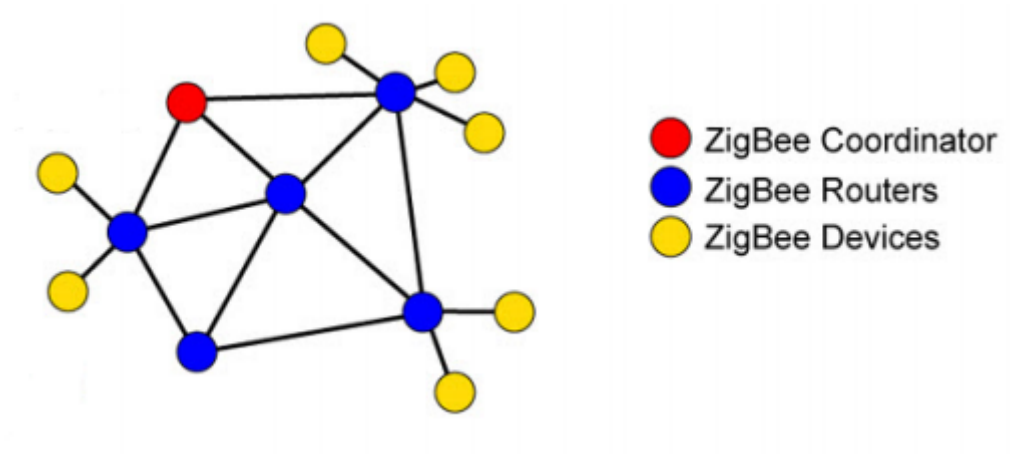

Figura 2.9: Topologia em malha. Fonte: (YAHIA, 2016).

2.3.5 Endereçamento e Canais da Rede Zigbee

Para enviar mensagens usando módulos ZigBee, é necessário saber o endereço do módulo do receptor. Os tranceptores ZigBee têm vários endereços diferentes, cada um projetado para um propósito específico e tem uma composição diferente. O endereço principal refere-se a um número de série de 64 bits atribuído de forma exclusiva e contínua a cada transceptor, ou seja, não é possível que os diferentes módulos ZigBee tenham o mesmo endereço de 64 bits. Esse endereço também é chamado de controle de acesso à mídia (Media Access Control - MAC) ou popularmente como long address. Há também um endereço de 16 bits conhecido como short address ou my address, este endereço é atribuído dinamicamente a cada tranceptor ao configurar a rede. Tal endereço é único em uma rede e fornece vários endereços na memória do ZigBee, porque é mais curto. Finalmente, cada módulo pode aceitar uma pequena string de texto chamada "identificador de nó", permitindo que seja endereçado da forma que achar mais "amigável"(FALUDI, 2010).

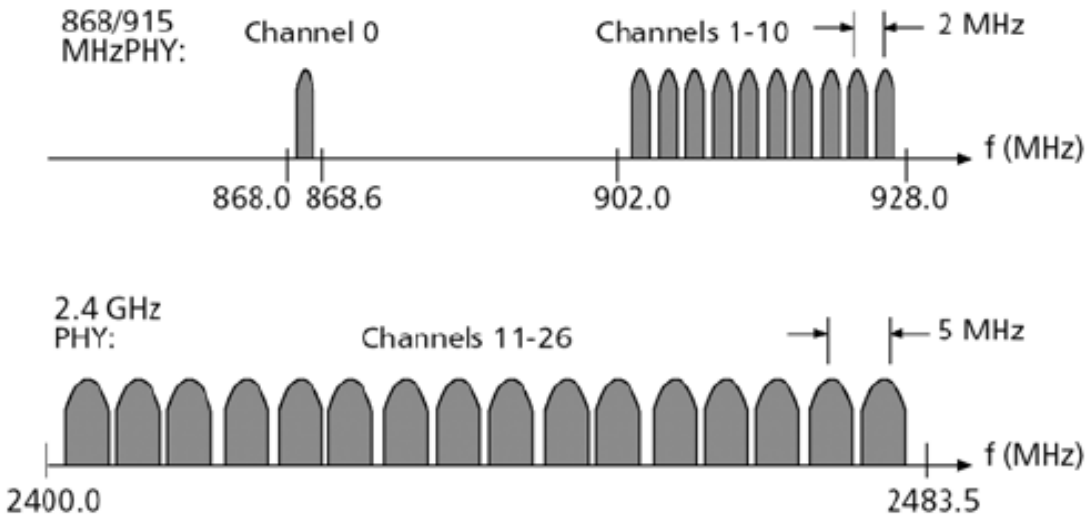

Figura 2.10: Endereçamento de canais da rede Zigbee. Fonte: (GUTIERREZ et al., 2006).

Embora a topologia de endereçamento seja funcional, somente o endereçamento correto não é suficiente para realizar a troca de mensagens entre os transceptores. Portanto, é essencial que os dois ou mais transmissores que trocam mensagens estejam sintonizados na mesma frequência. O coordenador ZigBee geralmente seleciona a rede PAN e verifica todos os canais disponíveis e seleciona um para estabelecer a conversa nessa rede, de modo que todos os nós dos componentes da rede em questão devem usar o mesmo canal (FALUDI, 2010). ZigBee trabalha na banda de 2,4 GHz, usando 16 canais. Ao iniciar uma rede, o coordenador deve selecionar um "bom" canal para a rede operar. Para fazer isso, ele realiza uma varredura de energia em múltiplos canais (isto é, frequências) para detectar níveis de energia em cada canal. O coordenador remove canais com níveis de energia excessiva de sua lista de canais potenciais para começar. A tabela 2.2, mostra em quais frequências cada um dos 16 canais operam.

Tabela 2.2: Tabela de representação dos números de frequência do canal Zigbee. Fonte: Traduzido de (LIBELIUM, 2018).

Número do canal	**Frequência**
0x0B – Canal 11	2.400 – 2.405 GHz
0x0C – Canal 12	2.405 – 2.410 GHz
0x0D – Canal 13	2.410 – 2.415 GHz
0x0E – Canal 14	2.415 – 2.420 GHz
0x0F – Canal 15	2.420 – 2.425 GHz
0x10 – Canal 16	2.425 – 2.430 GHz
0x11 – Canal 17	2.430 – 2.435 GHz
0x12 – Canal 18	2.435 – 2.440 GHz
0x13 – Canal 19	2.440 – 2.445 GHz
0x14 – Canal 20	2.445 – 2.450 GHz
0x15 – Canal 21	2.450 – 2.455 GHz
0x16 – Canal 22	2.455 – 2.460 GHz
0x17 – Canal 23	2.460 – 2.465 GHz
0x18 – Canal 24	2.465 – 2.470 GHz
0x19 – Canal 25	2.470 – 2.475 GHz
0x1A – Canal 26	2.480 – 2.485 GHz

2.4 Segurança em ZigBee

2.4.1 Conceitos de Segurança

Os quatro principais conceitos de segurança do ZigBee são: (VIDGREN et al., 2013)

1. ***Nível de segurança***: O ZigBee suporta dois níveis de segurança diferentes: Alta Segurança (também conhecida como Segurança Comercial) e Segurança Padrão (também conhecida

como Segurança Residencial). As diferenças entre esses níveis de segurança estão principalmente no gerenciamento e distribuição de chaves.

2. ***Trust Center (TC) ou Central de Confiabilidade***: O TC é responsável pelo gerenciamento de segurança. Ele fornece um mecanismo de segurança usando três tipos de chaves: a chave de rede, a chave mestre e a chave de link. Além disso, o TC é responsável por selecionar o nível de segurança adequado e pelo gerenciamento de chaves. Todos os dispositivos ZigBee compartilham a chave de rede comum, enquanto a chave de link pode ser compartilhada por apenas dois dispositivos ZigBee. A chave de link é derivada da chave mestre, que é a base para a segurança de longo prazo entre dois dispositivos ZigBee.

3. ***Autenticação e Criptografia de Dados*** Os dados são criptografados usando o Padrão de Criptografia Avançada (AES) de 128 bits com CCM (CCM = CBC-MAC = Counter with Cipher Block Chaining Message Authentication Code), permitindo a autenticação e criptografia de dados, formando assim uma FIPS (Federal Information Processing Standards), modo de segurança compatível chamado AES-CCM. O modo CCM é um modo de operação apenas para criptografias de bloco criptográfico de 128 bits. Ele combina o modo contador com a autenticação CBC-MAC e usa a mesma chave de criptografia para ambos os modos. O ZigBee usa uma versão ligeiramente modificada do CCM chamada CCM*, que oferece mais flexibilidade que o CCM padrão: o CCM* permite usar autenticação ou criptografia, enquanto ambos são sempre necessários no CCM.

4. ***Integridade de Dados***: Diversas chaves e métodos de segurança diferentes são usados para garantir a integridade dos dados. O Message Integrity Code (MIC) pode ser usado para garantir que os dados não tenham sido alterados durante a transmissão (veja a Figura 2.11). O ZigBee suporta comprimentos de MIC de 16, 32, 64 e 128 bits. O MIC é gerado usando o protocolo CCM*.

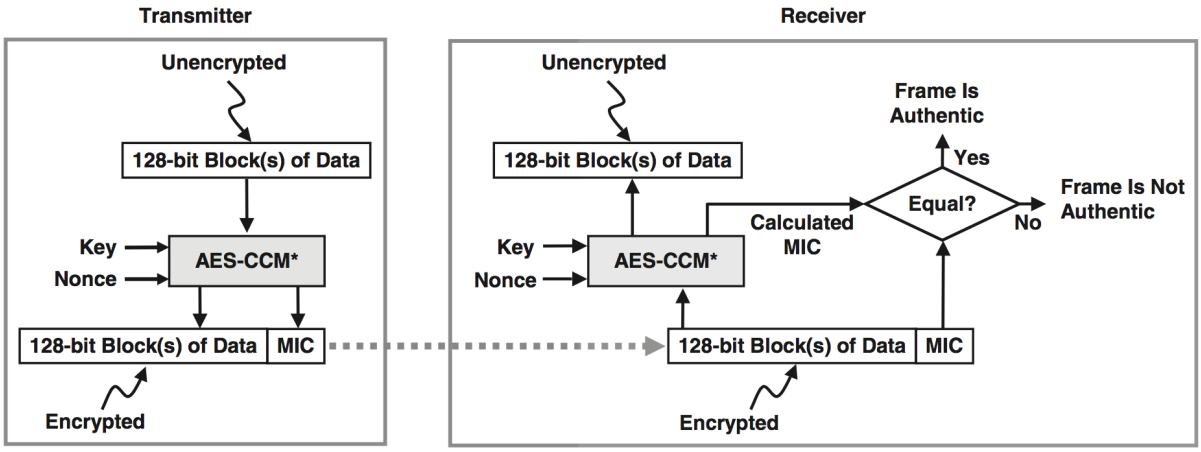

Figura 2.11: Garantia da integridade dos dados usando o MIC. Fonte: (VIDGREN et al., 2013).

2.4.2 Avaliação e Visão Geral dos Recursos de Segurança

A ZigBee Alliance é um grupo de empresas que desenvolve e mantém o padrão ZigBee. Este é definido como uma especificação para um conjunto de protocolos de comunicação de alto nível construído sobre o IEEE 802.15.4. Uma característica importante do ZigBee é que este procura ser mais simples e menos dispendioso do que outros Wireless Personal Padrões de redes de área (WPAN), como Bluetooth e IrDA (Infrared Data Association). O foco principal do padrão ZigBee são aplicações que exigem baixa taxa de dados, longa duração da bateria e segurança. A principal diferença entre ZigBee e outras WPAN's é o tipo de dispositivos que podem ser implantados na rede, nomeadamente: Full Function Devices (FFD) e Reduced Function Device (RFD). Um FFD pode receber e enviar mensagens sobre o 802.15.4, enquanto um RFD é geralmente um sensor que "dorme" a maior parte do tempo e só "acorda" para enviar mensagens.

Sendo baseado no padrão IEEE 802.15.4 (IEEE, 2006), ZigBee compartilha sua especificação de camadas de baixo nível, definida como (PHY) e as camadas Medium Access Control (MAC). Basicamente, o primeiro lida com a taxa de bits e canal de comunicação, enquanto o último lida com o acesso ao canal físico de rádio, este gerencia a sincronização de rádio e fornece um link confiável entre dois nós. No que diz respeito à segurança, ZigBee compartilha os recursos básicos definidos no IEEE 802.15.4, que operam na camada MAC (XIAO et al., 2006). Infelizmente, essas capacidades são parcialmente limitadas pelas diversas gamas de aplicações que devem ser suportadas. Eles basicamente consistem em manter uma lista de controle de acesso (ACL) e usa o Advanced Encryption Standard (AES) (NIST..., 2018) para projetar as transmissões de quadros. Além disso, ambos os serviços são somente opcionais, e o padrão IEEE 802.15.4 não inclui gerenciamento de chaves e esquemas de autenticação de dispositivos definidas pelas camadas superiores. No entanto, a especificação 802.15.4 / ZigBee define algumas capacidades adicionais de segurança para evitar potenciais vulnerabilidades, como interceptação de mensagens, modificação e fabricação, bem como a interrupção da comunicação. A especificação do ZigBee redigida em 2007, define dois modos especiais de segurança: **Segurança padrão** e **Alta segurança**. O primeiro é usado em aplicativos comuns, enquanto o último, que é implementado no ZigBee PRO, fornece mecanismos de segurança mais altos a um custo na demanda nos recursos do dispositivo.

Para uma descrição mais detalhada da tecnologia ZigBee, algumas características são relevantes e vale a pena ser descritas, como:

- ***Chaves ZigBee***: Dispositivos ZigBee estabelecem comunicações seguras através da rede, protegendo mensagens através do uso de chaves simétricas. Vale mencionar que a comunicação no modo de Segurança Padrão no ZigBee é feita de modo seguro pela chave de rede, que é compartilhada entre todos os dispositivos na rede. Enquanto a comunicação no modo de Alta Segurança no ZigBee PRO é garantido através da utilização de três chaves diferentes: *Chave de Ligação*, *Chave Mestre* e *Chave de Rede*. A Chave de Ligação é uma chave de 128 bits que é compartilhada entre dois nós e é aplicado para proteger as comunicações unicast. A geração das chaves de ligação é feita usando a Chave Mestre, que é pré-instalado na fábrica, adicionado pelo usuário final fora de banda ou enviado de um *Trust Center* (Centro de Confiabilidade), um dispositivo especial que outros dispositivos confiam para a distribuição

de chaves de segurança. A Chave de Rede é uma chave de 128 bits que é compartilhado entre dispositivos na rede e é usado para proteger as comunicações de transmissão. (RADMAND et al., 2010)

- ***Troca de Chaves***: Troca de chaves de chave simétrica (SKY) é um novo mecanismo de segurança no ZigBee PRO que é usado para atualizar periodicamente a chave de link. SKKE emprega a Chave Mestre para inicializar uma troca segura, aumentando a segurança.

- ***Camadas de Segurança Adicionais***: ZigBee basicamente fornece serviços de segurança em três camadas diferentes, *Enlace* (MAC), *Rede* (NWK) e *Aplicação* (APS). Se por um lado as rotas da camada NWK enviam quadros para o seu destino, descobre e mantém a tabela de roteamento. Por outro lado, a camada APS atua como uma extensão da camada de Aplicação (APP), que fornece serviços aos usuários, define o papel dos dispositivos e gerencia os dados na remontagem. Na camada MAC, o ZigBee fornece segurança adicional para mensagens de salto único usando o algoritmo de criptografia AES. Na camada NWK, as chaves de ligação e as de rede são usados para fornecer privacidade usando criptografia AES. Além disso, a integridade de dados também é fornecida usando um Código de Integridade de Mensagens (MIC) esquema de segurança. Finalmente, a subcamada APS executa as funções de segurança da camada APP. Esta função de segurança é baseada nas Chaves de Ligação e Chaves de Rede. A subcamada APS adiciona um auxiliar cabeçalho para transportar informações de segurança. Na camada APS, o MIC também é aplicado para determinar o nível de integridade dos dados.

- ***Mecanismo de Junção de Rede***: ZigBee define três tipos de dispositivos: ZigBee Coordenador (ZC), ZigBee Roteador (ZR) e ZigBee Dispositivo Final (ZED). Um ZC irá iniciar a rede e aceitar pedidos de junção originados de ZRs ou ZEDs. Somente um ZC ou outros ZRs que já aderiram à rede pode aceitar pedidos de junção e encaminhar pacotes (SONG; YANG, 2008). Uma vez que um dispositivo tenha ingressado na rede ZigBee, antes das comunicações começarem, uma mensagem é enviada ao ZC ou a um Trust Center. Nesta fase, é tomada uma decisão avaliando se o dispositivo está autorizado a entrar na rede ou não. Esta decisão baseia-se no tipo de chave e na configuração do Centro de Confiança (LEE et al., 2009). Como pode ser visto na tabela 2.3, existem quatro opções para configurar o Trust Center no ZigBee PRO, enquanto apenas as duas primeiras opções estão disponíveis para a do Centro de Confiabilidade no padrão ZigBee.

Tabela 2.3: Opções de autenticação e configuração do Trust Center (centro de confiança). Fonte: Traduzido de (LEE et al., 2009).

Opção	Informação Requerida	Descrição
1	Sem chaves pré-configuradas	Chave mestre, link e de rede são transmitidos sem criptografia pelo ar. [Over The Air (OTA)]
2	Chave de rede ativa	Se o dispositivo estiver conectado à rede, então a chave de rede ativa não deve mudar.
3	Endereço do Trust Center da chave de link	A base de dados é aberta utilizando o link de ligação entre a Central de Confiabilidade e o Dispositivo Final. Em seguida, a chave de rede é enviada com segurança a partir do Trust Center.
4	Endereço do Trust Center da chave mestre	A chave de link do dispositivo é gerada usando a chave mestre. A chave de rede, por sua vez, é enviada com segurança a partir do Trust Center.

2.4.3 Rogue Device

Uma das ameaças de segurança sem fio mais comuns é o ponto de acesso invasor, ele é usado em muitos ataques, tanto DoS quanto roubo de dados. Muitos outros pontos de acesso não autorizados, no entanto, são implantados por funcionários que desejam acesso sem fio irrestrito - esses pontos de acesso são chamados de pontos de acesso flexíveis. O ponto de acesso também pode pertencer a uma empresa próxima, como eles geralmente são instalados em seu modo padrão, a autenticação e a criptografia não estão ativadas, criando assim um risco à segurança. Como os sinais de LAN sem fio podem atravessar paredes de edifícios, um ponto de acesso aberto conectado à rede corporativa é o alvo perfeito para que esses dispositivos se conectem à rede e consumam a largura de banda da rede (JUNIPER NETWORKS, 2016).

