



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
ENGENHARIA ELETRÔNICA E DE TELECOMUNICAÇÕES

Raquel Maria Oliveira Ribeiro

**Segurança em IoT: simulação de ataque em uma rede RPL
utilizando *Contiki*.**

PATOS DE MINAS

2018

RAQUEL MARIA OLIVEIRA RIBEIRO

**SEGURANÇA EM IoT: SIMULAÇÃO DE ATAQUE EM UMA REDE RPL
UTILIZANDO *CONTIKI*.**

Trabalho apresentado à Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia, como pré-requisito da disciplina de Trabalho de Conclusão de Curso II do curso de graduação em Engenharia Eletrônica e de Telecomunicações

Orientador: Prof. Julio César Coelho

PATOS DE MINAS

2018

**SEGURANÇA EM IoT: SIMULAÇÃO DE ATAQUE EM UMA REDE RPL
UTILIZANDO *CONTIKI*.**

Trabalho apresentado à Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia, como pré-requisito da disciplina de Trabalho de Conclusão de Curso II do curso de graduação em Engenharia Eletrônica e de Telecomunicações

Aprovado em: 04/07/2018

Banca examinadora:

Prof. Me. Júlio César Coelho (Orientador,UFU)

Profa. Dra. Karine Barbosa Carbonaro (Examinadora,UFU)

Prof. Me. Alexander Bento Melo (Examinador,UFU)

AGRADECIMENTOS

À minha família pela compreensão, apoio, carinho e por ser a base onde me sustentei.

Ao professor e orientador Júlio César Coelho pela oportunidade de conhecê-lo pessoalmente e profissionalmente, além de toda ajuda durante o TCC.

Ao professor Alexander Bento Melo, por toda paciência com as falhas e a boa vontade em transmitir experiência e conhecimento.

À Universidade Federal de Uberlândia, *Campus* Patos de Minas, a qual levarei comigo com muito orgulho e todos os seus funcionários que diretamente ou indiretamente contribuíram para a formação.

À todos os professores que fizeram parte da minha trajetória na faculdade até aqui, pela transmissão do conhecimento e por conselhos quando tudo parecia difícil.

RESUMO

A Internet das Coisas internet das coisas é uma tecnologia que vem crescendo muito e ganhando destaque no cenário mundial. A Internet das Coisas proporciona aos objetos do dia a dia, com capacidade computacional e de comunicação, se conectarem à internet. Essa conexão viabilizará controlar remotamente os objetos, e acessá-los como provedores de serviços, e se tornarão objetos inteligentes ou smart objects. Os objetos inteligentes possuem capacidade de comunicação e processamento aliados a sensores.

Atualmente não só computadores convencionais estão conectados à internet, como também uma grande heterogeneidade de equipamentos, tais como TVs, laptops, geladeira, fogão, eletrodomésticos, automóveis, smartphones, entre outros. Nesse novo cenário, a pluralidade é crescente e previsões indicam que mais de 50 bilhões de dispositivos estarão conectados até 2020. Com o uso dos objetos inteligentes será possível detectar seu contexto, controlá-lo, viabilizar troca de informações uns com os outros, acessar serviços da internet e interagir com as pessoas. Em paralelo, uma gama de novas possibilidades de aplicações surge, como, por exemplo, cidades inteligentes (smart cities); saúde (smart healthcare); casas inteligentes (smart home) e desafios emergem (regulamentações, segurança, padronizações). Essas novas habilidades dos objetos inteligentes gerarão um grande número de oportunidades de pesquisas e projetos no âmbito acadêmico e empresarial. Além do desenvolvimento de tecnologias há de se efetuar a pesquisa e o tratamento dos aspectos de segurança aplicáveis em infraestrutura para Internet das coisas.

Para que este novo sistema seja possível é necessário que haja uma infraestrutura tecnológica de comunicação complexa, afim de manter todo o processo com uma segurança robusta e confiável, além de permitir rapidez ao usuário. Este trabalho foca na simulação de ataques contra redes de IoT, utilizando o software *Contiki* com o simulador *Cooja*, afim de verificar a vulnerabilidade e o potencial da rede IoT.

Palavras-chave: IoT; Segurança de rede; Hello-Flood; RPL; rede de sensores; protocolos.

ABSTRACT

The Internet of Things Internet of things is a technology that has been growing a lot and gaining prominence in the world scenario. The Internet of Things provides everyday objects with computing and communication capabilities to connect to the internet. This connection will enable remote control of objects, access them as service providers, and become intelligent objects or smart objects. Intelligent objects have communication and processing capabilities together with sensors.

Currently, not only are conventional computers connected to the Internet, but also a great variety of equipment such as TVs, laptops, refrigerators, stoves, appliances, automobiles, smartphones, among others. In this new scenario, plurality is growing and predictions indicate that more than 50 billion devices will be connected by 2020. Using smart objects will be able to detect their context, control it, make it possible to exchange information with each other, access services the internet and interact with people. In parallel, a range of new application possibilities arise, such as, for example, smart cities; health (smart healthcare); smart homes and challenges emerge (regulations, security, standardization). These new skills of intelligent objects will generate a large number of research and project opportunities in the academic and business spheres. In addition to the development of technologies, research and treatment of the applicable security aspects in Internet infrastructure of things will be carried out. In order for this new system to be possible, it is necessary to have a complex communication technology infrastructure, in order to maintain the entire process with a robust and reliable security, and to allow the user quickness. This work focuses on the simulation of attacks against IoT networks, using Contiki software with the Cooja simulator, in order to verify the vulnerability and potential of the IoT network.

Keywords:IoT; Network security; Blackhole; RPL; network of sensors; protocols..

LISTA DE FIGURAS

Figura 1- Relação entre conexões.....	12
Figura 2- Três principais paradigmas da Internet das Coisas.	19
Figura 3 - Rotinas de Middleware	21
Figura 4- Esquema de computação em nuvem.	23
Figura 5- Camadas de IoT.	26
Figura 6 - Objetivos de confiabilidade.	36
Figura 7 - Comparação entre simuladores.	42
Figura 8 - DODAG RPL.....	44
Figura 9 - Direção de comunicação.	44
Figura 10 - Topologia com 10 nós sensores sem nó malicioso.	48
Figura 11 - Topologia com 15 nós sensores sem nó malicioso.	48
Figura 12 - Topologia com 20 nós sensores sem nó malicioso.	49
Figura 13 - Topologia com 10 nós sensores com o nó malicioso.	49
Figura 14 - Topologia com 15 nós sensores com nó malicioso.	50
Figura 15 - Topologia com 20 nós sensores com nó malicioso.	50
Figura 16 - Consumo de energia por nó, topologia com 10 nós sensores sem nó malicioso.	52
Figura 17 - Consumo de energia por nó, topologia com 15 nós sensores sem nó malicioso.	52
Figura 18 - Consumo de energia por nó, topologia com 20 nós sensores sem nó malicioso.	53
Figura 19 - Consumo de energia por nó, topologia de 10 nós sensores com nó malicioso.	53
Figura 20 - Consumo de energia por nó, topologia de 15 nós sensores com nó malicioso.	54
Figura 21 - Consumo de energia por nó, topologia de 20 nós sensores com nó malicioso.	54
Figura 22 - Topologia com 10 nós sensores sem nó malicioso.	55
Figura 23 - Topologia com 15 nós sensores sem nó malicioso.	56
Figura 24 - Topologia com 20 nós sensores sem nó malicioso.	56
Figura 25 - Topologia de 10 nós sensores com nó malicioso.....	57
Figura 26- Topologia de 15 nós sensores com nó malicioso.....	57
Figura 27 - Topologia de 20 nós sensores com nó malicioso.....	58
Figura 28 - Consumo de energia por nó, topologia de 10 nós sensores sem nó malicioso	59
Figura 29 - Consumo de energia por nó, topologia de 15 nós sensores sem nó malicioso	59
Figura 30 - Consumo de energia por nó, topologia de 20 nós sensores sem nó malicioso	60

Figura 31 - Consumo de energia por nó, topologia de 10 nós sensores com nó malicioso.....	60
Figura 32 - Consumo de energia por nó, topologia de 15 nós sensores com nó malicioso.....	61
Figura 33 - Consumo de energia por nó, topologia de 20 nós sensores com nó malicioso.....	61

LISTA DE QUADROS

Quadro 1- Padrão proposto para IoT.	25
Quadro 2 - Diferenças entre MQTT e CoAP.....	28
Quadro 3 - Comparação entre protocolos de aplicação para IoT.	30
Quadro 4 - Protocolos usados por uma rede IP tradicional e IoT.....	32
Quadro 5 - Principais requerimentos de segurança e seus sub componentes.	37
Quadro 6 - Classificação de ataques IoT.....	39
Quadro 7 - Parâmetros de simulação do cenário.	43