Pontos de acesso desonestos e seus clientes minam a segurança de uma rede empresarial, permitindo potencialmente acesso não autorizado à rede por qualquer usuário ou cliente sem fio na vizinhança física. Os pontos de acesso não autorizados também podem interferir na operação da sua rede corporativa. Tais pontos de acesso não autorizados podem causar os seguintes danos (JUNIPER NETWORKS, 2016):

- Permite que um hacker conduza um ataque *man-in-the-middle*.
- O invasor faz conexões independentes com as vítimas e transmite mensagens entre elas, fazendo-as acreditar que estão falando diretamente umas com as outras por meio de uma conexão privada, quando, na verdade, toda a conversa é controlada pelo invasor.
- Inunde a rede com dados inúteis, criando uma negação de serviço.
- Fornecer um canal para o roubo de informações da empresa.

A melhor maneira de evitar pontos de acesso desonestos e dispositivos não autorizados em sua

rede é analisar cada dispositivo que se associa a ele como uma ameaça em potencial, consistente com abordagens de monitoramento de confiança zero. Se um invasor instalar um ponto de acesso, ele poderá executar vários tipos de scanners de vulnerabilidade e, em vez de precisar estar fisicamente dentro da organização, poderá atacar remotamente - talvez de uma área de recepção, prédio adjacente, estacionamento ou com um alto ganho antena, mesmo a partir de vários quilômetros de distância.

Para evitar a instalação de pontos de acesso não autorizados , as organizações podem instalar sistemas de prevenção contra invasão sem fio para monitorar o espectro da rede em busca de pontos de acesso não autorizados. Estes incluem pontos de acesso gerenciados na rede segura e pontos de acesso na vizinhança. Um sistema de prevenção de intrusões sem fio facilita a tarefa de examinar minuciosamente esses pontos de acesso continuamente para saber se existem pontos de acesso não autorizados entre eles.

Para a detecção de dispositivos desonestos, precisa-se conhecer a rede existente na empresa, além de listar todos os pontos de acessos gerenciados e identificar que se estão ligados a mesma. Esse é o maior problema que grande parte das empresas enfrentam em termos de acesso a dispositivos invasores. Se você não sabe o que está na sua rede, não sabe se a segurança da sua rede está em risco

2.5 Ataques e Vulnerabilidades da rede

Nesta seção, procura-se analisar os recursos atuais do Padrão ZigBee, a fim de avaliar o nível de segurança atualmente fornecida pela plataforma. Categorizamos as vulnerabilidades existentes de acordo com os seguintes fatores: restrições sobre a realização de um ataque bem sucedido e o tipo de perturbação um ataque pode causar na rede. Diante das análises, as vulnerabilidades existentes podem ser divididas em duas categorias principais: aquelas que requerem conhecimento da criptografia de Chaves ZigBee (Ligação, Mestre ou Rede), e aquelas que não.

2.5.1 Ataques que exigem *Key Compromise*

A Key Compromise é muito importante no que diz respeito à segurança. Uma vez que o atacante se apossar de uma chave, ele será capaz de agir sem pressa dentro da rede. Um invasor pode obter a chave de rede por meio de diferentes métodos como ataque remoto ou um ataque físico (IEEE, 2006). No primeiro caso, esse feito pode ser alcançado pela interceptação durante a transmissão fora de banda ou a captura de texto simples de tráfego enviado de um Coordenador ZigBee. Neste último caso, o dispositivo físico é roubado, extraindo a informação diretamente do seu hardware. Ataques remotos dependem da interceptação de mensagens e da exploração mecanismos de chave de troca fora de banda, por isso, nos concentramos no ataque físico, que é mais complexo em vez de se concentrar no ataque remoto.

1. *Eavesdropping*: No ZigBee, as mensagens de broadcast são criptografadas usando a chave de rede, que é compartilhada entre todos os dispositivos na rede. Infelizmente, é apenas neces-

sário comprometer um único dispositivo na rede para o invasor ser capaz de comprometer toda a rede. Usando esta chave o atacante é capaz de capturar o conteúdo de transmissão das mensagens na rede e, portanto, este é uma das vulnerabilidades mais importantes na tecnologia ZigBee. Isto é uma façanha viável, já que um adversário pode obter a criptografia usa chaves remotamente ou fisicamente. Em contraste, as comunicações unicast são protegidas por uma Chave de Ligação exclusiva compartilhada entre dois dispositivos na rede. Isso significa que, se um dispositivo da rede for comprometido por ataque físico, um atacante é capaz de capturar o conteúdo de toda a comunicação, diretamente pelo unicast do dispositivo.

Para resolver este problema, um mecanismo para proteger a troca de chaves deve ser usada. Além disso, a segurança física de dispositivos seriam necessários para evitar este ataque.

2. *Spoofing*: Este ataque é baseado na mesma vulnerabilidade mencionada no anterior: todas as mensagens de difusão são criptografadas usando a mesma chave, a Chave de Rede. Isso permite a entrada de invasores para representar a identidade de qualquer nó na transmissão de mensagens, uma vez que não há verificação de autenticação. Como essa vulnerabilidade se aplica somente a mensagens de difusão, o risco desta vulnerabilidade depende da quantidade de dados enviados por cada aplicação. Para resolver esse problema, um mecanismo para proteger as comunicações de difusão impondo uma autenticação o processo é proposto em (SHIEH; KO, 2006), usando um método unidirecional de assinatura.

2.5.2 Ataques com *Key Compromisse* não requerido

Ataques que não requerem que um atacante obtenha acesso às chaves criptográficas armazenadas em um dispositivo ZigBee são a maior preocupação, uma vez que eles podem ser executados remotamente no espaço sem fio. Não é necessário manipular fisicamente dispositivos. Os dois principais ataques existentes que seguem esta condição são: Repetição e Negação de Serviço (DoS).

Ataque de repetição: Esse tipo de ataque pode se aplicar a muitas aplicações. Por exemplo, em uma sala de servidores onde a temperatura é controlado pelo sensor ZigBee e os dados alterados é apenas +1 ou -1 graus. Ao executar o ataque de repetição, a temperatura pode ser alterada por um adversário. Isso significa que, se um atacante, que implementou o ataque de repetição, farejou o pacote enviado do dispositivo ZigBee para o ar condicionado e repetiu n vezes, a temperatura é adicionada ou diminuída por n graus. Esta temperatura incorreta pode causar danos aos servidores.

DoS: Um grande esforço foi feito pela ZigBee Alliance para poder executar a autenticação e fornecer confidencialidade aos dados transmitidos. No entanto, nenhum esforço foi feito para evitar ataques de Negação de Serviço (DoS). Este ataque pode ser realizado em várias camadas e depende se o atacante se juntou à rede, sendo parte dela (interno) ou não (externo)(MURALEEDHARAN; OSADCIW, 2006),(EGLI, 2006). Se o invasor for *interno*, o ataque DoS pode ser conduzido nas camadas física (PHY), enlace (MAC), rede (NWK) e aplicação (APS), enquanto que, se o atacante é *externo* só pode ser conduzido nas camadas PHY/MAC. A Figura 2.12 classifica todos os possíveis ataques DoS de acordo com cada camada. A possibilidade de executar o ataque DoS

em várias camadas é importante porque ataques mais complexos serão mais difícil de detectar, pois um atacante sempre pretende ser invisível.

2.5.3 Ataques Internos

Na camada de aplicação, o DoS é executado enviando muitas mensagens para o dispositivo (inundação) para interromper o processamento da mensagem.Além disso, esta ação esgota os recursos do dispositivo, como bateria. Este ataque pode ser facilmente detectado, já que as mensagens são enviadas de um dispositivo específico. Na camada NWK, o DoS é executado modificando o protocolo de roteamento padrão. Se o atacante, que é colocado dentro da rede, é um roteador comprometido, pode parar encaminhamento de mensagens entre nós, o que leva a alterações para o protocolo de roteamento. Felizmente, esse ataque DoS pode ser diretamente detectado e evitado pelo protocolo de roteamento padrão.

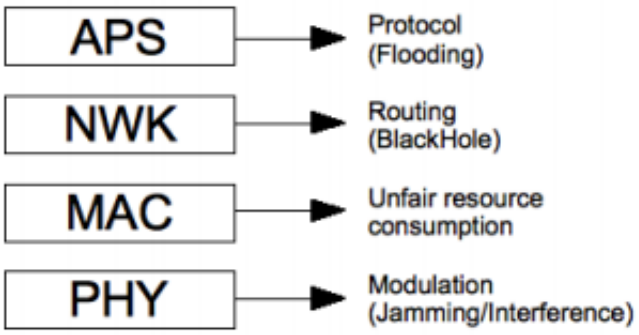

Figura 2.12: Ataques de Serviço de Negação (DoS). Fonte: (EGLI, 2006)

2.5.4 Ataques Externos

Na camada MAC, o ZigBee usa o CSMA/CA (IEEE, 2006) para garantir que todos os dispositivos podem se comunicar através do mesmo canal de comunicação. Quando um dispositivo pretende transmitir dados, o canal de comunicação deve ser ouvido durante o tempo específico. Se o canal é sentido inativo, então o nó tem permissão para começar a transmissão. No entanto, se o canal for detectado como ocupado, o nó adia sua transmissão por um período de tempo aleatório. Um ataque DoS ocorre quando um dispositivo começa a consumir largura de banda injustamente. Por exemplo, se o invasor começar a enviar continuamente dados através do canal de comunicação, outros dispositivos não comunicar um com o outro.

Na camada PHY, o ataque DoS é executado por meio de interferência do canal. Este ataque pode ser executado através de um dispositivo externo, interrompendo o sinal de outros dispositivos alterando a densidade espectral de potência (PSD, Power Spectral Density).

Além disso, a camada MAC também pode ser interferida usando um ataque ACK, ou seja, um ataque DoS otimizado que dificulta ser detectado. Desde que o ZigBee foi construído sobre a pilha do IEEE 802.15.4, algumas de suas vulnerabilidades foram herdadas. Em ZigBee, o remetente

tem a opção de ativar o ACK configurando uma *flag* dentro de cada mensagem enviada. Se esta *flag* estiver definida, o receptor envia uma nova mensagem contendo uma resposta ACK. Contudo, esta mensagem não é autenticada, então qualquer um pode responder com uma mensagem ACK (SOKULLU et al., 2007), (SASTRY; WAGNER, 2004). A especificação 802.15.4/ZigBee não fornece proteção de integridade e confidencialidade para pacotes de confirmação. (SASTRY; WAGNER, 2004)

Existem três dispositivos: o sensor (remetente), o roteador (receptor) e um dispositivo externo (atacante).

- (1) Enquanto o sensor está enviando uma mensagem para o roteador, o atacante interfere e corrompe a transmissão de dados, de modo que o receptor não receba a mensagem completa.
- (2) Para garantir que o sensor não reenvie a mensagem novamente, o atacante gera uma mensagem ACK e envia de volta para o sensor (remetente). Devido a não verificar a autenticação, o sensor assume que a mensagem foi enviada para o roteador.

Para uma maior compreensão do que foi explicitado, é apresentado na Figura 2.13 o cenário de como os três dispositivos interagem para realizar o ataque da camada de enlace:

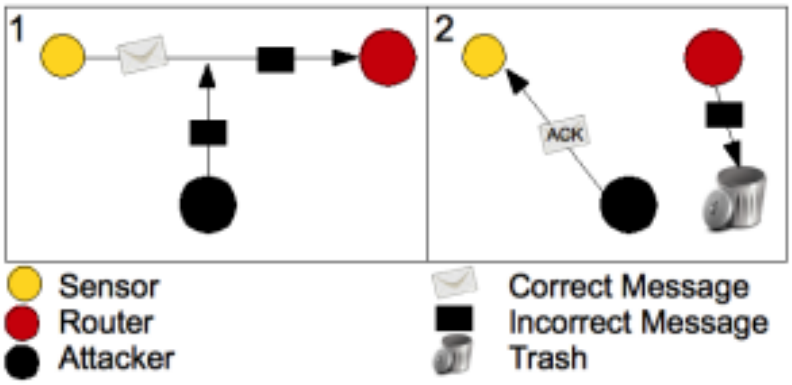

Figura 2.13: Ataque da camada de enlance MAC-ACK. Fonte: (SOKULLU et al., 2007)

Capítulo 3

Ferramentas Utilizadas e Métodos Propostos

Este capítulo apresenta os principais materiais utilizados durante o desenvolvimento deste trabalho, será feita uma apresentação básica sobre as plataformas Raspberry Pi e Arduino, os módulos XBee, Adaptador XBee para Protoboard e Adaptador USB para XBee, e finalmente o dispositivo USB Dongle CC2531. Em seguida será mostrada uma breve descrição e algumas instruções de utilização para cada uma das ferramentas de *software* X-CTU e SmartRF Packet Sniffer. Por fim, será descrita a metodologia que foi aplicada neste projeto.

3.1 Hardware

3.1.1 Raspberry PI

Figura 3.1: Placa Raspberry Pi.

Este "mini-computador" foi desenvolvido pela Fundação Raspberry Pi como um projeto de cunho educacional que tinha o propósito de fazer com que as crianças tivessem acesso a fundamentos de programação e também aprofundassem seu entendimento sobre a natureza dos computadores e seu funcionamento. Para isso, era importante desenvolver um computador simples e que tivesse baixo custo para as escolas.

A Fundação Raspberry Pi se descreve como "uma instituição de caridade com sede no Reino

Unido e que trabalha para colocar o poder da produção digital nas mãos de pessoas em todo o mundo, para que eles sejam capazes de entender e moldar nosso universo cada vez mais digital, capazes de resolver os problemas que importam e equipados para os empregos do futuro" (RASPBERRY PI FOUNDATION, 2018).

O Raspberry Pi contém processador, processador gráfico, slot para cartões de memória, interface USB, HDMI e seus respectivos controladores. Além disso, ele também apresenta memória RAM, entrada de energia e barramentos de expansão. Dessa forma, ele pode ser considerado como um computador completo.

Considerando seu *hardware* bem simples, o Raspberry Pi não é indicado para aplicações mais sofisticados, como *games* ou *softwares* de edição de vídeos. No entanto, suporta várias distribuições Linux e permite a adição de alguns acessórios como teclado, mouse e conexão a TV via saída HDMI, inclusive para a reprodução de vídeos em alta definição (GARRET, 2018).

O sistema operacional deve ser instalado em um cartão de memória SD, pois o computador não possui disco rígido próprio.

3.1.2 Arduino

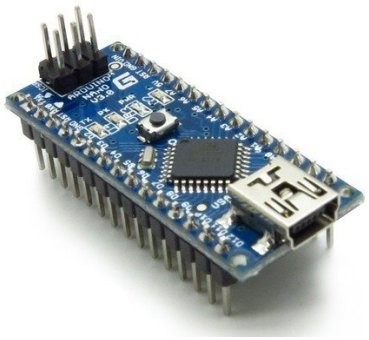

Figura 3.2: Placa Arduino Nano.

O Arduino é uma ferramenta de prototipagem muito popular para o desenvolvimento de produtos IoT. A empresa oferece uma gama de ferramentas de *software*, plataformas de *hardware* e documentação para que profissionais e estudantes possam desenvolver seus projetos em diferentes áreas do conhecimento. O Arduino é o primeiro projeto de *hardware* de código aberto amplamente difundido e tem o objetivo de promover o compartilhamento de informações entre seus usuários, a partir das contribuições de pessoas com a escrita de exemplos e bibliotecas, criação de tutoriais e fóruns (ARDUINO, 2018).

Esta plataforma é projetada com um microcontrolador Atmel AVR com suporte de entrada e saída embutido, uma linguagem de programação padrão baseada essencialmente em C/C++. Basicamente, uma placa Arduino possui um controlador, pinos de entrada e saída (digitais e analógicos) e uma interface serial, empregada para a comunicação entre a placa e o *host*, que por sua vez, é utilizado para a programação e interação. A placa em si não possui qualquer recurso de rede, no entanto, é possível combinar mais de um Arduino a partir de extensões apropriadas

denominadas *shields* (SOUZA et al., 2011).

A placa utilizada para aplicação dos cenários que serão analisados neste trabalho foi a Arduino Nano, mostrada na Figura 3.2.

3.1.2.1 IDE Arduino

Para programar esta plataforma, é necessário instalar a sua IDE *Arduino* que pode ser obtida na página do fabricante (ARDUINO, 2018). Na instalação, já estão inclusos alguns exemplos prontos que podem ser muito úteis para iniciantes e também aproveitados em outros códigos. A tela principal e alguns exemplos disponíveis estão mostrados na Figura 3.3.

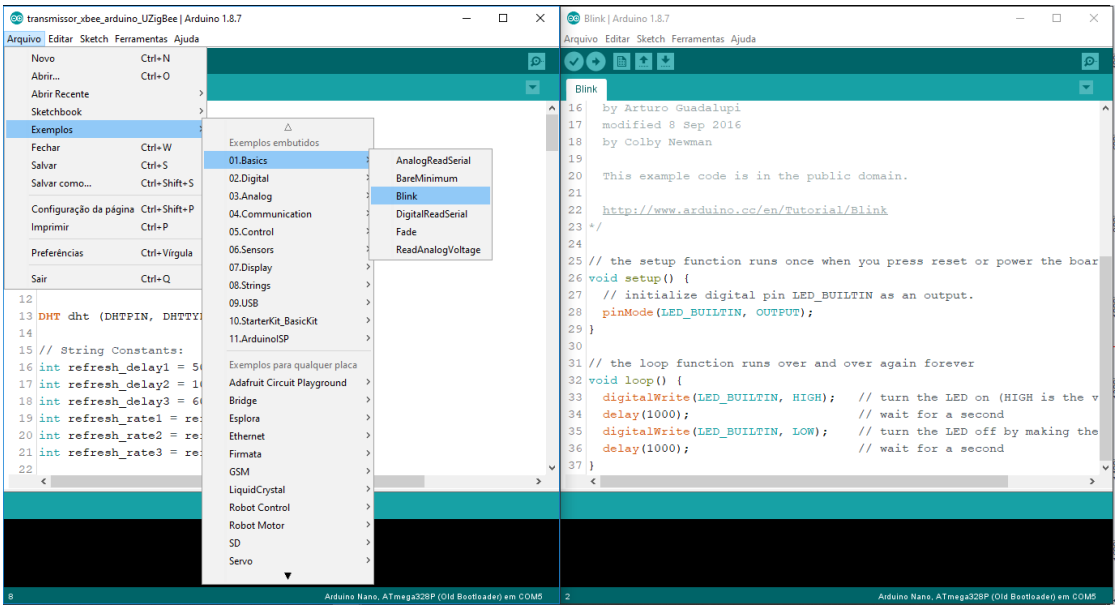

Figura 3.3: À esquerda: Lista com exemplos pré compilados disponíveis. À direita: Tela principal da IDE Arduino com um código exemplo carregado.

3.1.3 Módulo Xbee

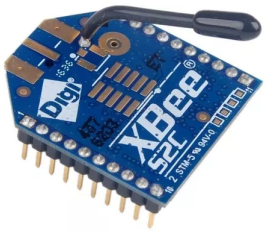

Figura 3.4: Módulo ZigBee Xbee

Os módulos RF XBee e XBee-PRO OEM foram projetados para atender aos padrões IEEE 802.15.4 e suportam as necessidades exclusivas de redes de sensores sem fio de baixo custo e baixo

consumo de energia. Os módulos requerem potência mínima e fornecem entrega confiável de dados entre dispositivos. Eles operam dentro da faixa de frequência ISM de 2,4 GHz e são compatíveis pino-a-pino entre si (MAXSTREAM, 2007).

O módulo Xbee S2C é basicamente uma placa de circuito impresso que comunica-se via radio-frequência a partir do padrão de comunicação de rede sem fio Zigbee. Dessa maneira, é possível interconectar alguns destes módulos a fim de que possam conversar entre si em uma rede *wireless*, ainda que estejam contidos em circuitos eletrônicos independentes. Além disso, sua aplicação é muito adequada em ambientes onde sejam requeridas redes de comunicação e transmissão de dados entre centrais eletrônicas que estejam localizadas distantes entre si, eliminando a necessidade de linhas de comunicação físicas. Permite ainda, o desenvolvimento de redes robustas, confiáveis e com uma boa relação de custo-benefício (EQUIPE MOUNTAINBAJA, 2018).

Especificações do Dispositivo

O módulo Xbee possui como características principais para operação e configuração (EQUIPE MOUNTAINBAJA, 2018),(MAXSTREAM, 2007):

- Distância de alcance: 60m em ambiente fechado e até 1200m em ambiente aberto;
- Frequência de trabalho: 2,4 GHz;
- Taxa de dados: 250Kbps;
- Tensão de alimentação: 2,8 à 3,4 VDC;
- Corrente de transmissão: 45mA (3,3V);
- Corrente de recepção: 50mA (3,3V);
- Antena: Conector Integrado de Chicote, Chip ou U.FL, Conector RPSMA;
- Potência de transmissão: 1mW (0dBm)
- Dimensões: 2,438cm x 2,761cm;
- Temperatura de operação: -40 to 85º C (industrial).

3.1.3.1 Adaptadores XBee

Os pinos do módulo XBee são pequenos e muito próximos entre si, o que impossibilita utilizá-los diretamente na *protoboard* durante a fase de desenvolvimento de um projeto, por esse motivo, utiliza-se um adaptador XBee para *protoboard*, na Figura 3.5. Este adaptador não contém nenhum circuito de conversão de sinal ou regulador de tensão, portanto deve ser observado o limite de nível de sinal do Xbee, que é de 3,3V.

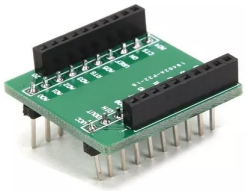

Figura 3.5: Adaptador XBee para protoboard.

Outro adaptador XBee muito importante é o Adaptador USB para XBee, ele é necessário no momento de configurar o módulo XBee, que deve ser conectado ao computador via USB e ser reconhecido pelo *software* XCTU, que será descrito mais adiante. este modelo de adaptador está representado na Figura 3.6. Um ponto que se deve ter atenção é o fato de que esta placa inverte os pinos de recepção (Rx) e transmissão (Tx) originais do módulo XBee, portanto, ao utilizar em um projeto, deve-se conectar aos pinos de maneira trocada.

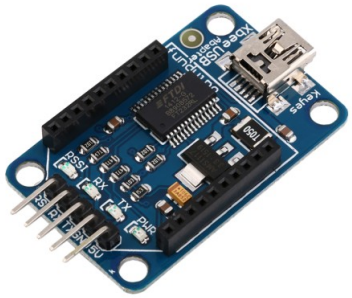

Figura 3.6: Adaptador USB para XBee.

3.1.4 USB Dongle CC2531

Figura 3.7: Adaptador ZigBee USB CC2531.

O Adaptador USB Dongle CC2531 funciona como um *sniffer* de uma rede ZigBee. Também chamado de kit CC2531EMK (CC2531 Evaluation Module Kit), ele contém um *chip* CC2531 e documentação para suportar uma interface de PC para aplicativos 802.15.4/ZigBee. O *dongle* pode

ser conectado diretamente ao PC e pode ser usado como um *sniffer* de pacotes IEEE 802.15.4 ou para outros propósitos. Com a Biblioteca de Firmware USB CC2531 disponível na Web, é possível desenvolver um software próprio para utilizar essa parte. Para programar o depurador, é necessário um programador externo (por exemplo, o SmartRF05EB), que por sua vez, não está incluído no kit. O *dongle* USB pode ser usado como um módulo de referência para prototipagem de dispositivos USB e para testar o desempenho de RF do CC2531 com uma antena PCB de tamanho pequeno (TEXAS INSTRUMENTS, 2018).

O CC2531 é uma solução *system-on-chip* (SoC) ativada por USB para aplicações IEEE 802.15.4, ZigBee e RF4CE. Ele combina o desempenho de um transceptor de RF com um MCU 8051 padrão do setor, memória flash programável no sistema, 8 KB de RAM, entre outros recursos. O CC2531 possui vários modos de operação, sendo adequado para sistemas em que o consumo de energia ultra-baixo é necessário. Tempos de transição curtos entre os modos de operação garantem um menor consumo de energia. O código-fonte para bibliotecas e exemplos de USB HID e CDC podem ser baixados da página do produto CC2531 em www.ti.com (TEXAS INSTRUMENTS, 2018)

3.2 Software

3.2.1 SmartRF Packet Sniffer

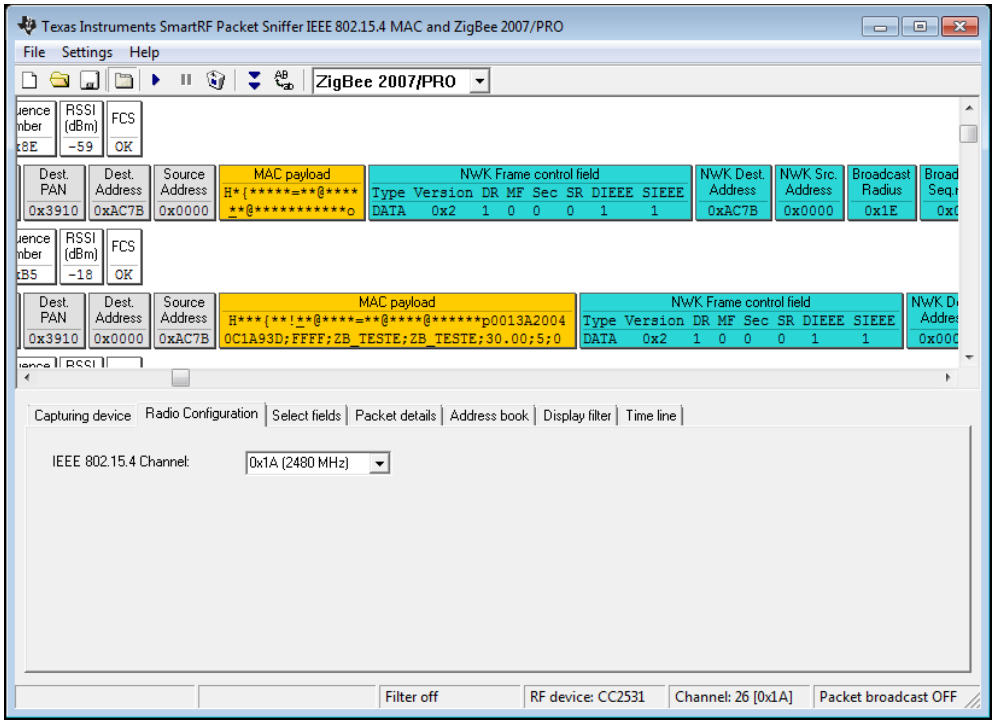

Figura 3.8: Exemplo de tela do SmartRF Packet Sniffer

O SmartRF Packet Sniffer é um aplicativo de software para PC usado para exibir e armazenar pacotes de RF capturados com um nó de hardware que "escuta" RF. Vários protocolos de RF são suportados. O Packet Sniffer filtra e decodifica pacotes e os exibe de uma maneira conve-

niente, com opções de filtragem e armazenamento para um formato de arquivo binário (TEXAS INSTRUMENTS, 2014).

3.2.1.1 Configurações

Inicialmente, após conectar o USB Dongle CC2531 ao computador, deve-se escolher qual o protocolo que será utilizado de acordo com o dispositivo, no caso deste trabalho, foi utilizado o **IEEE 802.15.4/ZigBee**.

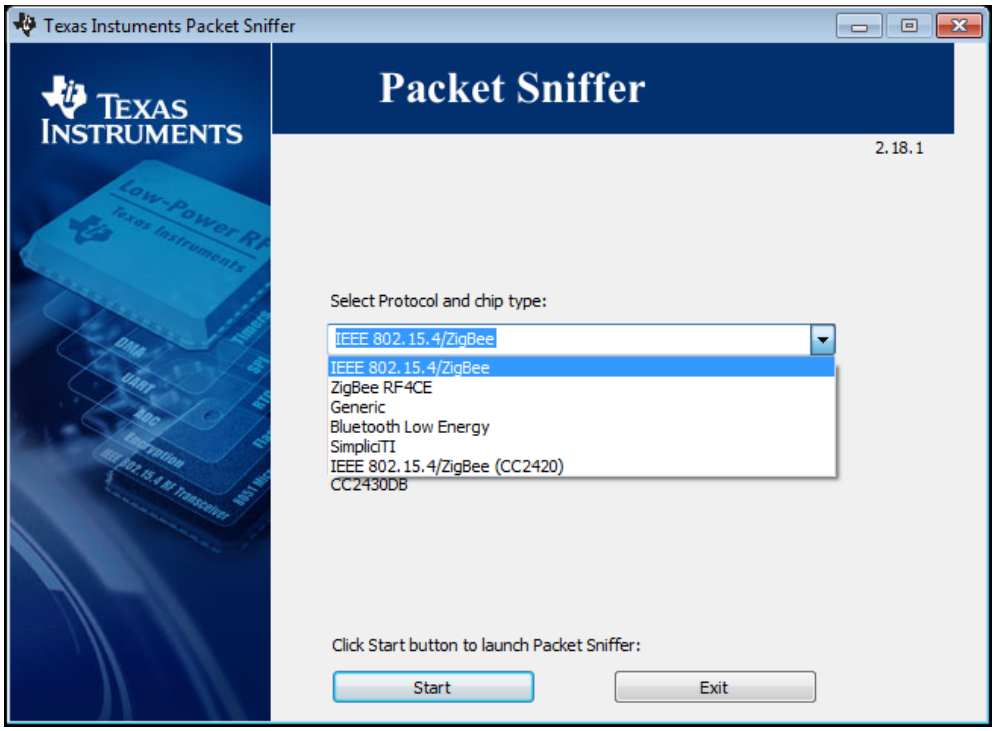

Figura 3.9: Tela de escolha do protocolo no SmartRF Packet Sniffer

Em seguida, será aberta uma nova janela, onde se deve selecionar os seguintes parâmetros, destacados na Figura 3.10:

1. O dispositivo que será utilizado como *sniffer* para executar a captura (indicado em **1**), dentre os reconhecidos pelo *software*;
2. A versão do protocolo (indicado em **2**);
3. E o canal a ser varrido pelo dispositivo *USB Dongle CC2531*, indicado em **3**.

Na Figura 3.10, estão indicados os campos onde se deve selecionar os parâmetros para iniciar uma captura. Em seguida, a captura pode ser iniciada, ao apertar o botão indicado em **4**. Quando for desejado, a captura pode ser salva clicando no botão indicado em **5**.

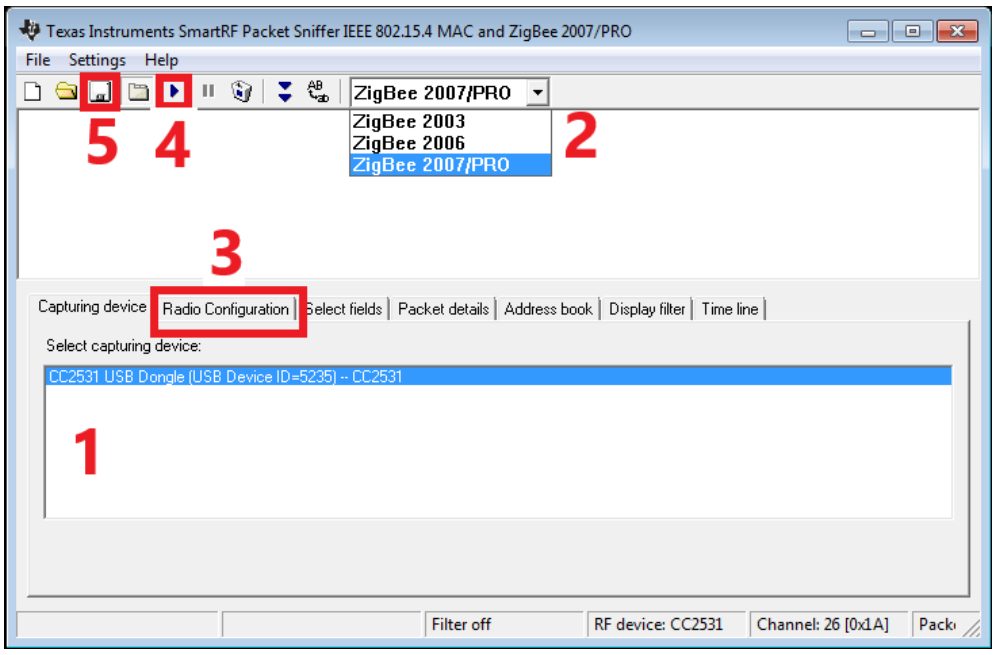

Figura 3.10: Tela de seleção de parâmetros essenciais para iniciar uma captura com o SmartRF Packet Sniffer

Por fim, pode-se analisar os *frames* capturados selecionando os campos que se quer visualizar, conforme a necessidade, isto pode ser feito durante a captura ou após tê-la salvo. Os campos do *frame* são selecionados na aba ***Select field***, conforme mostrado na Figura 3.11. Além disso, para finalizar a captura, basta clicar no botão com um símbolo de *pause*, indicado na Figura 3.11.

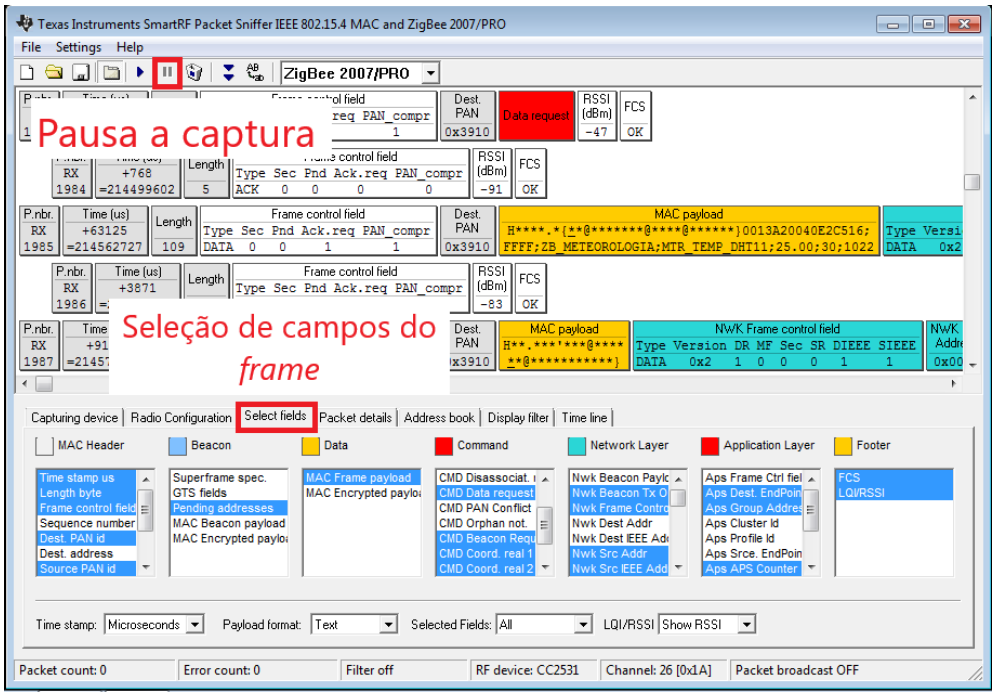

Figura 3.11: Tela de seleção de campos do frame.

3.2.2 XCTU

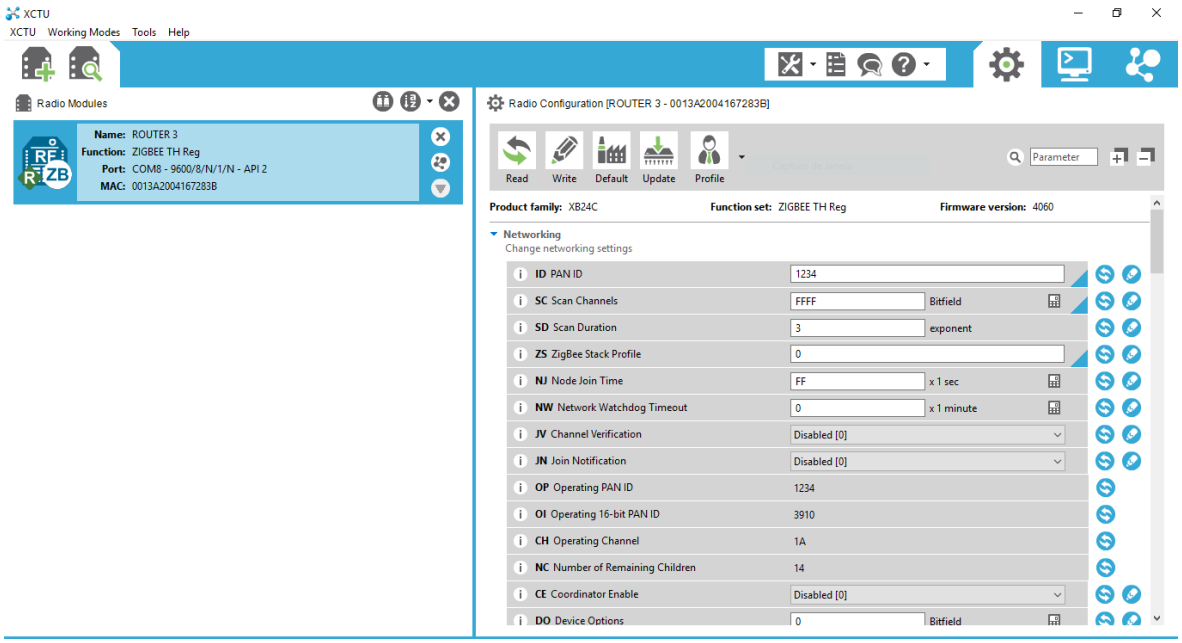

Figura 3.12: Exemplo de tela do XCTU

O XCTU é um *software* gratuito e multi-plataforma compatível com Windows, MacOS e Linux. Sua interface permite a configuração e arquitetura de redes sem fio simples. Esta aplicação possui também uma extensão que permite realizar testes de comunicação entre módulos para validar *range* e envio de pacote de dados. Além disso, há também o API Frame Builder que permite o desenvolvimento simples e rápido de quadros de API de módulos XBee.