LISTA DE ABREVIATURAS E SIGLAS

6LoWPAN	- <i>IPv6 over Low power wireless personal area networks</i>
AMI	- <i>Advanced metering infrastructure</i>
AMQP	- <i>Advanced message queuing protocol</i>
AMR	- <i>Automatic meter reading</i>
CoAP	- <i>Constrained application protocol</i>
DDS	- <i>Data distribution service</i>
DNS	- <i>Domain name system</i>
DNS-SD	- <i>DNS service discovery</i>
EPC	- <i>Electronic Product Code</i>
EXI	- <i>Efficient XML Interchange</i>
HAL	- <i>Hardware abstraction layer</i>
HAN	- <i>Home area network</i>
HTTP	- <i>Hypertext Transfer Protocol</i>
IDE	- <i>Integrated development environment</i>
IoT	- <i>Internet of things</i>
IPv4	- <i>Internet protocol version 4</i>
IPv6	- <i>Internet protocol version 6</i>
LTE-A	- <i>Long term Evolution advanced</i>
M2M	- <i>Machine to machine</i>
mDNS	- <i>multicast DNS</i>
MQTT	- <i>Message queue telemetry transport</i>
MQTT-SN	- <i>MQTT sensor network</i>
NAN	- <i>Neighborhood area network</i>
OS	- <i>Operating system</i>
OWL	- <i>Web ontology language</i>
REST	- <i>Representational state transfer</i>
RF	- <i>Radio frequency</i>
RDF	- <i>Resource description framework</i>
RPL	- <i>Routing protocol for low power and lossy networks</i>
RSSI	- <i>Relative received signal strength</i>

- SOS - *Simple operating system*
- SRAM - *Static random-access memory*
- TCP - *Transmission control protocol*
- UDP - *User data gramprotocol*
- USB - *Universal serial bus*
- WAN - *Wide area network*
- XML - *Extensible markup language*
- XMPP - *Extensible messaging and presence protocol*

SUMÁRIO

RESUMO.....	iii
ABSTRACT	iv
LISTA DE FIGURAS.....	iv
LISTA DE QUADROS.....	vii
LISTA DE ABREVIATURAS E SIGLAS	viii
CAPÍTULO 1	12
INTRODUÇÃO	12
1.1 – APRESENTAÇÃO	12
1.2 – OBJETIVO	14
1.3 – JUSTIFICATIVA	15
1.4 – ORGANIZAÇÃO DO TEXTO	17
CAPÍTULO 2	18
TECNOLOGIAS IOT	18
2.1 – INTERNET DAS COISAS	18
2.2 – PROTOCOLOS PARA REDES DE SENSORES E CAMADAS IoT.....	24
2.2.1 – Protocolos de Aplicação.....	27
2.2.2 – Protocolos de rede e tecnologias de comunicação	30
2.3 – RESUMO	33
CAPÍTULO 3	34
SEGURANÇA EM IOT	34
3.1 - PROBLEMAS DE SEGURANÇA.....	34
3.2 - SEGURANÇA E PRIVACIDADE.....	36
3.3 - ATAQUES E CONTRAMEDIDAS.	37
3.4 - RESUMO.....	41
CAPÍTULO 4	42
MATERIAIS E MÉTODOS.....	42
4.1 – AMBIENTE E CENÁRIO DE SIMULAÇÃO.....	42
4.2 - TESTES DA REDE	44
CAPÍTULO 5	46

RESULTADOS E DISCUSSÕES.....	46
5.1 –FUNCIONAMENTO DA REDE	46
5.2 – RESULTADOS DOS TESTES	47
CAPÍTULO 6.....	62
CONCLUSÃO E TRABALHOS FUTUROS	62
REFERÊNCIAS BIBLIOGRÁFICAS	63

CAPÍTULO 1

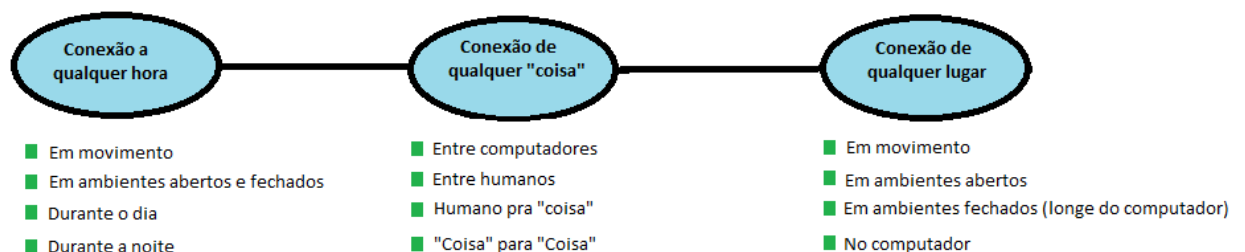
INTRODUÇÃO

1.1 – Apresentação

O termo Internet das coisas (IoT) foi usado pela primeira vez por Kevin Ashton em uma apresentação em 1998 [1]. A Internet das coisas é um paradigma no qual se estabelece conexão (comunicação) entre todos os objetos ao redor [2]. A ideia de uma rede mundial de objetos conectados que trocam informação entre si é bastante ampla e faz com que muitas tecnologias e aplicações diferentes atendam pelo nome de Internet das Coisas [3]. A IoT prevê um futuro no qual entidades físicas e digitais podem ser ligadas, por meio de informação e comunicação adequadas [4].

Com a tecnologia avançando cada vez mais, temos uma maior comunicação, onde pessoas, processos e equipamentos estão se integralizando e otimizando em diversas vertentes. Na IoT, "coisas" que estão presentes no dia a dia das pessoas, tais como cafeteiras, relógios e despertadores, são conectados à Internet de tal forma que as informações podem ser enviadas através da rede para qualquer lugar a qualquer momento [5]. A Figura 1 ilustra as novas dimensões que irão estar presentes no mundo das tecnologias da comunicação e informação da internet no futuro. Observa-se “o quê” pode ser conectado a internet, “quando” pode ser conectado e “onde” pode se conectar [2].

Figura 1- Relação entre conexões.



Fonte: Adaptado de [6].

A tendência de modos diversificados na comunicação da rede IoT motiva os profissionais e pesquisadores a desenvolverem múltiplas técnicas de comunicação, baseando-se em diferentes meios de propagação de sinais, sendo em meios guiados ou não.

Como o IoT é um paradigma que visa facilitar a troca de informações, ela é utilizada para comportar serviços globais e redes de fornecimento de bens em variadas áreas, tais como indústria, logística, educação e saúde [7]. Vê-se também comunicações via satélite sendo implementadas, são utilizadas por exemplo em carros, onde os mesmos trocam informações, como sua localidade, velocidade e caminho que será seguido evitando assim acidentes entre eles, facilitando tanto o trânsito em rodovias como em vias urbanas. Estes métodos de negócio oferecem redução de custos e de riscos, aprimorando os processos industriais e empresariais, criando oportunidades no meio corporativo, de forma a mudar o rumo e o ritmo de negócios [8].

Jara et al. (2014) fala que a rede IoT é composta por objetos inteligentes, ou seja, dispositivos físicos pequenos e altamente restritos em termos de capacidade de memória, capacidade de computação, autonomia energética e capacidade de comunicação. Ele fala também que o objetivo da Internet das coisas é a unificação e integralização dos sistemas de comunicação que nos cerca no dia a dia, dessa forma os sistemas poderão eventualmente obter um controle total e acesso a outros sistemas, a fim de permitir e proporcionar a comunicação e a computação ubíqua.

A internet das coisas está emergindo como uma das principais tendências que moldam o desenvolvimento das tecnologias de informação e comunicação [10]. Entretanto, juntamente com esta revolução da tecnologia há também uma grande preocupação com as possíveis vulnerabilidades que os dispositivos IoT podem sofrer. Temos problemas como a privacidade do usuário, controle de acesso, configuração do sistema, armazenamento de informações e gerenciamento, estes são os principais desafios a serem enfrentados nesse ambiente, então para que o uso do IoT possa ser feito de forma extensiva, deverá ser feito um grande estudo para a solução da segurança.

A segurança é um importante aspecto para implantação da Internet das coisas [11]. Neste contexto, requisitos de segurança e arquiteturas devem ser adequadamente formuladas e implementadas a fim de fazer cumprir as políticas de segurança durante o seu ciclo de vida [12]. A Internet das Coisas exige soluções de segurança sob medida [13]. Contra medidas tradicionais

de segurança não podem ser aplicadas diretamente ao IoT devido aos diferentes padrões envolvidos [14]. Hoje, há uma série de propostas capazes de atender às demandas da Internet das coisas no contexto de ataques de estranhos [13].

Com o aumento dos dispositivos inteligentes e a mobilidade de alguns destes, a Internet das coisas fica exposta a várias vulnerabilidades na comunicação pois apresenta uma infra-estrutura variável e a maioria dos dispositivos possuem recursos limitados, como baixa energia, capacidade de processamento e armazenamento restrita, conexão através de enlaces que possuem perdas e outros tipos de características [10].

Para a simulação de uma rede de sensores sem fio em uma rede IoT, pode ser usado o simulador Cooja que faz parte do sistema operacional Contiki. O Contiki é um sistema operacional de código aberto que foi desenvolvido para aplicações desenvolvidas sob o paradigma IoT, onde o maior enfoque é em sistemas que possuem baixo consumo de energia e pouca memória, por isso, ele consome apenas aproximadamente 10 Kbytes de memória RAM e 30 Kbytes de memória ROM [15]. Juntamente com o Contiki, há uma ferramenta de simulação de aplicações IoT denominada Cooja, a qual possui funcionalidades para construção das topologias, escolha do modelo do dispositivo a ser simulado, protocolos, entre outras coisas, permitindo analisar o comportamento da rede, consumo de energia e alcance do sinal do nó [16]. Tem-se que o sistema operacional Contiki possibilita suporte para aplicações em rede, utilizando protocolos UDP, TCP e HTTP, onde também suporta padrões IPv6 e IPv4 e as novas normas de baixa potência como 6LowPAN, RPL e Coap [17].