3.2.2.1 Configurações

Em primeiro lugar, conecta-se um módulo XBee ao computador, no XCTU inicia-se a busca por dispositivos na porta serial específica, ao encontrar, ele mostrará uma pequena janela com uma lista do que foi encontrado, este deve ser selecionado e adicionado à tela de configuração.

Em seguida, as informações do módulo XBee serão mostradas na tela principal, conforme a Figura 3.12. Tradicionalmente, os parâmetros alterados foram:

- **ID:** Corresponde ao ID da rede ou PAN ID.
- **SC:** Indica o código de identificação do canal.
- **SL e SH:** Correspondem ao Número Serial (endereço MAC), do dispositivo conectado.
- **DH e DL:** Endereço MAC do dispositivo de destino (H=*High*, L=*Low*), pode ser encontrado atrás do módulo ZigBee.
- **NI:** Neste parâmetro é possível designar um nome para o nó.

- **AP:** Normalmente, configurado para **2**. O modo API 2 garante que todos os bytes 0x7E recebidos sejam delimitadores de início: esse caractere não pode fazer parte de nenhum dos outros campos de quadro (comprimento, dados ou soma de verificação), pois deve ter *escape*. (DIGI, 2018).

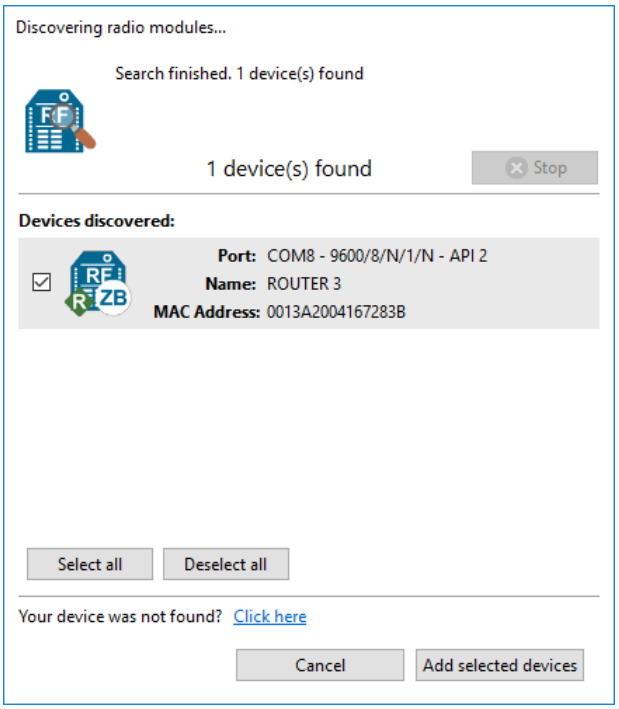

Figura 3.13: Lista de dispositivos encontrados pelo XCTU.

3.3 Topologia da rede ZigBee

A rede de sensores utilizada para a realização das simulações foi configurada conforme a implementação feita por (MAIA, 2017) no Laboratório UIoT do curso de Engeharia de Redes de Comunicação, localizado na Universidade de Brasília, campus Darcy Ribeiro. A topologia desta rede é mostrada na Figura 3.14. Esta topologia foi escolhida considerando a possibilidade de expansão para uma rede de sensores maior, do tipo árvore ou *mesh*, como foi explicado na Seção 2.3.4.

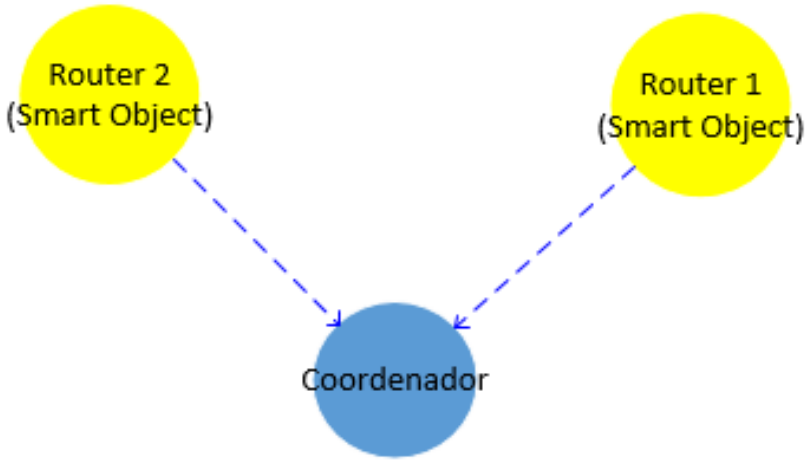

Figura 3.14: Topologia inicial utilizada para realização das simulações

Essencialmente, existem dois dispositivos configurados como *Router* e cada um conectado a um Arduino Nano com um sensor DHT11 (sensor de umidade e temperatura), que envia periodicamente os valores de umidade e temperatura local para o nó Coordenador que está conectado a um Raspberry Pi, a montagem dos circuitos é mostrada nas Figuras 3.16 e 3.15.

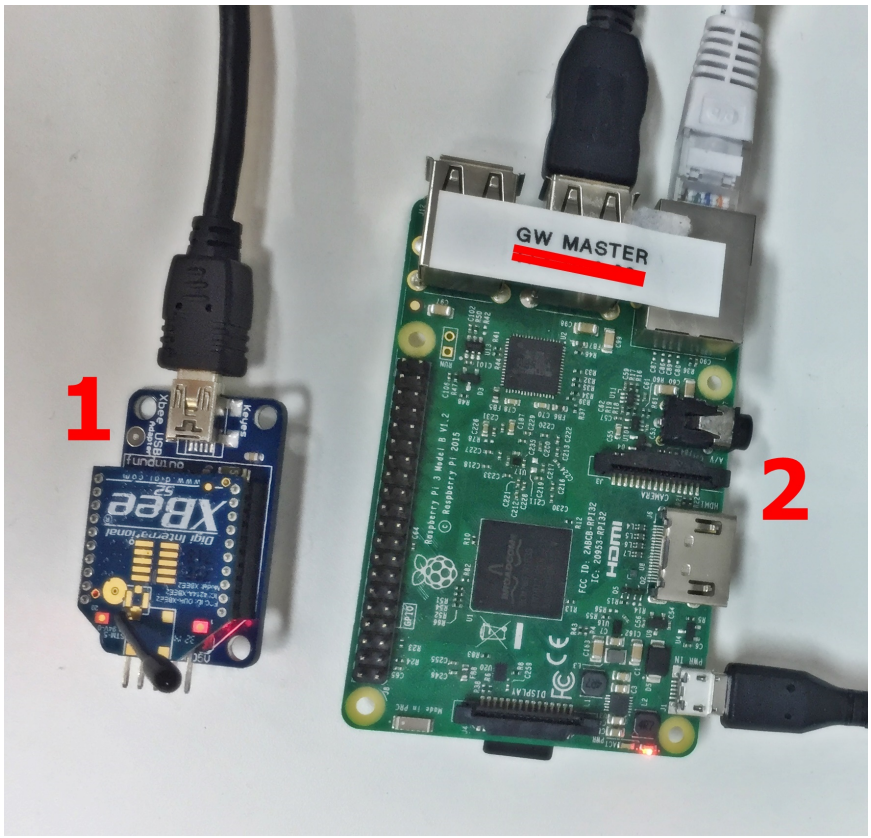

Figura 3.15: Dispositivo Zigbee (1) ligado ao um Raspberry Pi (2), que atua como Coordenador da rede Zigbee do Laboratório UIoT.

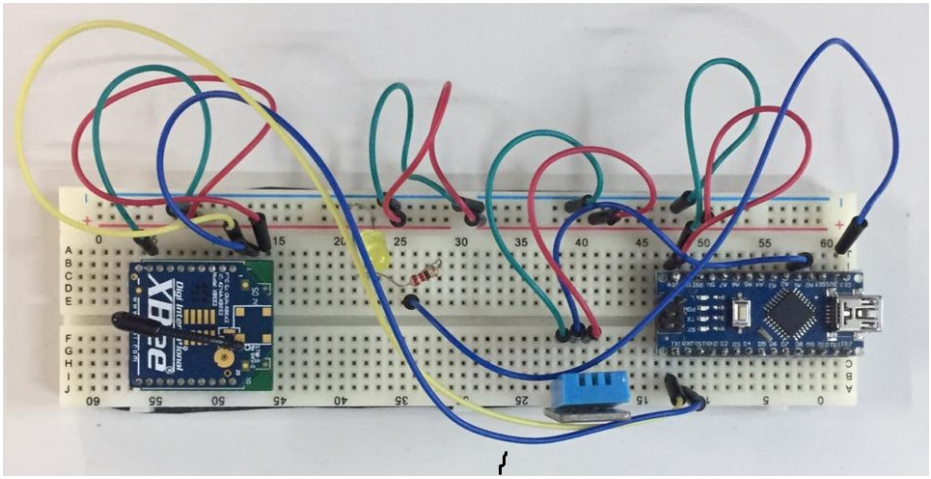

Figura 3.16: Exemplo da implementação de um *Smart Object*, utilizado na rede ZigBee, desenvolvida por (MAIA, 2017)

A partir desta rede de sensores inicial, foram definidos quatro cenários que serão detalhados no Capítulo 4, correspondente à explicação da Análise dos Resultados obtidos durante os experimentos para exploração da segurança do protocolo ZigBee.

Capítulo 4

Análise e Resultados

Neste capítulo, serão apresentados os resultados obtidos e, em seguida será feita uma análise referente a cada cenário proposto. Foram considerados quatro cenários. No primeiro, temos a rede inicial, tal como foi implementada no Laboratório UIoT, cuja topologia é mostrada na figura 3.14. No segundo cenário, é utilizado um *sniffer* para obter informações da rede. O terceiro cenário refere-se à introdução de um dispositivo adversário na rede, configurado com as informações que foram obtidas através do *sniffer*. Por fim, o quarto cenário sugere uma forma de utilização do USB Dongle CC2531 como forma de monitorar a rede ZigBee.

4.1 Cenário 1

Neste cenário considera-se a rede implementada no laboratório e supõe-se que se quer obter informações da mesma, a fim de explorar alguma falha de segurança. Nesse contexto, analisou-se quais seriam as possibilidades para atingir esta meta.

Vale lembrar que, para obtenção dos dados deste e dos outros cenários, foi requerido o acesso à rede e aos dispositivos presente no laboratório. Conforme a topologia mostrada na Seção 4.1, o diagrama de comunicação da rede funciona seguindo as etapas descritas na Figura 4.2.

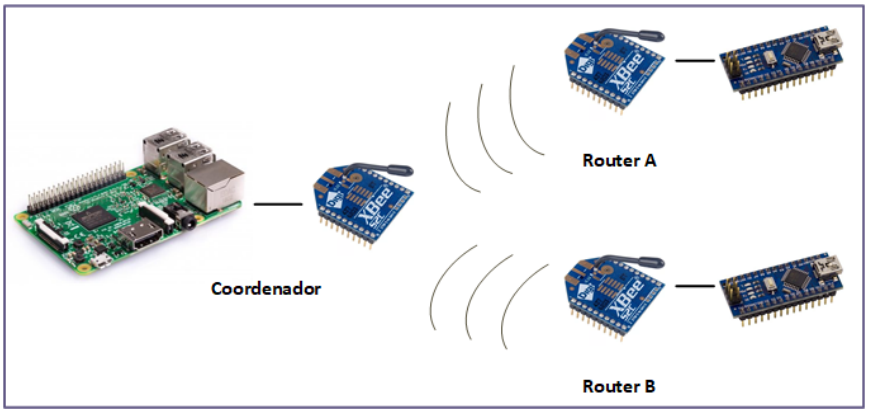

Figura 4.1: Topologia utilizada no cenário 1.

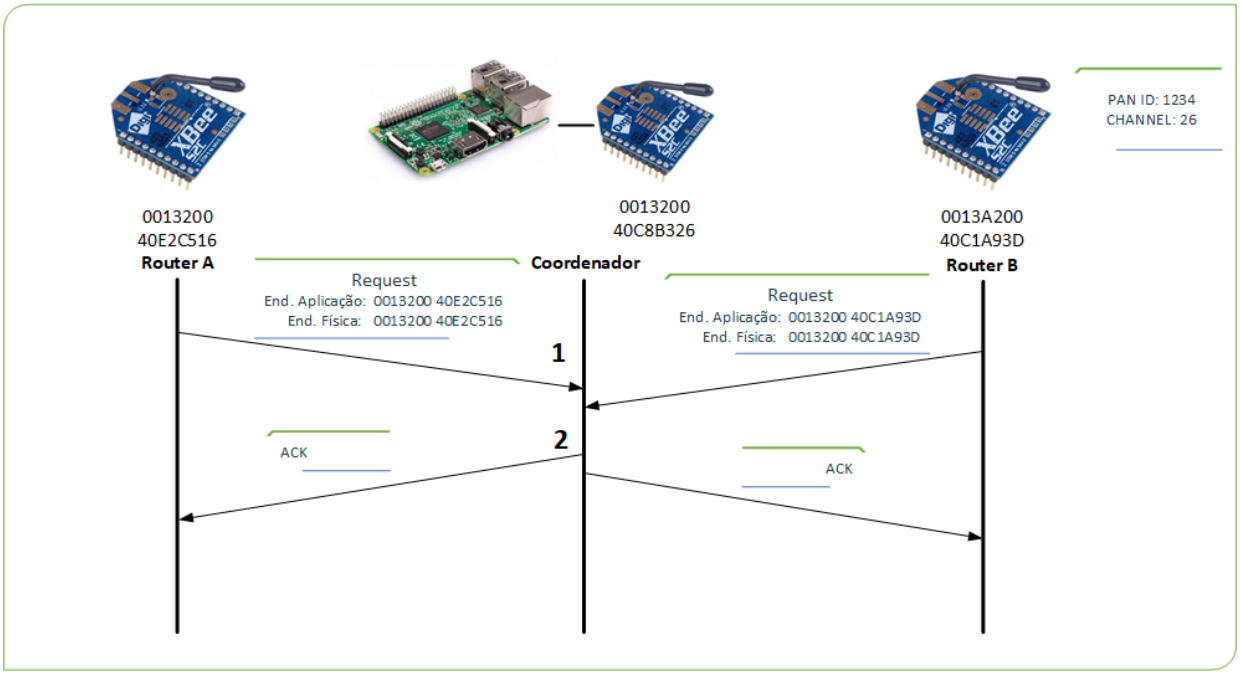

Figura 4.2: Diagrama de sequência do cenário 1.

A Figura 4.2 refere-se ao diagrama de sequência que ocorre na comunicação entre os dispositivos da rede. Inicialmente, todos os dispositivos devem estar configurados para o mesmo PAN ID. Uma vez configurados, os *routers* enviam *frames* de dados com **ACK REQUEST** para o coordenador, que por sua vez, ao receber a mensagem, envia um **ACK** confirmando o recebimento.

Observando a Figura 4.2, em **1**, os módulos XBee configurados como *router* enviam um *frame* com os dados coletados do sensor DHT11, como foi descrito em (MAIA, 2017), a aplicação considera o endereço que está escrito no corpo da mensagem, em forma de *string*, pois a aplicação não foi desenvolvida de forma a ler o cabeçalho das mensagens recebidas via ZigBee. No entanto, para fins de comunicação, o **ACK** enviado, em **2**, é encaminhado considerando o endereço **MAC** de origem conforme está no cabeçalho do frame, nos próximos cenários será explicado com mais clareza porque essa solução pode tornar a rede vulnerável.

Para um melhor entendimento do diagrama referente à Figura 4.2, tem-se a captura dos dados coletados via *PuTTY* do dispositivo coordenador. Esta captura do cenário inicial, mostra quais dispositivos XBee (*routers*) estão conectados à rede e quais as mensagens que estão sendo enviadas a ele.

Pela figura 4.3 pode-se ver o tipo de mensagem enviada e o ID dos dispositivos que estão transmitindo dados. Essas mensagens são enviadas como uma *string* com valores separados por ponto e vírgula e, são enviados da seguinte maneira:

ROUTER 1 -> `0013A20040E2C516;FFFF;ZB_METEOROLOGIA;MTR_TEMP_DHT11;25.00;30;0;789`

ROUTER 2 -> `0013A20040C1A93D;FFFF;ZB_DHT11;ZB_HUM;33.00;5;0`

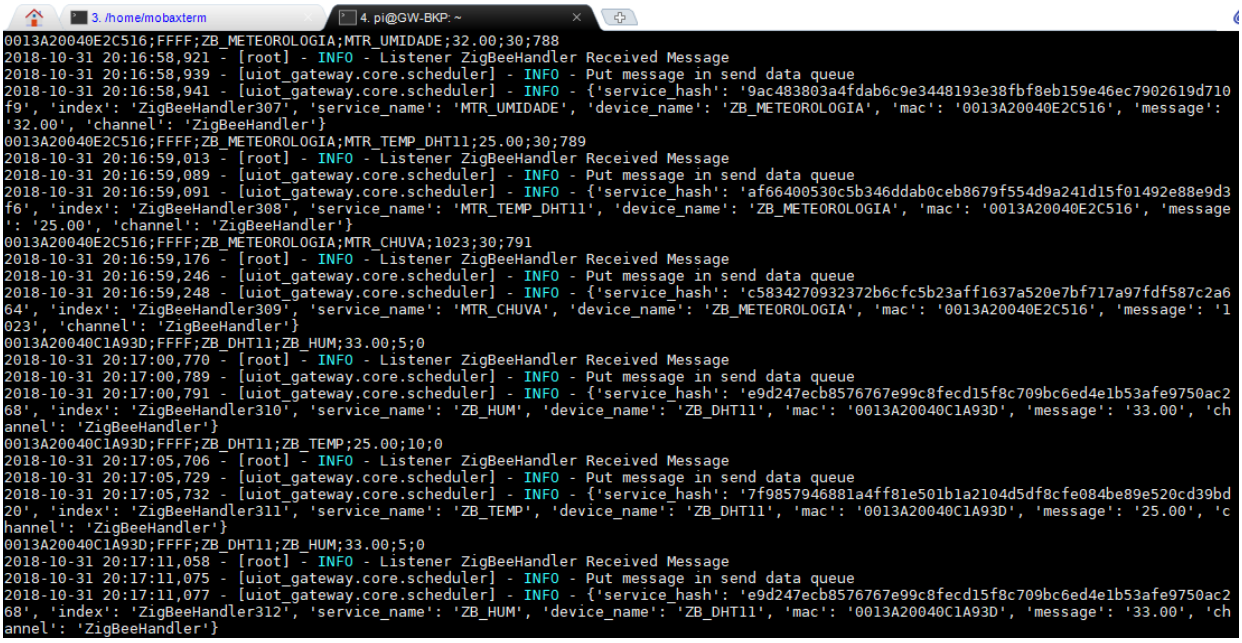

Figura 4.3: Captura da informação dos roteadores coletada pelo gateway.

Como foi explicado na Seção 2.3.5 sobre o funcionamento do protocolo ZigBee, sabe-se que para que haja comunicação entre os dispositivos, eles devem estar não só na mesma rede (com o mesmo PAN ID), mas também no mesmo canal. De acordo com as pesquisas feitas para este trabalho, a maneira encontrada para alcançar este objetivo seria utilizando um hardware específico para esta função.

Encontrou-se, portanto, alguns dispositivos com a função de *sniffer* para redes ZigBee.

1. **USB Dongle CC2531** Este dispositivo atua como analisador de pacotes Zigbee, ele é um dispostivo USB totalmente operacional que pode ser conectado a um PC. O *dongle* tem dois LEDs, dois pequenos botões e furos de conectores que permitem a conexão de sensores ou outros dispositivos. Além disso, ele também tem conector para programação e depuração do controlador USB CC2531. A vantagem deste dispositivo é que ele vem pré-programado com uma *firmware* de tal forma que pode ser usado como um *sniffer* de pacotes. O CC2531 oferece suporte extensivo de hardware para manuseio de pacotes, *buffer*, criptografia e autenticação de dados, possui avaliação clara de canal, indicação de qualidade de *link* e informações de tempo de pacote. Ele trabalha na banda de frequência de 2,4GHz, e canal de 16 transmissões. É operado em sistemas de 32 bits e funciona com o SmartRF Packet Sniffer, para conseguir a captura de dados Zigbee.

2. **Atmel RZ Raven USB stick (RZUSBstick)** Este dispositivo USB 2.0 inclui suporte para o protocolo IEEE 802.15.4 a 2,4 GHz com um microprocessador AVR integrado. Com a *firmware* original ele funciona como um *sniffer* para o protocolo ZigBee, no entanto, ele teve sua produção descontinuada (WRIGHT; CACHE, 2015).

3. **Q51 802.15.4 PANalyzer** Este *sniffer* captura pacotes em uma rede 802.15.4 e tem suporte

para ser utilizado junto ao *Wireshark*. O Q51 tem uma interface Ethernet, não USB, como os outros dispositivos apresentados. No entanto, seu custo-benefício é menor em relação aos anteriores, de forma que torna-se inviável a sua utilização para o presente estudo.

Considerando as funcionalidades dos dispositivos, optou-se por empregar neste trabalho, o USB Dongle CC2531, pois o mesmo satisfaz as necessidades deste projeto, além de possuir um maior custo-benefício em relação aos outros apresentados. Com isso, seguiu-se para o cenário 2, onde será empregado o dispositivo adquirido.

4.2 Cenário 2

O segundo cenário considera a existência do *sniffer* escolhido para a prática das simulações. Neste caso, tem-se a rede já implementada como mostrada na Figura 3.14 e então utiliza-se o USB Dongle CC2531 em conjunto com o software SmartRF Packet Sniffer para escutar o que está sendo transmitido pela rede.

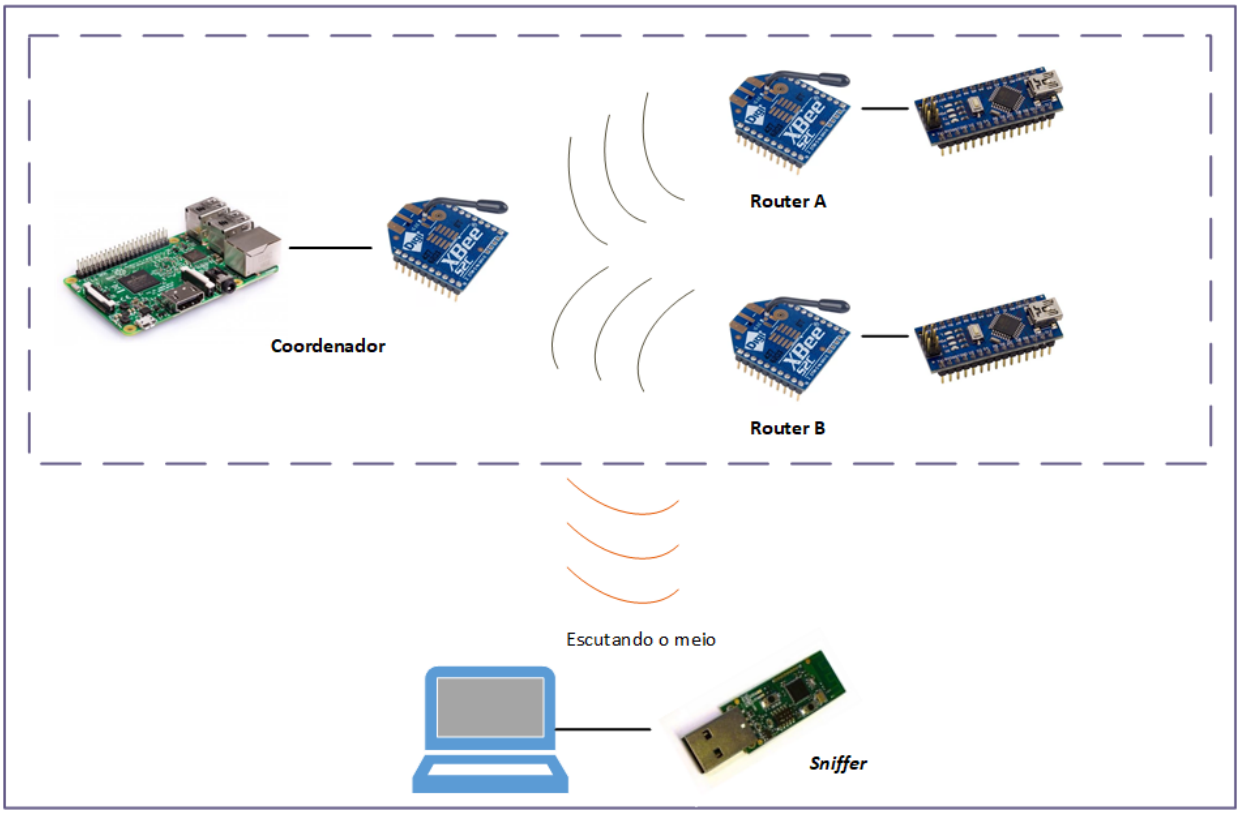

Figura 4.4: Topologia utilizada no cenário 2

Na Figura 4.5, observa-se o diagrama de sequência do cenário 2, onde foi introduzido o *sniffer* USB Dongle CC2531. Em comparação com a Figura 4.2, percebe-se que a comunicação entre os dispositivos não foi alterada, a linha tracejada significa que o *sniffer* está recebendo os *frames*, mesmo que eles estejam endereçados apenas para o coordenador, ou seja, nenhum dos dispositivos na rede consegue rastrear a sua presença neste cenário.

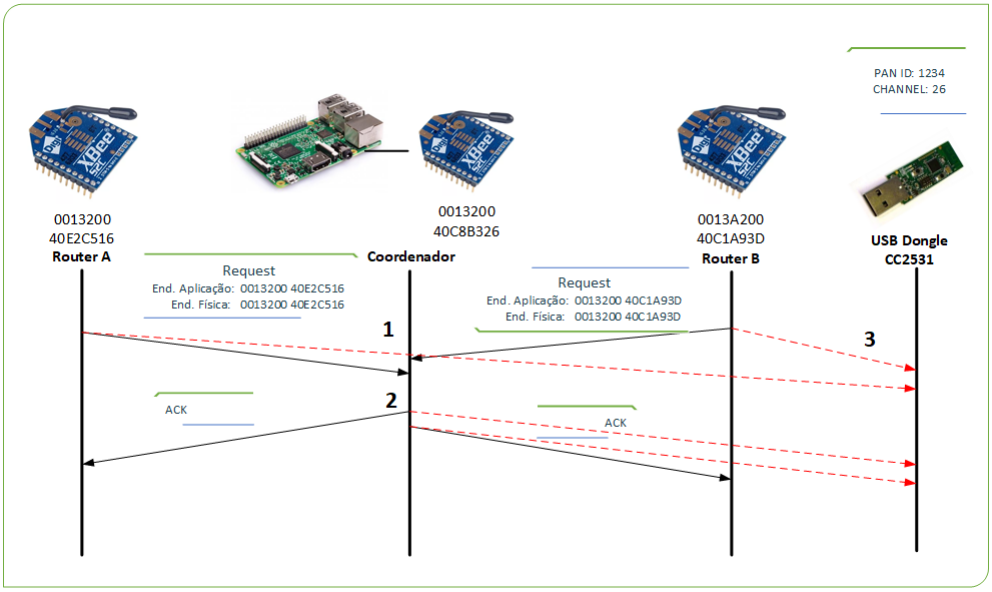

Figura 4.5: Diagrama de sequência do cenário 2.

Em um primeiro momento foi utilizado o 802.15.4 monitor (MITSHELL, 2016) para fazer uma varredura em todos os canais, no entanto, conforme pode ser visto na Figura 4.6, este software não traz tantas informações úteis como o SmartRF Packet Sniffer. Sua vantagem em relação ao Packet Sniffer é a capacidade de fazer uma varredura em todos os canais possíveis em vez de fazer com que o usuário escolha apenas um dos canais por vez, o que torna-se uma importante funcionalidade, visto que, de outra maneira, isto seria feito de forma manual, uma solução nada eficiente considerando a possibilidade de 16 diferentes canais.

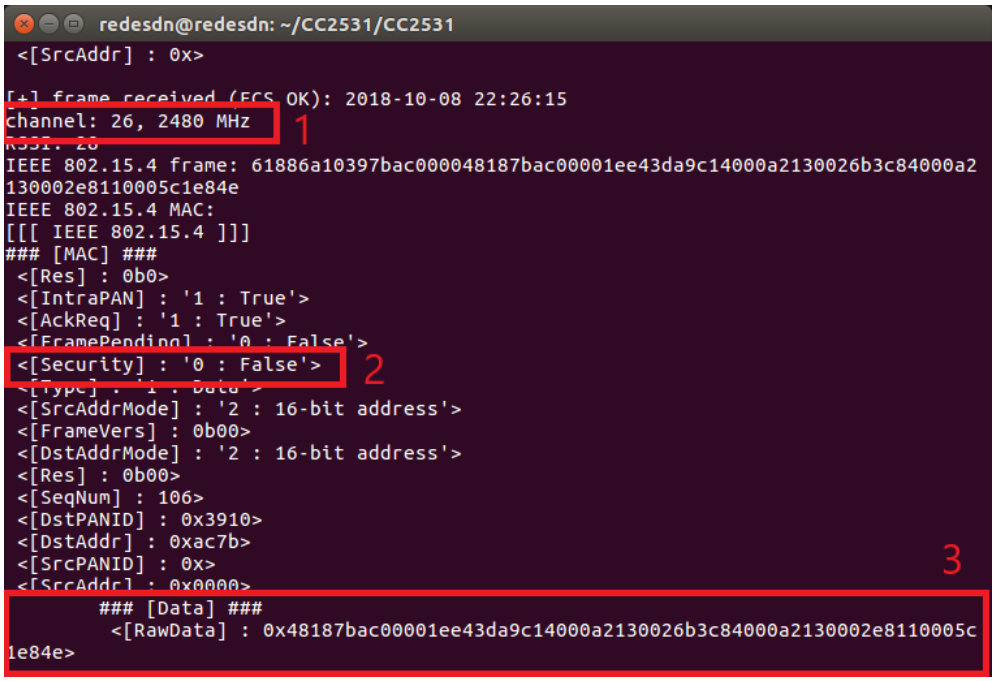

Figura 4.6: Captura feita utilizando o 804.15.4 Monitor no sistema operacional Ubuntu.

Os dados coletados com o 802.15.4 Monitor, são obtidos utilizando o comando:

```
python sniffer.py -p 10
```

O parâmetro `-p 10` indica que o software deve escutar por 10 segundos cada canal entre 11 e 26. A partir da utilização deste *software*, obtemos algumas informações úteis, entre as mais importantes:

1. A comunicação está acontecendo no canal 26;
2. Não há medidas de segurança presentes no *frame*;
3. O corpo da mensagem aparece no formato hexadecimal:
   `0x481800007bac1e0026b3c84000a213003da9c14000a2130040e8110005c1e829303031334132303034304331413933443b464646463b5a425f44485431313b5a425f54454d503b32392e30303b31303b30`

   ao converter esse valor para o formato de texto, obtém-se a seguinte *string*:
   `H#{¬#&³È©¢#=©Á©¢#©è##Áè)0013A20040C1A93D;FFFF;ZB_DHT11;ZB_TEMP;29.00;10;0`
   dessa forma, pode ser observado que a mensagem enviada consiste em uma *string* com valores separados por **;** (ponto e vírgula), os caracteres que precedem o valor da *string*, correspondem ao *header* do pacote encapsulado.

Em seguida, passou-se a empregar o Packet Sniffer para coleta de dados, a Figura 4.7 mostra as informações obtidas.

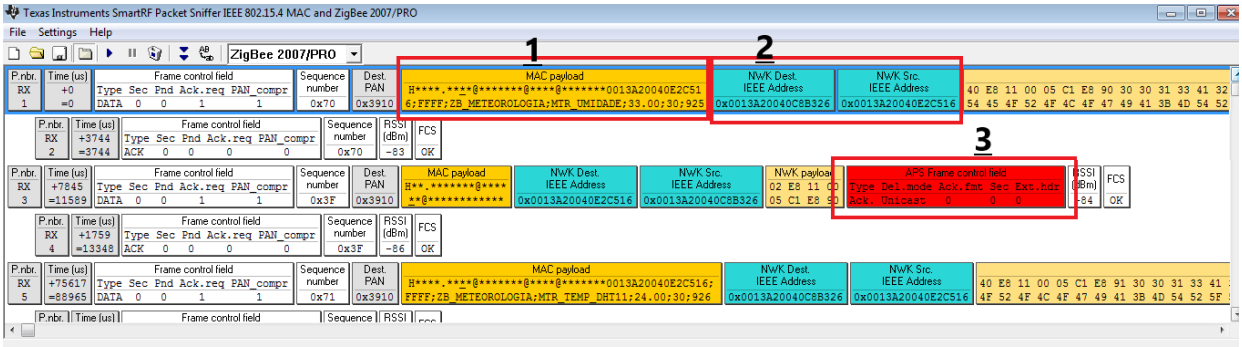

Figura 4.7: Captura executada utilizando o *software* Packet Sniffer

Sabendo que o canal onde havia a comunicação entre os dispositivos da rede Zigbee, então a captura já foi configurada para escutar o canal 26, com as informações coletadas, é possível visualizar os seguintes dados, muito úteis para compreender o funcionamento da rede.

1. O *MAC payload* compreende a mensagem enviada pela aplicação e o *header* da camada superior, como pode ser visualizado na Figura 4.7, em **1**, percebeu-se um padrão no texto das mensagens.

2. Uma informação muito importante que foi possível coletar, é o endereço de origem e destino dos dispositivos, como foi explicado na Seção 3.1.3 do Capítulo 3, este endereço individualiza cada dispositivo.

3. O campo *APS Frame control field* contém informações como o Tipo, se é um **ACK** ou **DATA**, e qual é o modo de transmissão, se é **Unicast** ou **Multicast**.

Uma observação a ser feita neste tópico que é de grande valia para a análise do trabalho e, também para os cenários posteriores é que, utilizando o dispositivo USB Dongle CC2531, conseguiu-se obter uma comunicação Zigbee tanto em distribuições linux, como no sistema operacional Windows 7. A diferença entre eles é que, enquanto no linux é feita uma varredura dos canais para detectar em qual canal dentre os disponíveis está ocorrendo a comunicação. No Windows, por sua vez, utilizando o software Packet Sniffer, é atribuído um canal, ou seja, o usuário tem a opção de selecionar em qual canal deseja analisar a comunicação.

Um grande diferencial neste projeto, foi que a princípio não era conhecido o canal em que estava ocorrendo a comunicação Zigbee presente no Laboratório UIoT na UnB e, ao utilizar o 804.15.4 Monitor, descobriu-se que a comunicação presente naquele ambiente pertencia ao canal 26. Descoberto isso, preferiu-se utilizar o software SmartRF Packet Sniffer disponibilizado pela (TEXAS INSTRUMENTS, 2014), que possui um design e uma maior abrangência de informações e dados que não encontramos no 804.15.4 Monitor.

4.3 Cenário 3

No terceiro cenário, foi feita uma análise dos padrões de mensagens trocadas entre os dispositivos da rede Zigbee, percebeu-se que os dispositivos com a função de *Router* estavam enviando, com certa periodicidade, uma string com valores separados por **;** (ponto e vírgula). Na seguinte ordem:

```
0013A20040C1A93D;FFFF;ZB_DHT11;ZB_TEMP;29.00;10;0
```

1. O primeiro valor se tratava do mesmo número serial do dispositivo de origem, que corresponde ao endereço MAC (64 bits) para os módulos XBee. Para este campo, encontrou-se dois valores diferentes: 001320040E2C516 e 001320040C1A93D.