1.2 – Objetivo

O objetivo deste trabalho é criar uma simulação de redes sem fio, utilizando o simulador *Cooja* juntamente com o *Contiki*. Onde possa constituir uma plataforma para a implementação de uma aplicação IoT e posteriormente colocar nós atacantes, afim de perceber como a rede IoT pode ser vulnerável a certos ataques.

Implementar o ataque Hello-Flood e Versioning Attack (Ataque de versão), em uma rede RPL afim de caracterizar a energia consumida por nó após o ataque ser iniciado.

1.3 – Justificativa

Ainda que o conceito de IoT esteja em alta, seu potencial e maturidade ainda está longe de ser alcançada, temos várias notícias de novos produtos que estão sendo idealizados para produção e também o melhoramento de novas tecnologias que utilizam-se da rede IoT. Isso deve ser pelo fato de ser uma tecnologia em fase de desenvolvimento e com grande complexidade, e que possui plataformas limitadas de hardware e software que são economicamente viáveis.

A internet das Coisas (IoT – *Internet of Things*) é considerada uma infra-estrutura de rede dinâmica e global que possui capacidade onde uma auto configuração onde “*coisas*” físicas e virtuais têm atributos físicos, personalidades virtuais e identidades. Tais “*coisas*” usam interfaces inteligentes e são comumente integradas à internet [10]. As “*coisas*” ou objetos acabam por se tornar participantes ativos em processos de negócio social e na parte de informações, onde adquirem a capacidade de interagir e comunicar entre si, trocar informações coletadas do ambiente, reagir autonomamente aos eventos do mundo físico real, assim como influenciar essa situação com ou sem intervenção direta do ser humano[13].

A internet das coisas (IoT) começou a ser plausível a partir da evolução da automação, da microeletrônica, das técnicas de processamento digital de sinais (PDS), das telecomunicações e da inteligência artificial. Um sistema IoT consiste em objetos com capacidade de capturar os sinais de forma adequada (microeletrônica), processar esses sinais (PDS), interagir com os outros objetos, trocando informações (Telecomunicações), usando a inteligência artificial para tomar decisões apropriadas e a automação faz com que a decisão se torne uma ação. Por exemplo, usando a concepção de IoT é possível que uma TV perceba que o filme que está sendo assistido é um romance e envie informação para o sistema de iluminação diminuir a luminosidade do ambiente ou mudar a luminosidade de uma lâmpada de 6500K para uma luminosidade de uma lâmpada de 2700K, envie informação para que o ar-condicionado se ajuste ao momento, que a persiana receba informação para se fechar, que o porta-retratos digital comece a mostrar fotos do casal e que os celulares só atendam a certos números preestabelecidos.

O IoT é implementado utilizando uma variedade de tecnologias existentes, mais especificamente, utilizando as três seguintes [18]:

- Radio Frequency Identification (RFID);

- Wireless Sensor Networks (WSN);
- Cloud Computing.

A Radio Frequency Identification (RFID) é uma tecnologia que permite identificar todos os tipos de objetos sem contato físico, permite também que os dados sejam adquiridos de maneira rápida e automática. A tecnologia RFID permite a leitura e gravação de dados em etiquetas eletrônicas, independentemente da sua localização, o tipo de ambiente, o contato ou visibilidade [19]. Esta tecnologia tem um princípio parecido com o de IoT, o uso de identificação de vários objetos sem haver um contato físico entre eles, os dados são enviados para um computador, não havendo necessidade de fios ligados entre eles.

Apesar do termo Internet das Coisas ter surgido a um bom tempo, o primeiro eletrodoméstico 'inteligente' que usou desta tecnologia aparece inicialmente em junho de 2000, a LG apresentou sua geladeira inteligente durante um evento na Coreia do Sul. O produto deveria fazer par com outros dispositivos, todos conectados à Internet e gerenciáveis através de um sistema da própria LG [3]. Prevê-se que até 2020 a Internet das coisas irá se expandir com mais de 50 milhões de dispositivos inteligentes (excluindo PCs, tablets e smartphones), o que é um número impressionante [20]. Contudo, a existência de uma rede de computadores e objetos interconectados tão extensa, vai apresentar desafios na área de segurança, privacidade e confiança, colocando os usuários e os dispositivos em um alto risco [18].

Então pode-se ver que haverá um grande aumento de dispositivos inteligentes e vários serão com mobilidade, já que está crescendo por exemplo a tecnologia 5G, com isso a internet das coisas ficará exposta a vários ataques e vulnerabilidades em sua comunicação, tendo em sua composição dispositivos com recursos limitados, como baixa energia, e também uma conexão que possui perdas, possibilitando que nós atacantes possam enviar várias informações comprometendo a eficiência da rede, como também que possa haver ataques que simulam as informações passada entre um nó e outro. Com as novas ideias e aplicações que o IoT permite, tem-se que levar em consideração os problemas apresentados, dando enfoque na questão de segurança, esta precisa ser levada bastante a sério e necessita de muitos estudos, afim de beneficiar a todos, tanto o usuário final quanto os responsáveis pela disseminação deste novo avanço, criando uma rede segura e com grande robustez.

1.4 – Organização do texto

Neste capítulo foi feita considerações a cerca dos tópicos que serão apresentados ao longe deste trabalho, fala sobre os principais temas que devem ser abordados no restante do texto, contendo seu maior detalhamento e teoria. No capítulo 2 será abordado de forma sucinta e teórica os conceitos necessários para a compreensão do básico de IoT, com enfoque em tecnologias de baixo consumo de energia como o RPL, e os possíveis ataques que a internet das coisas pode sofrer. Alguns destes conceitos são usados de forma mais prática no Capítulo 3 onde são descritos os materiais e métodos empregados para se construir a plataforma de comunicação RPL para IoT aqui abordada. No capítulo 4 há um maior detalhamento sobre a questão de segurança e privacidade que envolve a internet das coisas. Os resultados de testes feitos com o simulador Cooja e materiais aqui empregados são mostrados no Capítulo 5. Neste mesmo capítulo são feitos alguns comentários sobre os resultados dos experimentos conduzidos com esta tecnologia. Por fim, o Capítulo 6 conclui este trabalho.

CAPÍTULO 2

TECNOLOGIAS IOT

2.1 – Internet das Coisas

Tem-se que as aplicações da IoT compreendem diversas áreas, entre elas pode-se citar as áreas de energia, saúde, transporte, monitoramento, automação, agricultura, industrial e residencial [21]. Embora controverso, esse tipo de comunicação também é algumas vezes referenciado como M2M (*machine to machine*) [22]. Segundo Reis (2013) M2M é um termo utilizado para descrever tecnologias que permite aos computadores, sensores inteligentes, atuadores e dispositivos móveis se comunicarem entre si, podendo assim extrair informação e tomar decisões sobre o ambiente em que estão inseridos, sem que muitas vezes necessite de algum tipo de intervenção humana. Pode-se notar que é uma descrição bem parecida ao da Internet das Coisas, descrita por Atzori (2010). O M2M é regido por protocolos, sendo estes os principais: *Extensible Messaging and Presence Protocol* (XMPP) e *Advanced Message Queue Protocol* (AMQP).

Para permitir a implementação da IoT existem muitos requisitos que devem ser atendidos, como: protocolos de rede e aplicação empregados, tipo de tecnologia em meio físico *wireless* empregada, serviços de conexão à nuvem, consumo de energia e outros [22].

Vê-se na Figura 2, que a Internet das coisas pode ser vista por três principais paradigmas: uma visão focada no termo orientado a “Internet” (*middleware*) que observa a IoT do ponto de vista de rede, outra visão foi focada no termo orientado a “Coisas” (sensores), onde temos uma busca pela integração de objetos com uma única estrutura em comum, e uma última visão que foca nas questões relacionadas a semânticas (conhecimento), levando em conta a rede IoT do ponto de vista da comunicação entre vários dispositivos diferentes [23].

Figura 2- Três principais paradigmas da Internet das Coisas.



Fonte: Adaptado de [21].

Para se estabelecer as políticas de segurança é preciso rever alguns conceitos que são utilizados na infraestrutura da Internet das Coisas, especialmente: rede de sensores, *middleware* e computação em nuvem [25].

- Redes de sensores sem fio:

O sensor é um dispositivo indispensável na camada de percepção que compõe a infraestrutura da IoT, conhecido também como nó de sensor, é um elemento pequeno e autônomo que fica responsável pela comunicação e coleta dos dados físicos do ambiente [26]. O *Wireless Sensor Network* (WSN), rede de sensores sem fio, é o manejo de uma elevada quantidade de sensores que ficam dispersos num ambiente físico, sem o apoio de qualquer infraestrutura física estabilizada. Pode-se supor que os nós de sensores são capazes de monitorar o meio, coletar dados para posteriormente ser possível processá-los e em seguida enviá-los a um ou mais pontos de coleta, tudo isso individualmente [27].

Os sensores são distribuídos no ambiente levando em conta os requisitos de aplicação e o espaço geográfico de interesse. A dispersão quando planejada e fixa pode ser chamada de normal,

quando os nós estão posicionados de forma aleatória passa a ser dispersão aleatória, e móvel quando os nós possuem algum meio de locomoção. A distribuição influencia de forma direta no desempenho e na escolha do protocolo de roteamento dos sensores, e também no consumo de energia. A distribuição dos sensores no ambiente depende dos requisitos da aplicação e do espaço geográfico de interesse [27]. Porém, alguns obstáculos devem ser enfrentados, como a segurança e a privacidade de informações coletadas, sem, no entanto, produzir mais consumo de energia e sobrecarga de processamento na rede de sensores sem fio [28].