2. Em seguida, observa-se o valor **FFFF** que se refere ao *My Address* de 16 bits de cada dispositivo, no entanto, estão todos com o valor **FFFF**, foi feito assim, pois se está considerando o endereço de 64 bits para distinguir cada nó.

3. O próximo campo, tem o valor **ZB_DHT11** ou **ZB_METEOROLOGIA**, infere-se que este campo seria o nome do objeto, sabe-se que DHT11 é um sensor de umidade e temperatura que permite fazer leituras de temperaturas entre 0 e 50° Celsius, e de umidade [1] entre 20% e 90%, usado para projetos com Arduino.

[1] Umidade relativa do ar é razão entre a quantidade de água existente no ar (umidade absoluta) e a quantidade máxima que poderia haver na mesma temperatura (ponto de saturação), por isso é expressa em %.

4. O quarto campo insere o nome do serviço, foram coletados três diferentes valores: ZB_TEMP (para a temperatura), ZB_HUM (para a umidade) e MTR_LUMINOSIDADE.

5. Os três últimos valores da *string* são números. Observando o tipo e os valores coletados, sabendo o nome dos serviços e do objeto, ou seja, o sensor DHT11, entendeu-se que esses campos, correspondem respectivamente ao valor medido (de temperatura, umidade ou luminosidade), à periodicidade de atualização do valor medido (em segundos) e por último, um índice, que faz parte do padrão de *string* de auto-registro estabelecido pelo UIoT, que foi abordado em (MAIA, 2017).

Por fim, constatou-se que o significado de cada um desses campos, corresponde à estrutura da *string* de auto-registro cuja estrutura está descrita na Figura 4.8, e que foi proposta conforme o trabalho realizado por (MAIA, 2017), ao implementar a rede Zigbee no Laboratório UIoT na UnB.

"ENDEREÇO MAC XBEE(64 bits);	MY ADDRESS (16 bits);	NOME DO OBJETO;
SERVIÇO;	VALOR MEDIDO;	TEMPO DE ATUALIZAÇÃO;	ÍNDICE"

Figura 4.8: Estrutura da *string* de auto-registro proposta por (MAIA, 2017)

A partir dessas informações, foi configurado um outro dispositivo Zigbee, para simular um possível ataque a esta rede. Em outras palavras, introduziu-se um outro módulo Xbee configurado como *router*, conectado a um Arduino que monta a *string* de auto-registro seguindo o padrão adotado na rede implementada.

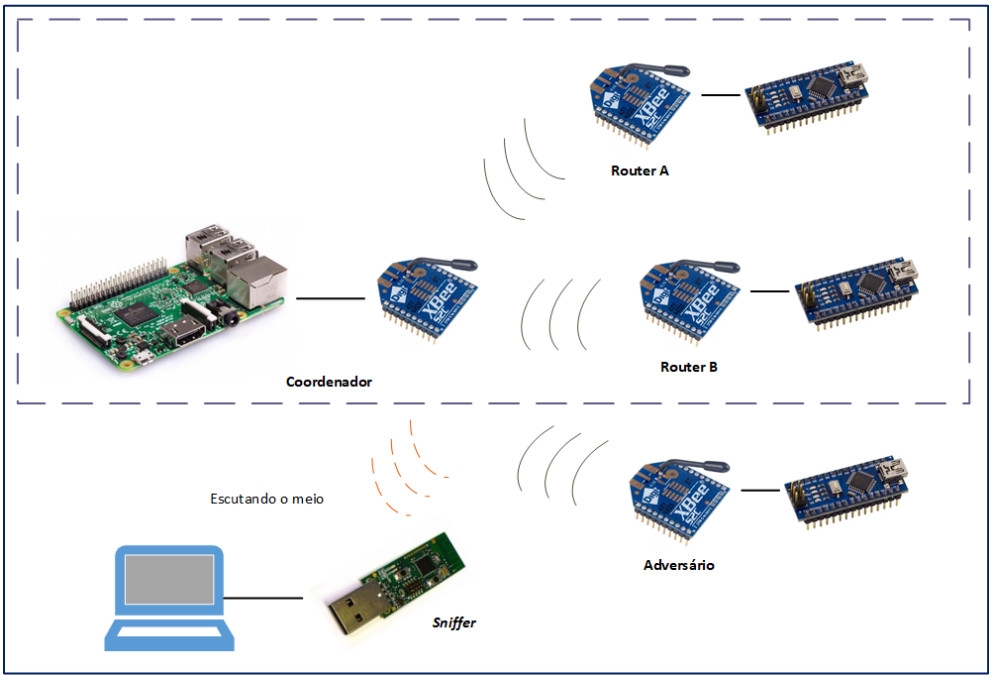

Figura 4.9: Topologia utilizada no cenário 3

Com isso, a topologia em funcionamento, passou a ser a que é apresentada na Figura 4.9 como

cenário 3, onde a topologia que está dentro do retângulo tracejado é aquela que foi configurada para utilização no laboratório e em que os dispositivos contidos nela são conhecidos pelo gerente da rede. Fora do retângulo tracejado está o *sniffer* atuando como uma espécie de "espião" e um novo módulo XBee atuando como "Adversário", ambos desconhecidos dos responsáveis pela rede.

O diagrama de sequência para este terceiro cenário ficou conforme a Figura 4.10. Para fins de simplicidade e clareza, nesta representação não será incluído o dispositivo *sniffer* USB Dongle CC2531, pois a sua atuação já foi explicada no cenári anterior. Além disso, será utilizado somente o Router B como exemplo para o diagrama de sequência, pois apenas o seu endereço MAC que foi "clonado" ao nível da camada de aplicação para a realização do ataque de segurança.

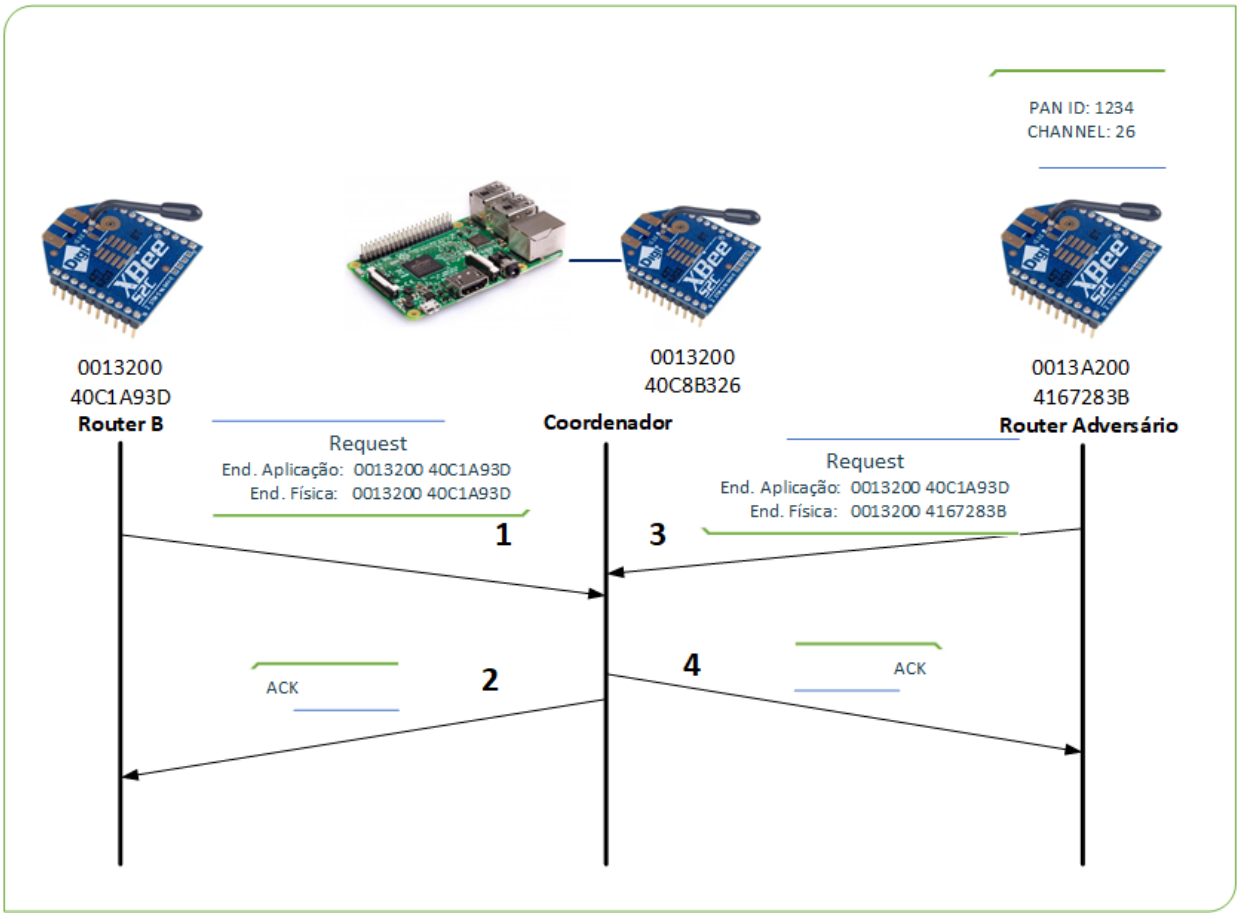

Figura 4.10: Diagrama de sequência do cenário 3.

Cada parte da Figura 4.10 será explicada conforme os itens a seguir:

1. O *Router B* envia uma mensagem ao coordenador seguindo o padrão descrito no início desta seção, naturalmente, ele envia como endereço MAC, o seu endereço real. Contudo, o *Router Adversário*, não age da mesma forma, como será explicado mais adiante.

2. Em seguida, como esperado, o nó *Coordenador* envia uma mensagem de **ACK** ao *Router B*.

3. O *Router Adversário*, por sua vez, gera uma *string* com o formato em que o nó coordenador

está preparado para receber, e no campo destinado ao endereço MAC, ele coloca o endereço do *Router B*.

4. Observe no diagrama de sequência que a mensagem de **ACK**, neste caso, é enviada para o *Router Adversário*, apesar de ter recebido, na *string*, o endereço do *Router B*. Isto já era esperado, visto que a comunicação Zigbee ocorre na camada MAC, por esse motivo, o endereço considerado é o que está no cabeçalho do *frame*. Portanto, o ataque descrito, não configura um ataque na camada física, mas sim na camada de aplicação, onde foi encontrada uma vulnerabilidade em sua implementação, a partir dos dados obtidos na captura com o USB Dongle CC2531.

Analisando cada item de forma mais detalhada, observa-se a Figura 4.11, que apresenta o *log* das mensagens recebidas pelo nó coordenador. Ao receber a *string* enviada pelos nós roteadores, o coordenador monta um *json* para que seja utilizado em sua aplicação.

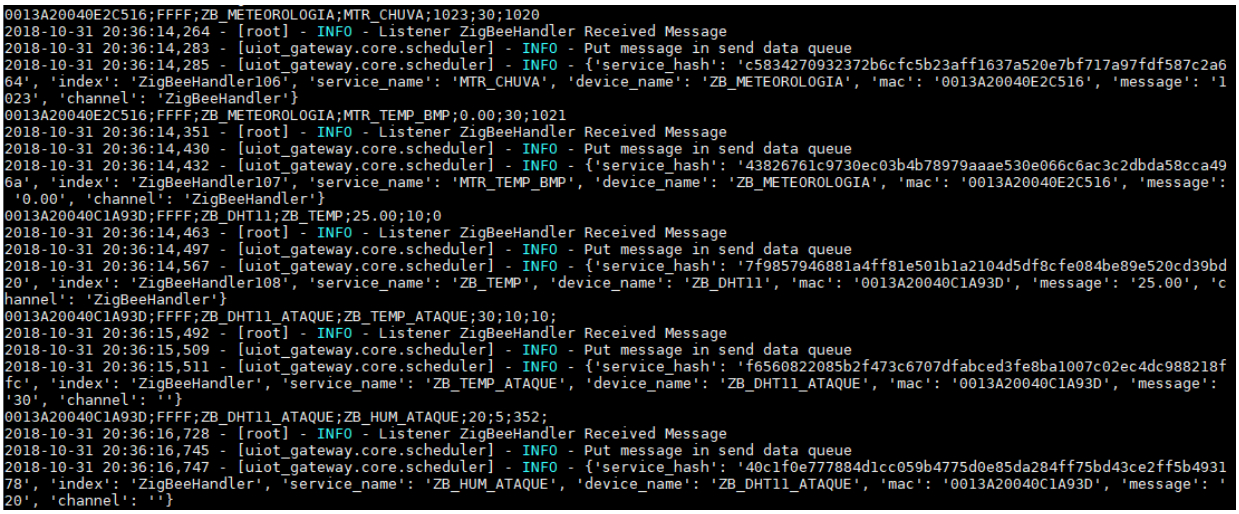

Figura 4.11: Captura da informação dos roteadores e atacante coletada pelo gateway.

Estas informações foram obtidas, a partir do acesso concedido pelos administradores da rede Zigbee do Laboratório UIoT, do curso de Engenharia de Redes de Comunicação, localizado na Universidade de Brasília, portanto, não foram coletadas através do USB Dongle CC2531. O acesso direto ao nó coordenador foi solicitado para que fosse possível verificar se as informações coletadas eram relevantes no contexto adotado.

Para que fosse possível visualizar, no momento da coleta de dados, qual seria a mensagem enviada pelo *Router Adversário*, o nome dos serviços enviados pelo mesmo, foram configurados como ZB_HUM_ATAQUE e ZB_TEMP_ATAQUE.

Exemplo de *string* enviada pelo *Router Adversário*:

```
0013A20040C1A93D;FFFF;ZB_DHT11_ATAQUE;ZB_TEMP_ATAQUE;30;10;10
```

Exemplo do *json* estruturado pela aplicação presente no nó coordenador:

```
'service_hash': '40c1f0e777884d1cc059b4775d0e85da284ff75bd43ce2ff5b493178', 'index':
```

'ZigBeeHandler', 'service_name': 'ZB_HUM_ATAQUE', 'device_name': 'ZB_DHT11_ATAQUE', 'mac': '0013A20040C1A93D', 'message': '20', 'channel': ''

Como pode ser visualizado na Figura 4.12, é possível obter tanto o *payload* do *frame*, quanto seu cabeçalho, dessa forma, é possível comparar o endereço MAC de origem do *header* com o que é enviado na mensagem, diferentemente do que é visualizado na aplicação do nó coordenador, onde visualiza-se apenas as informações da *string*. Portanto, constatou-se que sem o *sniffer* o nó coordenador, não sabe diferenciar quais nós realmente fazem parte da rede e quais são intrusos.

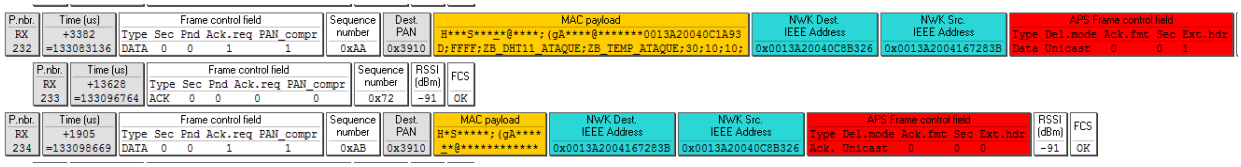

Figura 4.12: Captura da informação dos roteadores e atacante coletada pelo USB Dongle CC2531.

Ao analisar o impacto deste tipo de ataque, pode-se imaginar como afetaria um *Datacenter*, por exemplo, que possui sensores deste tipo, como temperatura, umidade ou também de porta aberta, por exemplo, e que é de grande importância controlar cada um desses elementos. Nesses casos, se fossem enviados dados falsos ao coordenador desta rede, possivelmente, o responsável pela rede não perceberia em tempo hábil esta intrusão, afinal, nem o nó coordenador da rede está apto, neste caso, a decidir quais são os dados confiáveis. Diante de um contexto como esse, o sucesso no ataque à rede ZigBee poderia ser a porta de entrada para ocasionar danos maiores no ambiente em questão.

4.4 Cenário 4

Este cenário visa explicar a automatização dos dados coletados, através das informações obtidas mediante a análise da rede Zigbee juntamente com o dispositivo *USB Dongle CC2531*. De posse das informações adquiridas, surge um questionamento bastante relevante a ser tratado neste trabalho. De que maneira, o dispositivo USB Dongle CC2531, em conjunto com o software SmartRF Packet Sniffer, podem ser usados de forma benéfica para alertar o gerente da rede sobre possíveis ataques e intrusos na rede?

Mediante este questionamento, foi criada uma proposta rápida e a curto prazo, de forma a alertar e lembrar o gerente da rede para verificar os dados coletados durante o dia. Dessa forma, o dispositivo traria um novo significado à rede, pois ao invés de agir como um espião, passaria a ser utilizado como um ajudante da rede, detectando assim, novos usuários para alertar o gerente sobre possíveis ataques.

Neste contexto, considera-se que toda a informação obtida através deste poderoso dispositivo seria encaminhada via *Telegram* ao responsável pela rede ZigBee. Neste caso, como a coleta do tráfego da rede seria feita ininterruptamente, ao menos um vez por dia (isto pode ser ajustado a cada necessidade), uma notificação seria encaminhada como forma de lembrete para que o gerente

verifique a última captura salva. Uma ilustração deste cenário é apresentado na Figura 4.13.