Como sensores são capazes de manipular informações sensíveis, tais como a intimidade, parâmetros de saúde e outros, surge a questão de segurança, para a garantia de manipulação de dados, sem ataque vindos de terceiros [25].

O trabalho de Kumari (2010) coloca em destaque algumas das principais ameaças de caráter físico e dá exemplos das ameaças que apresentam em cada uma das camadas de comunicação da IoT sendo elas:

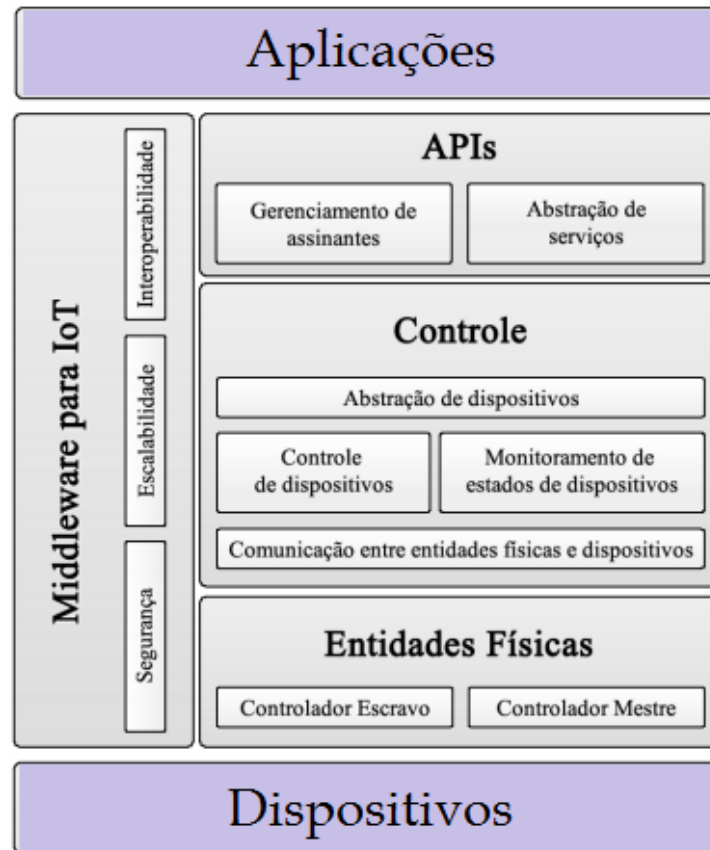
- a) Subcamadas física: "Jamming";
- b) de rede (Networks): "Selective Forwarding", "Black hole" ou "Sinkhole", "Hello Flood", "Wormhole", "Sybil Attack";
- c) de transporte: "Flooding Attack"; "Denial of Service" (DoS);
- d) de aplicação: invasão e comprometimento dos dados da rede de sensores ("Wireless").

Após a realização deste levantamento, o estudo feito por Kumari (2010) exemplifica alguns protocolos para melhorar a segurança de uma rede de sensores sem fio, tais como: confidencialidade, autenticação, integridade, atualidade, disponibilidade, auto-organização, sincronização temporal e localização segura.

- "Middleware".

O estudo feito por Ferreira (2014) mostra na Figura 3 uma proposta de "*Middleware*", que integra dispositivos e aplicações, contendo duas rotinas, a de controle e a de monitoramento de estados dos dispositivos.

Figura 3 - Rotinas de Middleware



Fonte: Adaptado de [29].

A partir da Figura 3, Ferreira (2014) explica que a primeira rotina é capaz de enviar requisições de controle de uma aplicação para um dispositivo. Isso quer dizer que as aplicações trocam dados com o "middleware" através de abstrações de servidores que são providas por APIs (application programming interfaces), assim o "middleware" pode traduzir as requisições de aplicações em comandos executáveis, transmitindo os comandos corretos para o dispositivo alvo. A segunda rotina é capaz de monitorar um dispositivo e capturar seu estado assim, se este receber alguma alteração, ela irá notificar quais são as novas aplicações deste novo estado. Para isso acontecer, a entidade física monitora o dispositivo obtendo seu estado, e quando há a percepção de uma modificação, comunica o novo estado de um controlador escravo para um controlador mestre.

As propostas de protocolo encontradas na literatura para a segurança do "Middleware" são a de adoção do controle de acesso ou ao uso de criptografia na camada "WebServices". Porém

estas abordagens não são suficientes para o tratamento de todas ameaças existentes na infraestrutura de Internet das coisas, principalmente devido a necessidade de se tratar as questões de segurança e privacidade dos usuários [30].

Segundo Casado (2009) o sistema *Contiki*, voltado para sistemas operacionais de dispositivos inteligentes, propõe uma solução de segurança para a camada de rede de WSN, entretanto, ainda está em desenvolvimento os trabalhos realizados [32] e, deste modo, se encontram no início de estudos de proteção para os dispositivos que eles gerenciam [25].

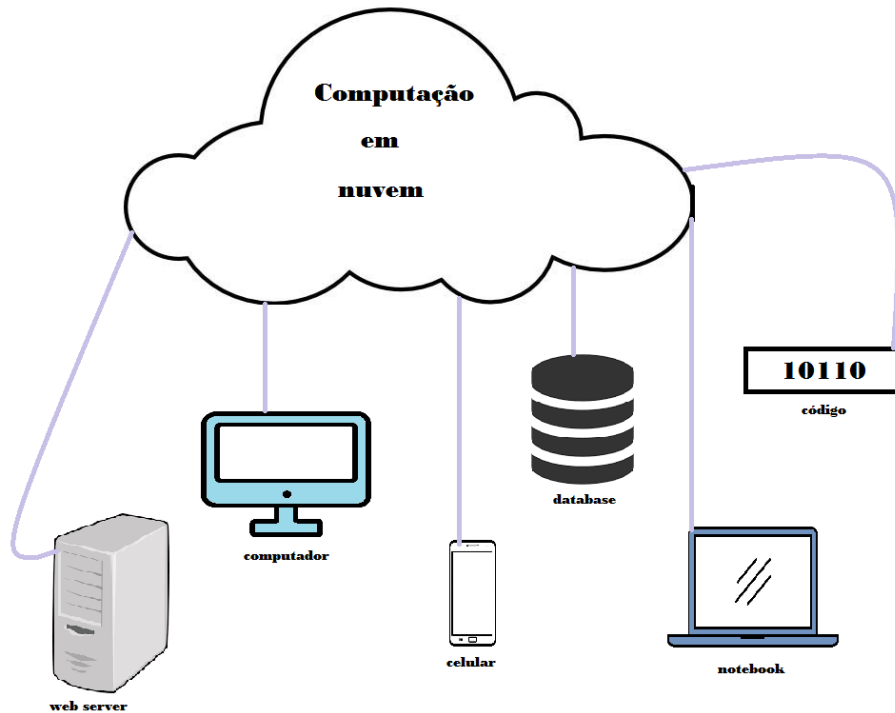
- Computação em Nuvem.

A Computação em nuvem é uma tecnologia importante para a implantação das camadas da infraestrutura de IoT. É a abordagem de uma metáfora para a plataforma computacional utilitária em destaque, em virtude da expansão do acesso à banda larga, da intercomunicação de objetos inteligentes e da disseminação de serviços e aplicativos online [33].

Segundo Ruscel (2010) computação em nuvem consiste na ideia de se utilizar, em qualquer lugar e plataforma, os variados tipos de aplicações que a internet proporciona com a mesma facilidade de tê-las instaladas no nosso próprio computador. Nenhuma abordagem para a utilização real foi tão global e completa: não apenas recursos de computação e armazenamento são entregues sob demanda, mas toda a pilha de computação pode ser aproveitada na nuvem.

Eles demonstram na Figura 4 uma visão geral de uma nuvem computacional.

Figura 4- Esquema de computação em nuvem.



Fonte: A autora.

O NIST Mell (2009) propõe que a infraestrutura de computação em nuvem possui cinco características essenciais: serviço de auto atendimento sob demanda; acesso por meio de banda larga; "pool" de recursos computacionais (processamento, memória, rede e armazenamento); rápida elasticidade com o propósito de prover maior escalabilidade de serviços e medição transparente da utilização dos recursos/serviços.

Consequentemente, para se ter uma ampla adoção da computação em nuvem deverá haver a superação de desafios em segurança, interoperabilidade e portabilidade [35]. Desafios bem parecidos com os enfrentados pela Internet das Coisas. Esta preocupação tem despertado a curiosidade da academia e da indústria, afim de buscar soluções para estes diversos problemas [25].

Para se alcançar a implementação do sistema de gerenciamento da segurança da informação no ambiente computacional em nuvem o "framework" [36] usa o padrão NIST-FISMA [37] e executa os requisitos previstos nas seis fases da norma:

- 1) serviço de categorização de segurança;

- 2) seleção dos controles de segurança;
- 3) implementação dos controles de segurança;
- 4) avaliação dos controles de segurança;
- 5) serviço de autorização e
- 6) serviço de monitoramento.

A arquitetura do "framework" tem três camadas: gerenciamento, aplicação e retroalimentação [36]. A primeira camada, gerenciamento, é a captura dos requisitos de segurança das partes interessadas. A camada de aplicação é a que se compromete com a gestão da segurança em termos da seleção de controles ajustados aos riscos identificados. E a camada de retroalimentação possui dois principais serviços: monitoramento e análise dos dados coletados. A combinação destes dois últimos serviços permite alcançar as metas de segurança pré-estabelecidas.

Pode-se ver que para o futuro tem-se a necessidade de desenvolver uma solução na perspectiva de rede de sensores que efetue o devido tratamento aos aspectos de autenticação, controle de acesso, sem, no entanto, comprometer o desempenho e sobrevivência da rede de sensores, ou seja, manter a configuração escalável e adaptativa ao contexto sem perder o objetivo na avaliação de segurança [38].

2.2 – Protocolos para redes de sensores e camadas IoT

Na construção da rede de internet das coisas se destaca uma tecnologia importante, o padrão IEEE 802.15.4 sendo um padrão para redes sem fio, de baixo consumo e baixa capacidade. A internet das coisas irá utilizar o protocolo 6LoWPAN, RPL e o IPv6 fornecerá o endereçamento e conectividade necessária para os dispositivos embarcados [39].

O protocolo 6LoWPAN significa “Internet Protocol version 6 over Low power Wireless Personal Area Networks.” Para que os dispositivos possam funcionar com o protocolo necessita-se de certas características, como: baixa capacidade de processamento; baixa capacidade de memória; baixo gasto de energia; baixo alcance do sinal e baixa taxa de transmissão [40].

Tem-se que a rede convencional de computadores adota um padrão de camadas, logo a IoT também faz uso de padronizações que possui abordagens parecidas. No momento ainda está sendo criados alguns padrões, onde vários grupos de trabalhos estão interagindo e tentando criar modelos de camadas e novos protocolos para a IoT. Entre estes grupos estão o IETF e o IEEE [22]. No Quadro 1 pode-se ver alguns protocolos que foram criados e vem se destacando nesta área de aplicação.

Quadro 1- Padrão proposto para IoT.

Protocolo de Aplicação		DDS	CoAP	AMQP	MQTT	MQTT-SN	XMPP	HTTP REST
Descoberta de Serviços		mDNS				DNS-SD		
Protocolos de Infraestrutura	Protocolo de Roteamento	RPL						
	Camada de Rede	6LoWPAN RPL				IPv4/IPV6		
	Camada de Enlace	IEEE 802.15.4						
	Camada Física	LTE-A		EPCglobal		IEE 802.15.4		Z-Wave

Fonte: Adaptado de [22].

Na Figura 5 pode-se ver que a rede IoT está dividida em 5 camadas, como: camada de percepção; camada de rede; camada de middleware; camada de aplicação; e a camada de negócios [41].

Figura 5- Camadas de IoT.

Camada de negócio			
Sistema de gerenciamento	Modelo de negócios	Fluxogramas	Gráficos
Camada de aplicação	Aplicações inteligentes e gestão		
Camada de Middleware			
Informações de processamento	Gestão de serviços	Banco de dados	Unidade de decisão
Camada de rede			
	Transmissão segura	3G, UMTS, WiFi, Bluetooth, Infravermelho, ZigBee	
Camada de percepção			
	Objetos físicos	RFID, Códigos de barras Sensores infravermelho	

Fonte: A autora.

Camada de percepção - Nota-se que é na camada de percepção que temos os objetos físicos e os sensores. Esta camada é a responsável pela identificação de objetos por dispositivos sensores, dependendo do tipo do sensor temos variadas informações, como: temperatura, pressão, velocidade, luminosidade e outras. Toda a informação aqui contida será passada para a próxima camada que é a de rede [41].

Camada de rede – Na camada de rede tem-se um sistema de transmissão, ela transfere as informações advindas dos sensores para o sistema de processamento. Sendo essa transferência feita de forma segura e confiável. A esta transferência pode ser feita por meio de tecnologias como o 3G, UMTS, Wifi, Bluetooth, Infravermelho e ZigBee, podendo conter transmissão com fio ou sem fio. Então, as informações da camada de rede são transferidas para a camada de Middleware.

Camada de Middleware – Nesta camada será recebido as informações vindas da camada de rede e são armazenadas no banco de dados. Ela cuida de tomadas de decisão para suprir as informações de processamento, já que a rede IoT possui vários dispositivos que

forneçam vários serviços. E cada dispositivo comunica-se entre si, desde que haja a implementação do mesmo tipo de serviço.

Camada de aplicação – Esta camada é responsável pelo gerenciamento global do aplicativo, é possível por meio das informações obtidas na camada Middleware, é nessa camada que se encontram variadas aplicações que são implementadas pra rede IoT.

Camada de negócios – Essa camada é responsável pelo gerenciamento do sistema global, incluindo as aplicações IoT e serviços. São construídos os modelos de negócios, os gráficos, os fluxogramas, entre outros, isso tudo com base nos dados recebidos [41].

2.2.1 – Protocolos de Aplicação

Como visto anteriormente a camada de aplicação é responsável pelo gerenciamento global, logo ela faz a interação final com o usuário, sendo assim ela é de grande importância dentro do contexto de IoT. Nessa camada há a interface entre pessoas e objetos, segue abaixo protocolos utilizados:

CoAP (*Constrained Application Protocol*): é um protocolo web que roda sobre o protocolo UDP (*User Datagram Protocol*) e é projetado principalmente para a Internet das coisas. CoAP é uma variante do protocolo síncrono mais usado da web, o HTTP (*Hypertext Transfer Protocol*) e é adaptado para dispositivos restritos e comunicação máquina a máquina foi desenvolvido especificamente para aplicações em IoT [46]. Segue o estilo de arquitetura REST (*Representational State Transfer*), que é um jeito mais simples do modelo servidor-cliente. CoAP usa UDP, o que agrega ainda mais na simplicidade do funcionamento e economiza em processamentos. As principais vantagens deste protocolo são a simplicidade, integração com serviços *web*, suporte a *multicast* e possui um cabeçalho bem pequeno [21;43;44].

HTTP REST: Como visto anteriormente, este protocolo é mais simplificado que o protocolo HTTP, e é integrado aos serviços web de forma mais fácil. Ele é baseado em uma arquitetura de servidor-cliente, onde o servidor é um nó pai que controla o acesso e os recursos de rede, cabendo ao cliente apenas requisitar o acesso. Faz uso das mesmas funções que são utilizadas no HTTP, como GET, PUT, DELETE e POST [43;45].

MQTT (*Message Queue Telemetry Transport*) é um protocolo da camada de aplicação projetado para dispositivos com restrição de recursos [23]. Como no HTTP (*Hypertext Transfer Protocol*), o MQTT depende do Protocolo de Controle de Transmissão (TCP) e IP como suas camadas subjacentes. Entretanto, quando comparado ao HTTP, o MQTT foi projetado para ter uma sobrecarga de protocolo menor. A confiabilidade das mensagens no MQTT é atendida por três níveis de Qualidade de Serviço (QoS). No nível 0 do QoS a mensagem é entregue somente uma vez e não há nenhum reconhecimento de recepção que seja necessário. No nível 1 toda mensagem é, pelo menos uma vez, entregue e é necessário a confirmação de recepção da mensagem. No nível 2, um mecanismo de handshake de quatro vias é usado, assim a entrega de uma mensagem será feita apenas uma vez [47].

Quadro 2 - Diferenças entre MQTT e CoAP.

	MQTT	CoAP
Camada de aplicação	Completamente Single Layered, dispositivos com restrição de recursos	Single Layered com 2 conceitos de sub-camadas (Messages Layer e Request Response Layer), suporte <i>amulticast</i> , integração com serviços <i>web</i> .
Camada de transporte	TCP	UDP
Mecanismo de confiabilidade	3 níveis de qualidade de serviço.	Mensagens confirmadas, mensagens não confirmadas, confirmações e retransmissões
Arquiteturas	Servidor-cliente do tipo <i>publish/subscribe, Broker</i> .	REST (<i>Representational State Transfer</i>), recurso, observação/Servidor-cliente,

Fonte: A autora.

MQTT-SN: É uma versão do MQTT para redes de sensores que suporta o uso do protocolo UDP como protocolo de transporte [21].

DDS (*Data Distribution Service*): este protocolo faz o uso da arquitetura *publish/subscribe*, mas diferente do MQTT não faz o uso do *broker*, o gerenciamento é feito por meio de camadas específicas deste protocolo que posteriormente são distribuídos por *multicast*. A principal vantagem que este protocolo possui é possuir uma alta qualidade de serviço (QoS), que acaba proporcionando uma melhor confiabilidade [21;45].

DTLS (*Datagram Transport Layer Security*): é um protocolo de segurança bastante completo que pode realizar a autenticação, troca de chaves, e a proteção de aplicação dos dados com o chaveamento de materiais e algoritmos de codificação. Ao se utilizar do DTLS, a segurança e a proteção para IoT, podem ser alcançadas de forma satisfatória [42].

AMQP (*Advanced Message Queuing Protocol*): é um protocolo orientado a troca de mensagens corporativas que preza a interoperabilidade entre diversos dispositivos e linguagens de programação, por este motivo é bastante utilizado no meio comercial [22;45]. Um sistema de mensagens AMQP consiste em 3 componentes principais: *Publisher(s)*, *Consumer(s)* e *Broker/Server(s)*. Ele fornece entrega de mensagens confiável, garantida e em ordem [47].