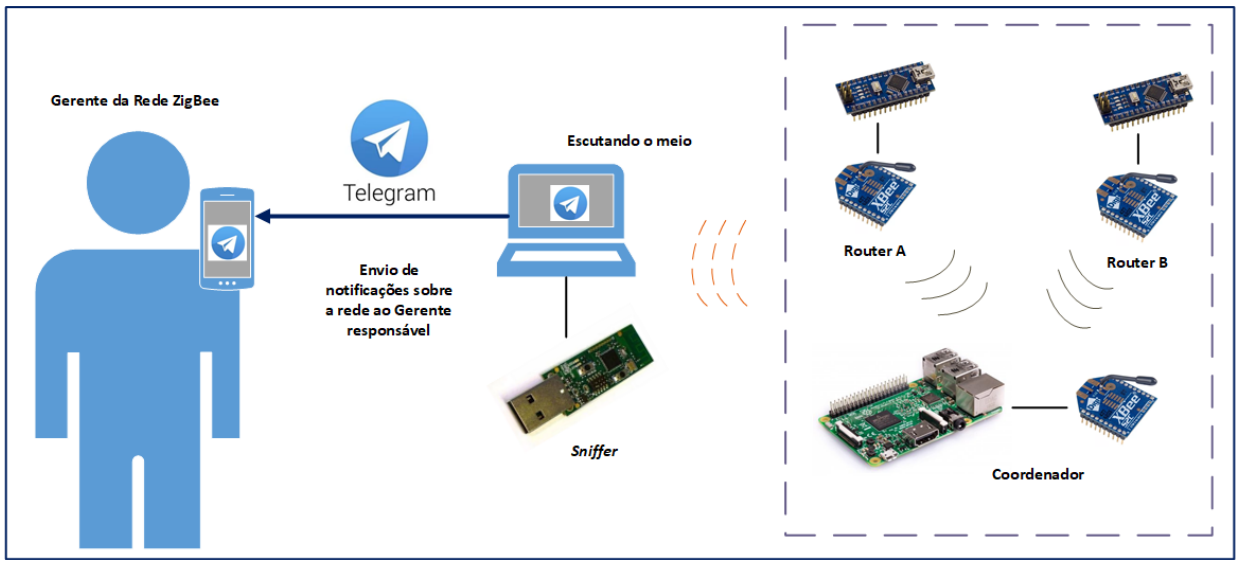

Figura 4.13: Topologia final utilizada para implementação da proposta do sistema de monitoramento na camada MAC.

Levando em conta que o processo descrito trata-se da repetição das mesmas tarefas diariamente em um horário específico, propôs-se uma automatização para este processo. Haja vista que o sistema operacional utilizado é o *Windows 7*, aproveitou-se de alguns de seus recursos para a realização desta proposta.

A linguagem adotada para a preparação do *script* foi *vba*, devido à sua fácil execução e possibilidade de aplicação para outros casos. Além disso, o *Agendador de Tarefas do Windows* foi utilizado para que o *script* fosse executado no mesmo horário todos os dias.

Ao preparar a automatização para a utilização desse *software* de coleta de informações de tráfego, foram considerados alguns passos a se observar. Primeiramente, considera-se que o *software* SmartRF Packet Sniffer está sempre em funcionamento, a fim de captar todo o tráfego na rede. Então o primeiro passo é pausar e salvar a captura em determinado horário, combinado entre os responsáveis pela rede.

Logo em seguida, reinicia-se a captura para que nada seja perdido. Tendo salvo o arquivo em um diretório específico, utiliza-se o *Telegram* para enviar uma notificação ao gerente, contendo o caminho do diretório onde foi salvo e o arquivo gerado em anexo.

Em resumo, os passos seguidos foram:

1. Pausa a captura;
2. Salva a captura;
3. Reinicia a captura;
4. Coloca o arquivo salvo, com a data e hora atual no nome, em um diretório específico no

servidor, conforme a Figura 4.14;

5. Envia uma mensagem via *Telegram* ao gerente da rede, mostrado na Figura 4.15.

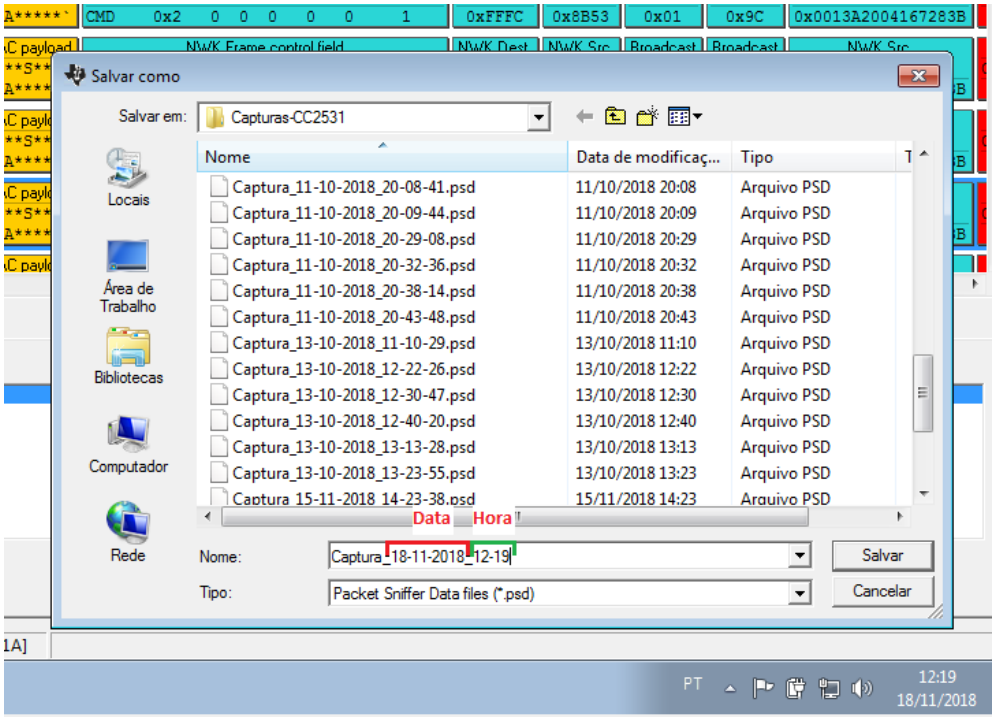

Figura 4.14: A captura é salva com a data e hora atuais em um diretório no servidor.

A mensagem automática via *Telegram* para o gerente da rede é enviada através do *browser* no servidor em que está sendo executado o *software* SmartRF Packet Sniffer. Após o envio da notificação, o próprio *script* finaliza todos os processos exceto a coleta dos *frames* ZigBee, na próxima execução programada do *script* de automatização, estas tarefas se repetirão sem que seja necessária qualquer intervenção humana.

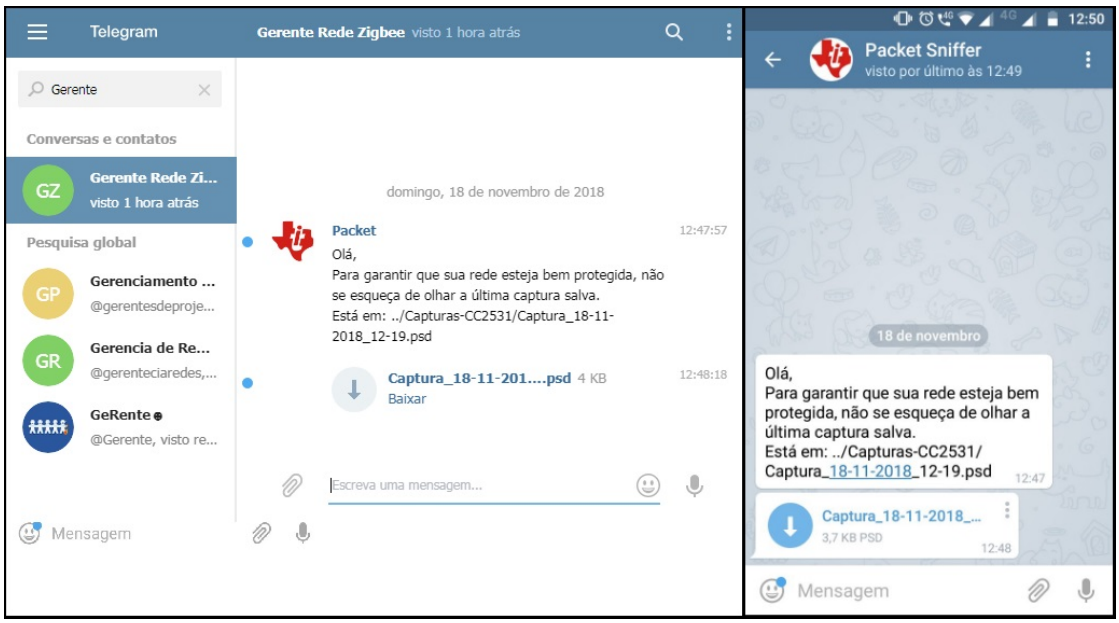

Figura 4.15: O arquivo salvo é enviado com uma notificação ao *Gerente da Rede Zigbee*.

Finalmente, entende-se como ponto alto deste trabalho, a possibilidade de ter um controle também sobre as camadas mais baixas da pilha do protocolo ZigBee, o que se quer ressaltar aqui, não é somente o *script* da forma como foi realizado, pois o mesmo poderia ser desenvolvido de outras maneiras com as mesmas funções. O mais importante é garantir que o gerente da rede possa ter uma maior segurança de que todos os dispositivos associados ao nó Coordenador, são realmente os dispositivos configurados pelos desenvolvedores do laboratório para este fim, e que ele possa tomar as medidas necessárias ao detectar algum nó intruso.

4.5 Cenário 5

Neste cenário propõe-se mais um mecanismo de segurança para garantir a privacidade das informações trafegadas, utiliza-se da criptografia de dados para impedir que qualquer dispositivo *sniffer* consiga obter a informação que está sendo enviada. Apesar de conseguir coletar o *frame*, não será possível ler o *payload*, onde estão as informações que se quer proteger. Nesse caso, ainda estariam visíveis as informações de cabeçalho do *frame*, ou seja, os endereços MAC de origem e destino, O PAN ID e o canal em que ocorre a comunicação. Contudo, embora estas sejam as informações para tentar conectar-se à rede, sem a chave de segurança, é inviável estabelecer uma conexão junto ao coordenador.

Dessa forma, é de fundamental importância a solução aqui apresentada, visto que trata-se de mais um obstáculo em que o atacante deve superar para acessar as informações enviadas entre os dispositivos.

Para ativar a segurança em um módulo XBee, o parâmetro *Encryption Enable* (EE) deve ser definido como 1. Quando o valor do parâmetro é alterado, o módulo XBee deixa a rede (PAN ID

e canal) em operação e tenta formar ou ingressar em uma nova rede. Se **EE** for definido como 1, todas as transmissões de dados serão criptografadas com a chave de rede. É importante ressaltar que se a chave de segurança **KY** (*Encryption Key*) for configurada para o valor padrão "0" (zero), o coordenador enviará uma chave aleatória não criptografada no momento da conexão, por isso, esta opção não é recomendada. O ideal é configurar todos os dispositivos com a mesma chave.

As Figuras 4.16 e 4.17 apresentam as definições para o nó coordenador e para o roteador, respectivamente. é importante observar que para o coordenador existe um parâmetro extra, a chave de rede **NK** (*Network Encryption Key*), ela define a chave usada para criptografia e descriptografia de rede. Se definido como 0 (padrão), o coordenador seleciona uma chave de criptografia de rede aleatória (recomendada). Caso contrário, se for definido **NK** para um valor diferente de zero, ele usa esse valor como chave de segurança de rede.

O parâmetro **NK** é suportado apenas no coordenador. Roteadores e dispositivos finais com segurança ativada (*EE = 1*) adquirem esta chave quando ingressam na rede. Eles recebem a chave de rede criptografada com a chave de link se compartilharem a chave de link **KY** pré-configurada com o coordenador.

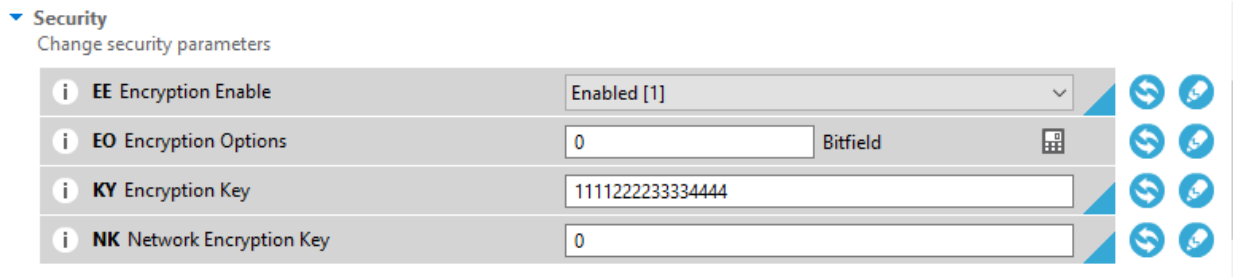

Figura 4.16: Configuração dos parâmetros de segurança do nó Coordenador, feito no *software XCTU*

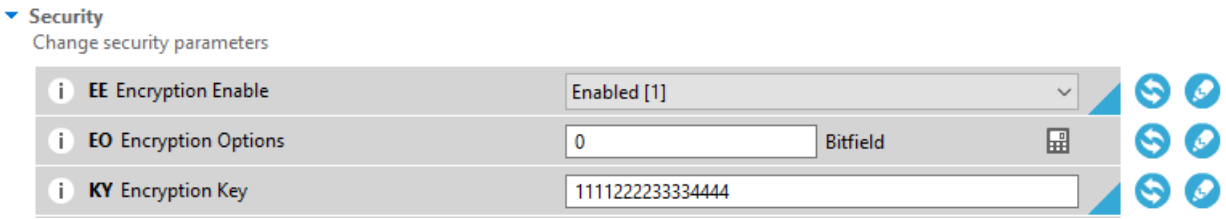

Figura 4.17: Configuração dos parâmetros de segurança do nó Roteador, feito no *software XCTU*

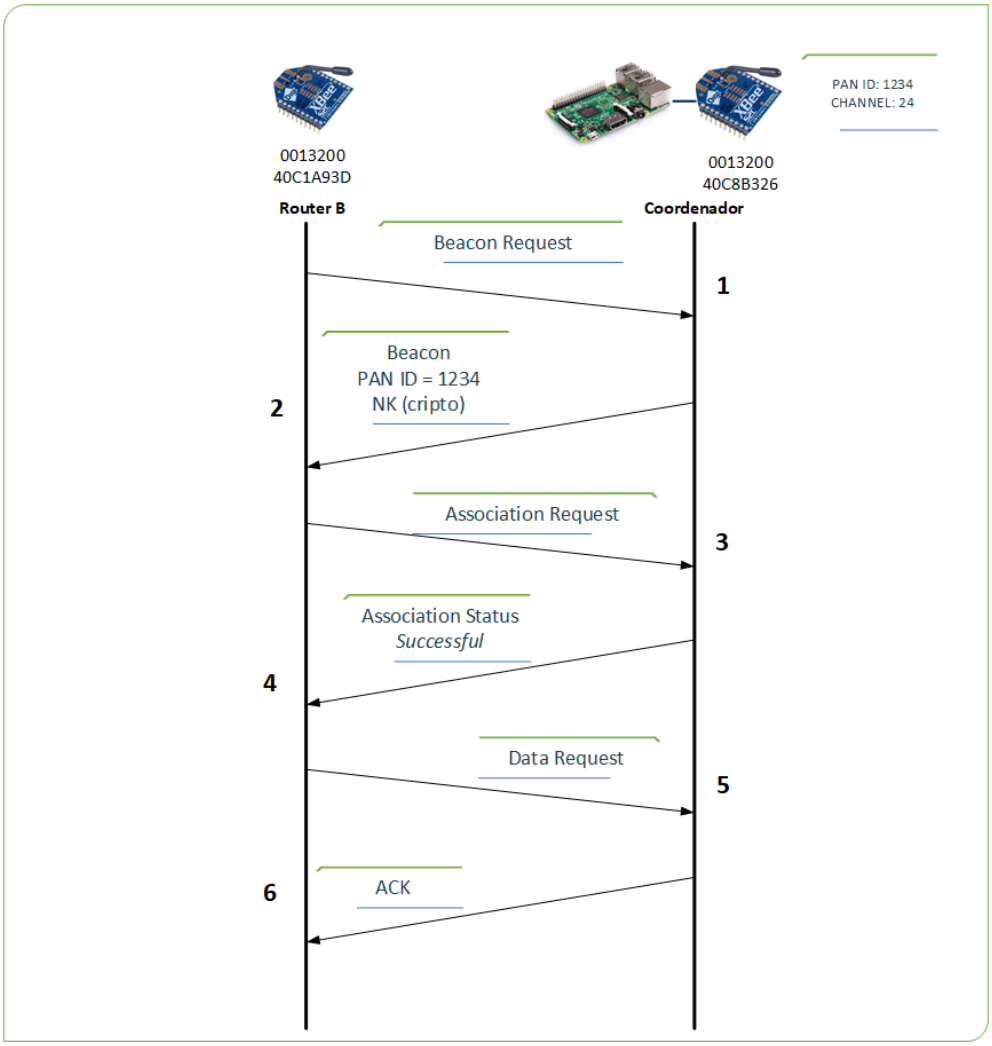

Figura 4.18: Diagrama de sequência para a comunicação ZigBee com criptografia.

Os passos indicados no diagrama de sequência, apresentado na Figura 4.18, são descritos a seguir:

1. O roteador envia um *Beacon Request* para obter as informações da rede que quer se conectar.
2. O coordenador envia um *Beacon* com as informações da rede e a chave da rede, que por sua vez é enviada criptografada com a chave de link, pré configurada igualmente no roteador e coordenador.
3. Em seguida, o roteador envia uma requisição para associar-se a esta rede.
4. Logo, o coordenador, caso aceite a conexão, responde com um *frame* de confirmação do estabelecimento da comunicação.
5. Com a conexão estabelecida, o próximo passo é enviar um *Data Request* com a mensagem que se quer enviar ao nó coordenador.
6. Ao receber cada *frame*, o coordenador responde com um **ACK** ao nó roteador.

A Figura 4.19 mostra o resultado de como é coletada a mensagem através do USB Dongle e mostrado pelo SmartRF Packet Sniffer. Vale ressaltar que não é possível visualizar nenhuma *string* nitidamente, diferente do que ocorre quando não se utiliza a função de criptografia dos *frames* na camada MAC.

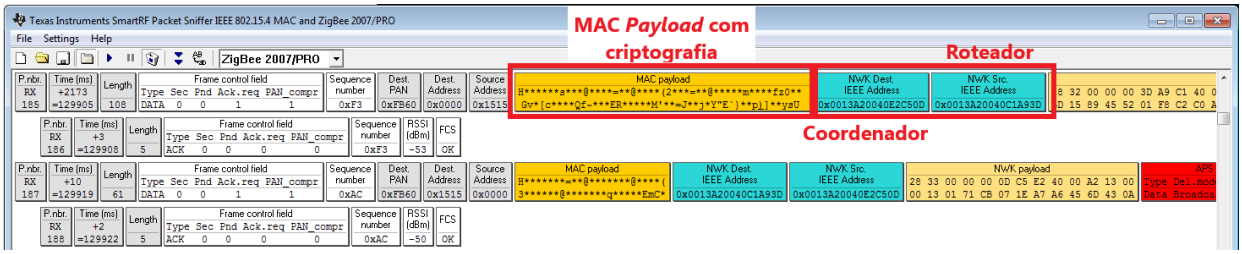

Figura 4.19: Captura de *frame* com criptografia. Visualização obtida com o *software* SmartRF-Packet Sniffer.