XMPP (*Extensible Messaging and Presence Protocol*) : este protocolo faz uso da linguagem XML para troca de mensagens instantâneas. Costuma ser implementado em arquiteturas do tipo servidor/cliente, mas suporta também outros tipos de modelos, como: *publish/subscribe request/response*. Possui várias extensões e por isso é bastante maleável e funciona em diversos sistemas operacionais [45;48].

Pode-se observar que a camada de aplicação conta com diversos protocolos para uma ampla variedade de serviços. O Quadro 3 resume as principais características dos protocolos de aplicação aqui citados.

Quadro 3 - Comparação entre protocolos de aplicação para IoT.

Protocolo	RFSTfull	Transporte	Publish/subscribe	Request/response	QoS	Tamanho de cabeçalho
CoAP	✓	UDP	✓	✓	✓	4
HTTP	✓	TCP	✗	✓	✗	-
MQTT	✗	TCP	✓	✗	✓	2
DDS	✗	TCP UDP	✓	✗	✓	-
AMQP	✗	TCP	✓	✗	✓	8
XMPP	✗	TCP	✓	✓	✗	-

Fonte: Adaptado de [21;45].

2.2.2 – Protocolos de rede e tecnologias de comunicação

Existem protocolos e tecnologias de comunicação que possuem curto alcance, e são comumente usados nas aplicações de IoT, como: ZigBee, 6LoWPAN, RPL, Wi-fi, Bluetooth, Thread [49]. O IEEE 802.15 é adotado por diversos grupos que atuam no desenvolvimento de objetos e protocolos IoT. Ele define protocolos e regras para a comunicação de dados entre dispositivos utilizando baixa taxa de transferência, com baixo consumo e complexidade. Abaixo será feita uma breve discussão sobre cada tecnologia citada anteriormente:

ZigBee - é um protocolo de comunicação de padrão aberto desenvolvido para aplicações IoT de baixo consumo baseado no padrão IEEE802.15.4 criado pela ZigBee Alliance. A implementação dele é barata e mais simples, por isso é comumente usado no desenvolvimento de redes de sensores, sendo um dos seus principais diferenciais o suporte a rede mesh, permitindo um maior alcance [23;50;51].

Bluetooth - é um sistema que utiliza uma frequência de rádio de onda curta (2.4 GHz), baseado no padrão IEEE 802.15.1, para criar uma comunicação entre aparelhos habilitados. Como seu alcance é curto e só permite a comunicação entre dispositivos próximos, seu consumo de energia é bem baixo. Apresenta uma taxa de dados relativamente alta, que pode chegar até 1

Mbps, seu consumo de energia baixo pode ser visto melhor na sua mais recente versão 4.0, também conhecida como *Bluetooth Low Energy* (BLE) [22;50].

Wi-Fi - esta tecnologia é baseada no padrão IEEE 802.11, opera na faixa de 2.4 e 5.8 GHz. Possui taxas de dados relativamente altas, e permite a conexão de dispositivos diretamente na Internet por meio do protocolo IP. Porém, possui um número pequeno de canais, o que limita o número de dispositivos que podem se conectar ao mesmo tempo, tendo um consumo de energia elevado em relação às outras tecnologias citadas [22;50].

Thread - esta tecnologia, assim como o Zigbee, define camadas a serem em cima do padrão IEEE 802.15.4. Entretanto não fornece a camada de aplicação, utiliza-se sempre de padrões abertos de protocolos para garantir integridade dos pacotes, utilizando os protocolos UDP e 6LoWPAN. Fornece também serviço de segurança e tolerância a erros [52; 53]

6LoWPAN -Este protocolo promete conectar dispositivos com recursos limitados de rede ao domínio IP [50;54], sendo essa a sua principal vantagem, pois não precisa de tradução entre protocolos para se conectar à Internet e garante alta interoperabilidade[49]. O protocolo IPv6 surgiu para substituir o IPv4, pois este possui um número de endereços possíveis menor que o IPv6. Como a IoT pretende conectar bilhões de dispositivos à Internet, o IPv6 torna-se um protocolo essencial para a sua implementação [55;56]. Outra vantagem do 6LoWPAN é o protocolo de roteamento RPL (*Routing Protocol for Low power and lossy networks*). RPL usa topologia *meshe* foi desenvolvido para redes de sensores de coleta de dados, o que o faz adequado para sistemas AMI [54].

RPL- é o protocolo de roteamento usado na camada de rede em IoT, que possui roteamento para baixa potência e rede com perdas. Temos que a topologia RPL contém um nó raiz/coletor conectado diretamente à Internet usando o 6BR (*IPv6 Border Router*). A topologia RPL forma a árvore DODAG (Destination Oriented Directed Acyclic Graph), que contém apenas 1 nós raiz, então o nó raiz começa a formação da topologia transmitindo as mensagens DIO (DODAG Information Object). Logo após, os nós que recebem a mensagem DIO selecionam o pai para o remetente respondendo a mensagem DAO (Objeto de Anúncio de Destino), pedindo a permissão

para se juntar ao nó, então o nó pai irá permitir o ingresso ao enviar a mensagem DIO ACK, informando ao nó que o mesmo pode se juntar a ele. O valor da classificação pode depender da distância do nó raiz, energia do link, etc, sendo o proprietário capaz de decidir quais os parâmetros de cálculo do valor da classificação serão usados. Então, se o novo nó quiser entrar na rede, ele primeiro irá perguntar se há algum DODAG, logo após enviando a mensagem DIS (solicitação do DODAG info). Então os nós continuam transmitindo a mensagem DIO, formando a topologia de árvore [57]. O Quadro 4 mostra uma comparação dos protocolos que são comumente usados em uma rede IP tradicional e uma rede IoT.

Quadro 4 - Protocolos usados por uma rede IP tradicional e IoT.

Protocolo IoT	Camadas	IP tradicional protocolo
CoAP	Aplicação	HTTP,FTP,SMTP, ect.
UDP	Transporte	TCP/UDP
IPv6 + RPL 6LoWPAN	Rede	IPv4/IPv6 + AODV/DSR/DSDV
802.15.4	Link	802.3/802.11

Fonte: Adaptado de [57].

2.3 – Resumo

Neste capítulo foram abordados os principais aspectos tecnológicos envolvidos na IoT, assim como os principais protocolos propostos. Nas subseções foi apresentado uma breve descrição dos protocolos de aplicação e comunicação que têm sido estudados como fortes candidatos a uso pela IoT. No próximo capítulo tem-se um levantamento de ataques que a internet das coisas pode vir a sofrer, métodos de defesas existente e os danos que cada um pode causar a rede. No capítulo 5 será descrito os materiais e métodos que serão empregados nesta pesquisa para se construir a estrutura tecnológica desejada para aplicação em IoT. Dando enfoque ao sistema *Contiki* com o simulador *Cooja*, mostrando uma topologia e arquitetura contendo o protocolo RPL.

CAPÍTULO 3

SEGURANÇA EM IOT

3.1 - Problemas de segurança

Com a evolução da internet das coisas, considerações sobre a segurança não é novidade e precisam ser feitas. Como o IoT possui características novas em sua implementação, temos novos desafios na segurança, logo, é de suma importância que estes desafios sejam enfrentados e que soluções sejam propostas, afim de encontrar soluções capazes de garantir a segurança do usuário, dos objetos interligados e os produtos oferecidos. Se os serviços e produtos ofertados pelo IoT não estiverem protegidos, os mesmos poderão ser alvos de ataques de terceiros, levando a exposição de dados sensíveis e também a uma sobrecarga de energia.

Na IoT há uma interconexão de dispositivos, onde vários objetos se encontram conectados, caso um deles tiver sua segurança comprometida e conectado à internet irá afetar todo o conjunto de dispositivos conectados, assim prejudicando a segurança e a resiliência da internet. Na IoT, o usuário precisa confiar de que sua segurança estará sendo resguardada, mesmo com a implantação em massa de dispositivos ligados à internet, existe a necessidade de que suas informações e produtos estejam devidamente protegidas contra uso indevidos e danos causados por ataques externos. É mais fácil e útil conceituar a segurança dos objetos IoT, como um espectro de vulnerabilidade que os mesmos estão sujeitos no dia a dia. Este espectro engloba desde dispositivos com alta segurança empregada até dispositivos que podem estar totalmente desprotegidos.

Sabe-se que a IoT ainda se encontra bem longe de estar completamente segura contra os problemas de segurança que afligem a internet, ainda mais levando em consideração de que a mesma enfrenta questões a mais de segurança, devido a suas peculiaridades e desafios, abaixo citamos algumas:

- Comunicações feitas por meio de redes sem fio, onde qualquer indivíduo com más intenções pode interceptar transmissões, se comunicar através desse meio e modificar o que o usuário vê na tela de seu dispositivo. É necessário um serviço de autenticação que se refere à garantia de que a comunicação entre os dispositivos

será autêntica, ou seja, o serviço irá assegurar de que a conexão não sofrerá interferência de um agente externo se passando por uma das partes legítimas.

- Os dispositivos podem ser acessados fisicamente, podendo ser colocados em locais públicos ao alcance de vários indivíduos. Precisa haver um controle de acesso, que consiste na prevenção do uso não autorizado de um recurso, com a capacidade de limitação do acesso.
- Na IoT é preciso a confidencialidade dos dados, onde tenha proteção das mensagens enviadas pelos usuários dentro de uma comunicação, impossibilitando que um atacante consiga observar o conteúdo transmitido e as características de informação.
- É preciso zelar pela integridade, onde haverá a garantia de que as informações serão mantidas em seu estado original, protegendo contra alterações indesejadas.
- O não-repúdio, é a proteção contra uma possível negação de uma das partes durante a comunicação, ou seja, ambas as partes podem comprovar que a mensagem foi recebida ou enviada.

Outro desafio à segurança está relacionado a sobrecarga adicional de recursos computacionais que pode vir a ocorrer, onde a capacidade de poder confiar nas mensagens transmitidas não é totalmente sem desvantagens. Pois, executar medidas de segurança como o código criptografado adicional para a garantia de que as comunicações serão confiáveis, pode acarretar na demanda de recursos computacionais como:

- Energia, onde o dispositivo deverá ser mantido em modo ativo e não em modo de baixo consumo de energia, consumindo mais energia utilizada para operar unidades de CPU durante a execução de algum algoritmo;
- Tempo adicional do CPU, o mesmo terá que dedicar mais tempo na execução de criptografias, tirando sua atenção de outros códigos relacionadas ao próprio CPU;
- Será consumido uma quantidade maior de memória RAM, chegando até mesmo a ser permanente caso o estado de operação precise;
- Afetará a memória permanente, pois o código aumentará de tamanho na adição de primitivas criptográficas;

Ao adicionar medidas de segurança, será necessário a escolha apropriada tanto de dispositivos, quanto de primitivas adotadas para melhorar a funcionalidade do dispositivo sem que o mesmo seja prejudicado no processo. Caso isto não seja feito, teremos alguns impactos negativos, como o aumento de consumo de energia, redução de funcionalidades, grande demanda de recursos, e diminuição da rapidez acarretando na insuficiência de resposta em tempo real.

3.2 - Segurança e privacidade.

Neste trabalho, segurança de dados e privacidade referem-se à proteção de quaisquer dados coletados ou armazenados em qualquer sistema de rede IoT. O que significa que o mesmo precisa fornecer confidencialidade, integridade e disponibilidade de dados. Podemos conseguir isto tudo por meio da utilização de autenticação, controle de acesso, criptografia de dados e disponibilidade de dados e redundância através de back-ups e etc [18].

Tem-se que a confiança consiste em diferentes propriedades e objetivos. Neste trabalho, a confiança refere-se ao cumprimento das metas de segurança explicadas acima, que consistem nos seguintes objetivos [18].

Figura 6 - Objetivos de confiabilidade.

Objetivos	Relações de confiança entre cada camada de IoT	Comunicação adequada e transição entre as diferentes camadas de uma rede IoT, e entre os diferentes nós existentes são necessários para a garantia de segurança de privacidade de dados
	Confiança e privacidade em cada camada de IoT	As metas de segurança e privacidade em cada camada do sistema IoT precisam ser preservadas em qualquer tipo de circunstância, fornecendo confiabilidade, integridade e confidencialidade de dados.
	Confiança entre o usuário e o sistema IoT	Em um certo ponto, o sistema IoT irá divulgar alguns dos dados disponíveis para o usuário final e vice-versa. Portanto, há necessidade de fornecer um nível de confiança entre o sistema IoT e o usuário final, para que seja possível o sucesso do emprego do conceito IoT. Como o usuário interage com o sistema IoT, suas ações afetam a exatidão da operação, criando a necessidade de classificar as ações do usuário.

Fonte: Adaptado de [18].

Quadro 5 - Principais requerimentos de segurança e seus sub componentes.

Segurança de rede	Gerenciamento de identidade	Privacidade	Confiança	Resiliência
Confidencialidade	Autenticação	Privacidade de dados	Confiança do dispositivo	Robustez contra ataques
Integridade	Autorização	Anonimato	Confiança de identidade	Resiliência contra falhas
Autenticidade	Prestação de conta	Pseudônimo	Confiança de dados	
Acessibilidade	Revogação	Indelabilidade		

Fonte: Adaptado de [62].

Segundo Vasilomanolaki et al. (2015) a resiliência, é a mesclagem de escala da IoT em termos de dispositivos, criando uma grande superfície que possibilita ataques e falhas. Por esse motivo, podemos falar que a resiliência e a robustez contra ataques e falhas aplicam-se, como requisitos importantes, à IoT.

3.3 - Ataques e contramedidas.

Pode-se dizer que com os avanços tecnológicos surgem novas maneiras e oportunidades de ataques cibernéticos. Quanto maior o crescimento de dispositivos que podem ser acessados, maior é a extensão de danos que podem ser causados. E com essa grande quantidade de dispositivos, menor é o controle sobre quais deles possuem as últimas medidas de segurança atualizadas.

Kalita e Kar (2009), fizeram uma enumeração e apresentação onde descrevem sobre vários ataques possíveis de afetar as redes sem fio podem sofrer e que atacam também as redes IoT, como estes:

- **Ataques de Negação de Serviço (DoS):** este ataque costuma fazer o uso de impedimento ou restrição do uso normal da rede ou da administração da rede, com ou sem fio.
- **Ataques Sybil:** São vários dispositivos maliciosos que de forma ilegítima faz uso de inúmeras identidades como se fossem sua;
- **Wormhole:** É o encaminhamento de mensagens recebidas em um link de baixa latência que depois as reproduz em outra parte diferente da rede com o propósito de interromper o roteamento;
- **Skinhole ou BlackHole (Buraco Negro):** Este ataque transforma um nó comprometido como o local mais atraente para troca de informações na perspectiva do roteamento, o que faz com que todo o tráfego que irá para uma área específica passe através deste nó;
- **Hello Flood:** Sabe-se que muito protocolos exigem que os nós transmitam pacotes HELLO para se anunciarem aos seus vizinhos, e um nó que recebe esse pacote pode assumir que ele está dentro do intervalo de rádio (normal) do remetente. Entretanto, essa suposição pode ser falsa: então um invasor irá transmitir no roteamento ou outras informações com potência de transmissão grande o suficiente que será capaz de convencer todos os outros nós da rede, de que o adversário é o seu vizinho e começar a trocar informações com os nós;
- **Encaminhamento seletivo:** Neste ataque os nós que estão comprometidos se comportam como buracos negros recusando a encaminhar as mensagens recebidas e simplesmente descartando-as;
- **Análise de Tráfego:** Este ataque monitora a transmissão via redes sem fio, com o propósito de identificar padrões de comunicação e também dos usuários envolvidos;
- **Espionagem:** Neste ataque um atacante externo pode monitorar de forma passiva redes sem fio, coletando dados, informações sigilosas e até mesmo credenciais de autenticação.

O trabalho de Ioannis et al. (2015), propõe um espectro mais amplo das vulnerabilidades de segurança e ataques em sistemas IoT. Eles classificam em diferentes ataques sob quatro classes distintas, como: Ataques físicos, de rede, de software e de criptografia. Um sistema IoT

pode ser atacado fisicamente ou atacado de dentro de sua rede, ou de aplicativos no sistema e, por último, de ataques a esquemas de criptografia. Como a IoT é implementada usando várias tecnologias de rede existentes (redes de sensores sem fio, RFIDs, Internet, etc.) há uma necessidade de uma categorização adequada dos ataques, de modo que encapsule todos os diferentes tipos de ameaças, para que melhores medidas possam ser desenvolvidas e implementadas para protegê-las. Um resumo da classificação dos ataques é mostrado no Quadro 6 abaixo.

Quadro 6 - Classificação de ataques IoT.

Ataques físicos	Ataques de rede	Ataques de software	Ataques de criptografias
Adulteração de nós	Ataques de análise de tráfego	Vírus	Ataques de canal lateral
Interferência Rf	RFID Spoofing	Spyware	Ataque de texto Cifrado
Node Jamming	RFID Cloning	Adware	Ataque de texto cifrado desconhecido
Injeção de nó malicioso	Autorização não permitida RFID	Cavalo de tróia	Ataque de código cifrado escolhido
Dano físico	Sinkhole	Scripts maliciosos	Man in the middle
Engenharia social	Man in the middle	Negação de serviço	
Ataque de privação de sono	Negação de serviço		
Injeção de código malicioso no nó	Routing information attacks		
	Sybil		

Fonte: Adaptado de [18].

Ainda não há soluções definitivas para todos os ataques que a IoT pode sofrer. Os ataques DoS podem ser evitados ou até mesmo diminuídos em algumas situações específicas, por exemplo, ao não processar mensagens que podem vir ser originadas de dispositivos que aparentemente transmitem um número maior que o previsto. Porém, o usuário pode investir em dispositivos que possui maiores recursos, e consequentemente com capacidade de gerar mais

colisões, ou dispositivos usados exclusivamente para inundação de tráfego de uma rede sem fio [64].

Deve-se levar em consideração uma medida importante na segurança, que é a prevenção de adulterações físicas, visto que estas podem destruir e ou danificar os nós de uma determinada rede, o que permite efetivamente a reprogramação de nós, engenharia reversa e aquisição de chaves criptografadas [63].

Outra contramedida que pode ser usada para melhorar as questões de privacidade dos dados e diminuir o número de ataques bem-sucedidos, é o uso de mensagens seguras por meio de medidas criptografadas inclusivamente nas camadas de roteamento [63]. Quando as mensagens possuem criptografia elas garantem que um invasor não tenha acesso ao conteúdo sem possuir uma chave que possa descriptografar. E mensagens que são autenticadas corretamente não podem ser falsificadas. Dispositivos externos não poderão adulterar as identidades quando a autenticação for pedida. Os atacantes que não participam da rede autenticada não irão ter acesso a transmissão de mensagens falsas afim de atrair o tráfego.