Outro ponto a se observar é que o tamanho de *frame* se tornou maior se comparado com o tamanho do *frame* em que não se implementou criptografia. Isso ocorre devido ao algoritmo de criptografia que foi utilizado, que é o *AES 128-bit*, conforme explicado no Capítulo 2 que trata da Fundamentação Teórica deste trabalho.

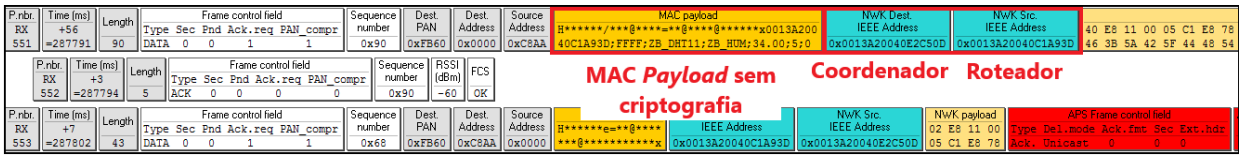

Figura 4.20: Captura de *frame* sem criptografia. Visualização obtida com o *software* SmartRF-Packet Sniffer.

Embora o *sniffer* seja incapaz de decifrar a mensagem criptografada, o nó coordenador recebe e é capaz de ler a *string* enviada via Zigbee, exatamente no padrão em que era esperado pela aplicação implementada no laboratório.

Esta informação pode ser observada, utilizado o modo de *Console* da ferramenta XCTU conectada ao nó coordenador. De acordo com a Figura 4.21, está indicado em **1** os dados enviados a partir do roteador, em **2**, o endereço MAC de origem e em **3**, é possível observar que o tamanho do pacote, 59 bytes, é menor do que aquele coletado pelo SmartRF Packet Sniffer, que era de 90 bytes, pois aqui tem-se o tamanho após a descriptografia do mesmo.

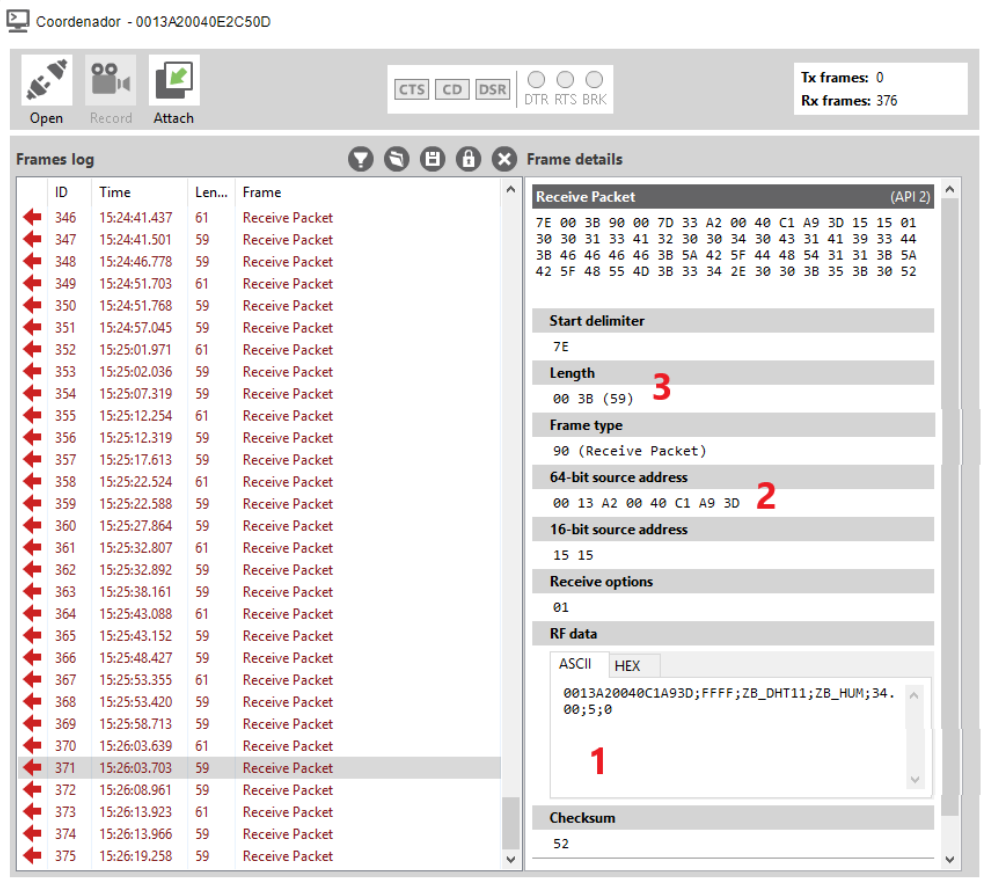

Figura 4.21: Detalhes do *frame* recebido pelo nó Coordenador. Visualização obtida com o *software* XCTU.

Com a implementação da criptografia nos módulos XBee, constatou-se que é possível deixar a comunicação da rede Zigbee mais segura aplicando as configurações apresentadas. Tendo em vista a importância da privacidade dos dados trafegados, é de crucial relevância que sejam empregados métodos que impeçam a conexão de um eventual intruso na rede, uma atidude mais eficaz do que a tomada de decisão após a conexão estabelecida, pois nesse caso, seria difícil saber quais informações o atacante já poderia ter coletado.

Destaca-se ainda que a utilização deste método não exclui de maneira alguma a implementação dos outros mecanismos de monitoramento, o sistema torna-se ainda mais eficiente com a união de ambos os procedimentos. Portanto, a proposta de monitoramento apresentada deve ser aplicada em conjunto com a criptografia de dados como estratégias de segurança da rede Zigbee.

Capítulo 5

Conclusão e Trabalhos Futuros

As redes IoT constituem atualmente uma das principais tecnologias emergentes deste século. Dentre os protocolos escolhidos para implementações deste tipo de rede, ZigBee é considerado um dos mais aplicados em soluções residenciais e empresariais, por esse motivo, é de fundamental importância que os métodos de segurança continuem a evoluir para garantir integridade, autenticidade e privacidade.

Neste contexto, o presente trabalho demonstrou eficiência no uso de inspeção da topologia da tecnologia Zigbee, presente no Laboratório UIoT, do curso de Engenharia de Redes de Comunicação, localizado na Universidade de Brasília, campus Darcy Ribeiro, investigando os dados e mensagens recebidas. Analisando a segurança e vulnerabilidades da rede, de forma a introduzir ataques modificando a topologia atual e comprometendo a integridade dos dados, recebidos pelo nó coordenador da rede.

Levando esses fatores em conta, foi feita uma proposta de monitoração para a rede Zigbee do laboratório, visando facilitar a detecção de dispositivos intrusos e proteção de vulnerabilidades. Contudo, a solução proposta também possui limitações no sentido de não avisar de maneira imediata a detecção de um novo nó não esperado.

Além disso, propôs-se ainda, a utilização do mecanismo de criptografia de dados para módulos Zigbee, de forma que a privacidade dos dados trafegados na rede seja protegida, considerando que mesmo que seja empregado algum dispositivo *sniffer* para coleta de pacotes, não seja possível decifrar o conteúdo das mensagens enviadas e também impossibilite a entrada de um novo dispositivo sem o conhecimento dos responsáveis pela rede Zigbee.

Como dificuldades encontradas na realização dos objetivos propostos, é importante destacar, que, por ser um assunto ainda de bastante estudo, as referências bibliográficas são restritas, o que dificulta e limita o acesso às informações.

Diante do exposto, como trabalhos futuros, sugere-se o desenvolvimento de um sistema de segurança mais robusto que possibilite a detecção instantânea de intrusos na rede, além da notificação imediata ao responsável.

Bibliografia

ARDUINO. **Arduino - About us**. [S.l.: s.n.], 2018. Disponível em: <https://www.arduino.cc/en/Main/AboutUs>. Acesso em: 23 out. 2018.

AZZI, Charbel. **Vulnerability Analysis and Security Framework for Zigbee Communication in IOT**. University of Nevada, Las Vegas: [s.n.], 2016.

BRONZATTI, Luiz Fernando Casarin. **Análise sobre a tecnologia de rede sem fio Zigbee / IEEE 802.15.4**. São Carlos, SP, Brasil: [s.n.], 2013.

CAMHI, Jonathan. **How different networking standards are competing to connect the Internet of Things**. [S.l.: s.n.], 2016. Disponível em: <http://www.businessinsider.com/iot-networks-report-2016-4-23>. Acesso em: 14 jun. 2018.

DIGI. **XBee Zigbee Mesh Kit - User Guide**. [S.l.], 2018.

EGLI, Peter. Susceptibility of wireless devices to denial of service attacks. **Technical white paper, Netmodule AG**, 2006.

EQUIPE MOUNTAINBAJA. **Configurando o Xbee pelo software XCTU**. [S.l.: s.n.]. Disponível em: <https://portal.vidadesilicio.com.br/configurando-o-xbee-pelo-software-xctu/>. Acesso em: 13 out. 2018.

EVANS, Dave. A Internet das Coisas: Como a próxima evolução da Internet está mudando tudo. **White Paper**, 2011.

FALUDI, Robert. **Building wireless sensor network: with ZigBee, XBee, Arduino, and Processing**. Edição: Inc. O'Reilly Media. Sebastopol, Califórnia, EUA: [s.n.], 2010.

FILHO, José Gonçalves Pereira. **Redes de Sensores Sem Fio - Sub Camada MAC, Formato do Frame**. [S.l.], 2015.

FRARE, Bruno Pinaffi; ARAKI, Flávio Nobuteru Tachikawa; XAVIER, Marcelo Fantini. **Aplicação do ZigBee na Segurança**. Itajubá - Minas Gerais, Brasil: 15 de junho de 2009, 2009.

GARRET, Filipe. **Como funciona o Raspberry Pi? Entenda a tecnologia e sua aplicabilidade**. [S.l.: s.n.]. Disponível em: <https://www.techtudo.com.br/noticias/noticia/2014/11/como-funciona-o-raspberry-pi-entenda-tecnologia-e-sua-aplicabilidade.html>. Acesso em: 13 out. 2018.

GARTNER. **Gartner Hype Cycle**. [S.l.: s.n.], 2018. Disponível em: <https://www.gartner.com/en/research/methodologies/gartner-hype-cycle>. Acesso em: 26 out. 2018.

GUTIERREZ, J.A. et al. Applying Wireless Sensor Networks in Industrial Plant Energy Evaluation and Planning Systems. IEEE Conference Record of Annual Pulp e Paper Industry Technical Conference, 2006.

IEEE. **IEEE 802.15.4-2006 Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)**. New York, NY, EUA, set. 2006.

IEEE 802.15. **IEEE 802.15 Working Group for Wireless Personal Area Networks**. [S.l.: s.n.], 2018. Disponível em: <http://www.ieee802.org/15/about.html>. Acesso em: 20 out. 2018.

JUNIPER NETWORKS. **Understanding Rogue Access Points**. [S.l.: s.n.], 2016. Disponível em: <https://www.juniper.net/documentation/en_US/junos-space-apps/network-director3.1/topics/concept/wireless-rogue-ap.html>. Acesso em: 4 nov. 2018.

KINNEY, Patrick. **ZigBee Technology: Wireless Control that Simply Works**. [S.l.]: Communications Design Conference, 2003.

LEE, Kyunghwa et al. An enhanced Trust Center based authentication in ZigBee networks. **in Advances in Information Security and Assurance**, p. 471–484, 2009.

LIBELIUM. **Waspmote ZigBee**. [S.l.], 2018.

MADAKAM, Somayya; RAMASWAMY, R.; TRIPATHI, Siddharth. Internet of Things (IoT): A Literature Review. **Journal of Computer and Communications**, v. 3, p. 164–173, 2015. Disponível em: <http://dx.doi.org/10.4236/jcc.2015.35021>. Acesso em: 26 out. 2018.

MAIA, Caio Rodrigo Nascimento. **Implementação da Tecnologia de comunicação ZigBee em plataforma de gerenciamento de serviços IoT**. Brasília, DF, Brasil: [s.n.], 2017.

MAXSTREAM. **XBee™/XBee-PRO™ OEM RF Modules**. 1. ed. Lindon, UT, EUA, maio 2007.

MITSHELL. **802.15.4 monitor**. [S.l.: s.n.], 2016. Disponível em: <https://github.com/mitshell/CC2531>. Acesso em: 25 out. 2018.

MURALEEDHARAN, Rajani; OSADCIW, Lisa Ann. **Jamming attack detection and countermeasures in Wireless Sensor Network using ant system**. [S.l.]: in Proceedings of the SPIE, 2006. v. 6248.

NIST. AES: Advanced Encryption Standard. [S.l.: s.n.]. Disponível em: <http://csrc.nist.gov/CryptoToolkit/aes/>. Acesso em: 7 jun. 2018.

PANETTA, Kasey. **5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018**. [S.l.: s.n.], 2018. Disponível em: <https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/>. Acesso em: 26 out. 2018.

PINHEIRO, José Maurício Santos. **As Redes com Zigbee**. [S.l.: s.n.], 2004. Disponível em: <https://www.projetoderedes.com.br/artigos/artigo_zigbee.php>. Acesso em: 20 out. 2018.

PIRES, Felipe Marques; MIANI, Rodrigo Sanches; SOUZA MENDES, Leonardo de. An Architecture for Environmental Monitoringand Control in Municipal Scale. **International Conference on Wireless Information Networks and Systems**, p. 27–31, 2009.

RADMAND, Pedram et al. ZigBee/ZigBee PRO security assessment based on compromised cryptographic keys. **International Conference on P2P, Parallel, Grid, Cloud and Internet Computing**, 2010.

RASPBERRY PI FOUNDATION. **Raspberry Pi - About us**. [S.l.: s.n.]. Disponível em: <`https://www.raspberrypi.org/about/`>. Acesso em: 22 out. 2018.

RF WIRELESS WORLD. **Difference between LoRa and Zigbee**. [S.l.: s.n.], 2012. Disponível em: <`http://www.rfwireless-world.com/Terminology/LoRa-vs-Zigbee.html`>. Acesso em: 16 nov. 2018.

ROWEBOTS. **Wireless Protocols**. [S.l.: s.n.], 2015. Disponível em: <`https://rowebots.com/en/products/unison-rtos-article/wireless-protocols`>. Acesso em: 20 out. 2018.

SANTOS, Bruno P. et al. **Internet das Coisas: da Teoria à Prática**. Edição: Universidade Federal de Minas Gerais. Belo Horizonte, Minas Gerais, Brasil: [s.n.], 2016.

SASTRY, Naveen; WAGNER, David. **Security considerations for IEEE 802.15.4 networks**. Edição: ACM. New York, NY, USA: in WiSe 04: Proceedings of the 3rd ACM workshop on Wireless security, 2004. p. 32–42.

SHIEH, Ji-Tsong; KO, Li Chun. **Implementation of a broadcast authentication mechanism in ZigBee**. [S.l.]: in The 2nd Workshop on Wireless, Ad Hoc, e Sensor Networks (WASN), 2006. v. 2006.

SILVA, Bruna Roberta Seewald da. **Sistema de automação residencial de baixo custo para redes sem fio**. Porto Alegre,RS, Brasil: [s.n.], 2014.

SOKULLU, Radosveta et al. **An investigation on IEEE 802.15.4 MAC layer attacks**. [S.l.]: in Proceedings of The 10th International Symposium on Wireless Personal Multimedia Communications (WPMC), 2007.

SONG, Tian-Wen; YANG, Chu-Sing. **A connectivity improving mechanism for ZigBee Wireless Sensor Networks**. Shanghai, China: Embedded e Ubiquitous Computing, IEEE/IFIP International Conference on, 2008. v. 2, p. 495–500.

SOUZA, Anderson R. de et al. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC. **Revista Brasileira de Ensino de Física**, v. 33, n. 1, p. 1702, 2011.

TEIXEIRA, Catharina Daher; LACERDA CLARIM, Mariana de. **Estudo das Vulnerabilidades de Tecnologias Sem Fio Utilizadas em Ambientes IOT.** Brasília, DF, Brasil: [s.n.], 2017.

TEXAS INSTRUMENTS. **CC2531 USB Evaluation Module Kit.** [S.l.: s.n.]. Disponível em: <`http://www.ti.com/tool/CC2531EMK`>. Acesso em: 6 out. 2018.

_____. **SmartRF™ Packet Sniffer User's Manual**. Dallas, Texas, EUA, maio 2014.

_____. **System-on-Chip Solution for IEEE 802.15.4 and ZigBee Applications.** [S.l.: s.n.]. Disponível em: <`http://www.ti.com/product/CC2531`>. Acesso em: 6 out. 2018.

VASQUES, Bruna Luisa Ramos Prado et al. **As Redes com Zigbee**. [S.l.: s.n.], 2010. Disponível em: <https://www.gta.ufrj.br/grad/10_1/zigbee/aplicacoes.html>. Acesso em: 20 out. 2018.

VIDGREN, Niko et al. Security Threats in ZigBee-Enabled Systems: Vulnerability Evaluation, Practical Experiments, Countermeasures, and Lessons Learned. **46th Hawaii International Conference on System Sciences**, 2013.

WRIGHT, Joshua; CACHE, Jonnhy. **Hacking Exposed Wireless: Wireless Security Secrets & Solutions**. 3. ed. New York, NY, EUA: McGrawHill Education, 2015. p. 540–600.

XIAO, Yang et al. **MAC security and security overhead analysis in the IEEE 802.15.4 Wireless Sensor Networks**. [S.l.]: EURASIP Journal on Wireless Communications e Networking, 2006. v. 2006, p. 1–12.

YAHIA, Hazha Saeed. **Performance Analysis of Real-Timewireless Body Sensor Networks Using 802.15.4 and Zigbee Standards Under Maximum Payload Conditions**. Curdistão, Iraque: [s.n.], 2016.

ZIGBEE ALLIANCE. **Zigbee is the only complete loT solution, from the mesh network to the universal language that allows smart objects to work together.** [S.l.: s.n.], 2018. Disponível em: <http://www.zigbee.org/zigbee-for-developers/zigbee-3-0/>. Acesso em: 15 jun. 2018.

_____. **ZigBee Specification**. San Ramon, USA, set. 2012.

ZUCATO, Fábio Labegalini. **Rede ZigBee Gerenciada por Sistema de Monitoramento Remoto Utilizando TCP/IP e GPRS**. São Carlos, SP, Brasil: [s.n.], 2009.