Outra contramedida a ser usada é o projeto de protocolos meticulosos, que possuem políticas de prevenção de ataques, além de garantir a comunicação segura [63]. Estes protocolos permitem que as partes confiáveis tenham participação nas decisões de roteamento. Ao combinar criptografia e protocolos protegidos irá ter a mitigação na análise de tráfego, garantindo que todas as mensagens tenham o mesmo tamanho, e diminuir significativamente a transmissão de mensagens falsas durante uma comunicação.

Contra ataques de Wormhole pode-se usar medidas com abordagens de nível superior e abordagens de comportamento afim de ajudar na detecção desses ataques, identificando os nós que foram replicados em diferente locais da rede [63]. Deve-se planejar com bastante cuidado os protocolos de roteamento para que os mesmos possam vencer estes tipos de ataques, constatar este ataque não é muito simples pois os dispositivos aparentemente podem funcionar de forma normal no ponto de vista da rede, mas ao mesmo tempo estar à espera de um gatilho ou condição de disparo com uma comunicação em um dispositivo fora da percepção da rede.

3.4 - Resumo

Neste capítulo foram abordados os principais aspectos de segurança e privacidade envolvidos na IoT, teve-se um resumo do que é segurança, foram citados vários tipos de ataques que podem afetar as redes IoT e algumas contramedidas que podem ser tomadas para melhorar a segurança dos dispositivos. No próximo será descrito os materiais e métodos que serão empregados nesta pesquisa para se construir a estrutura tecnológica desejada para aplicação em IoT.

CAPÍTULO 4

MATERIAIS E MÉTODOS

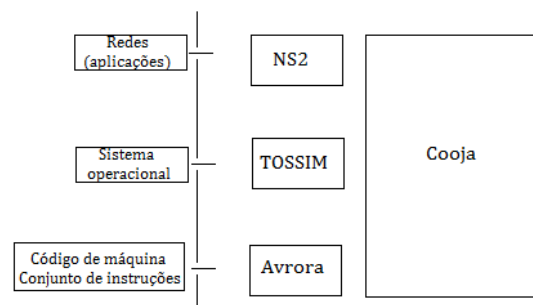
4.1 – Ambiente e cenário de simulação.

O sistema operacional *Contiki* trabalha juntamente com o simulador *Cooja*, possibilitando que a simulação dos comportamentos das aplicações advindas da rede IoT, em diversos cenários. Este simulador apresenta diversos relatórios a respeito do hardware (sinalização de rádio, alcance, potência, energia, luzes, botões, etc.) e das comunicações que ocorrem entre os dispositivos da rede (mensagens, protocolos, endereços, temporização, etc.). Esses relatórios podem ser visualizados em tempo de simulação, através da opção collect-view [58].

Um sistema operacional bem conhecido para o IoT, o *Contiki* possui uma implementação do RPL, *ContikiRPL* [15]. O modo de armazenamento *ContikiRPL* usa roteamento dentro da rede, onde os nós rastreiam todos os descendentes. Será demonstrado ataques contra uma rede RPL, simulada no ambiente *Cooja*. Nas simulações feitas neste trabalho, usa-se nós emulados do Tmote Sky executando o *ContikiRPL*.

A configuração de *Cooja* é flexível, dessa forma várias partes do simulador podem ser substituídas com uma funcionalidade adicional. Além disso, o simulador permite uma configuração dos recursos de hardware e do software dos dispositivos, os nós, e também dos periféricos. O *Cooja* faz parte do sistema operacional (SO) *Contiki* [15], que é de código aberto, escrito em linguagem C. Os níveis de simulação do *Cooja* são ilustrados na Figura 7 em comparação com outros simuladores como *NS2* [59], *TOSSIM* [60] e *AVRORA* [61].

Figura 7 - Comparação entre simuladores.



Fonte: A autora.

O cenário corresponde a 10, 15 e 20 nós sensores que são espalhados ao redor do nó raiz em um raio de 100 metros, na configuração para observar o ataque é colocado um nó malicioso no meio da mesma topologia. Os nós usam o canal sem fio, seguindo o modelo de propagação *Unit Disk Graph Medium (UDGM)* podendo assim os nós estabelecerem comunicação dentro da rede. O tempo de simulação estabelecido nesta avaliação é de 120s, e ele foi definido a fim de que os nós possam trocar uma quantidade de pacotes, e a simulação ocorra sem erros. O nó atacante age de forma maliciosa durante todo o período da simulação e no momento do encaminhamento dos dados. Os resultados apresentados são as médias de várias simulações. Foi usado a configuração e framework feito por Alexandre D'Hondt et. al¹, o mesmo faz uso da modificação de blocos de construção para a implementação do ataque, sendo possível alterar depois os parâmetros de acordo com as suas necessidades, no Quadro 7 pode-se ver quais foram os parametros escolhidos para este trabalho, os mesmos foram escolhidos para demonstrar a diferença de consumo com números diferentes de nós, e que coubessem de forma organizada na área que já é demarcada por Alexandre D'Hondt et. al.

Quadro 7 - Parâmetros de simulação do cenário.

Parâmetro	Valores
Tipo de nó sensor	Tmote Sky mote
Número de nós	10,15,20
Tempo de simulação	120s
Raio de alcance	100m
Área	200m
Canal sem fio	Unit disk graph Medium (UDGM)
Número de nós maliciosos	1

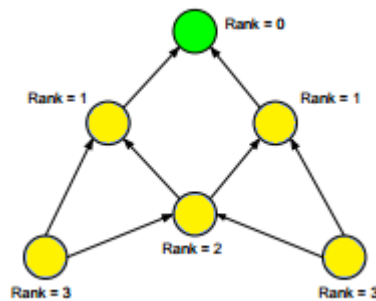
Fonte: A autora.

¹ <https://github.com/dhondta/rpl-attacks>

4.2 - Testes da rede

Após a implementação da rede, é interessante que se avalie um de seus parâmetros mais básicos, porém muito importante, o consumo de energia gerado por cada nó participante. A Figura 8 ilustra um arranjo básico de dispositivos de rede executando RPL, estes estão conectados de forma acíclica. Temos um Grafo Acíclico Direcionado Orientado ao Destino (DODAG), cada nó irá manter o melhor destino para a raiz do DODAG, para que o melhor caminho seja encontrado os nós usam uma função objetivo que define a métrica de roteamento a ser computada.

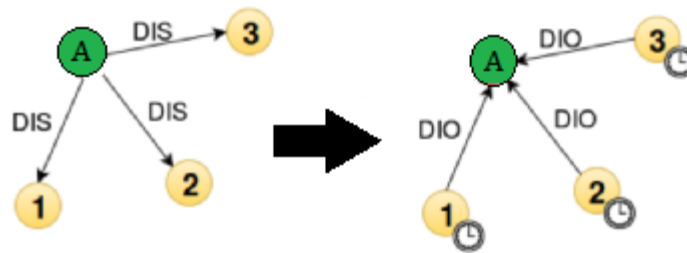
Figura 8 - DODAG RPL.



Fonte: A autora.

A Figura 9 ilustra a comunicação entre esses nós, com as mensagens DIS e DIO.

Figura 9 - Direção de comunicação.



Fonte: A autora.

O RPL relaciona três tipos de mensagens utilizando o ICMPv6: *DODAG Destination Advertisement Object (DAO)*, *DODAG Information Object (DIO)* e *DODAG Information Solicitation Message (DIS)* descritas a seguir:

DIO: essas mensagens são utilizadas para anunciar um DODAG e suas respectivas características. Sendo usadas para a descoberta de DODAGs com a sua formação e manutenção, o intervalo em que essas mensagens são enviadas é controlado de modo eficiente pelo algoritmo Trickle [60].

DIS: esta mensagem é parecida com as que pedem solicitação de rotas (RS) do IPv6, são comumente usadas para descobrir DODAGs na vizinhança e logo após fazer a solicitação DIOS de nós vizinhos, sem possuir um corpo de mensagem.

DAO: estas mensagens são utilizadas durante a ocorrência do processo de notificação de rotas descendentes. Elas são enviadas em sentido ascendente (pelo nós que querem receber mensagens dos nós raízes, conhecidos como pai) para propagar informações de destino ao longo do DODAG.

Neste trabalho tem-se que, o framework foi construído em Python usando o *Contiki* juntamente com a ferramenta de simulação *Cooja*. Sabe-se que as simulações usam um Javascript para poder automatizar a coleta de dados por meio dos plugins disponíveis no *Cooja*, podendo coletar vários parâmetros como o consumo de energia, coleta de mensagens, registros do RPL e o relacionamento dos nós. Para que a implementação dos ataques fique mais fácil, o nó malicioso pode ser ajustado em uma biblioteca RPL modificada externa.

Necessita-se instalar a máquina virtual *Contiki* em um sistema que seja *Linux*, para isso utiliza-se neste trabalho o *VMware workstation 12 player*, após a instalação da máquina virtual pode-se utilizar suas ferramentas, em um primeiro momento testa-se exemplos básicos do próprio software *Cooja* afim de verificar se o funcionamento está ocorrendo sem erros. Neste trabalho utiliza-se do framework feito por Alexandre D'Hondt et. al, para que o mesmo funcione é preciso baixar bibliotecas.

No processo de instalação baixou-se os requerimentos do sistema, neste momento houve falhas pois alguns desses requerimentos necessitam de certas versões de outras bibliotecas, buscou-se na internet, principalmente no site da GitHub, soluções para os problemas enfrentados. Pode-se fazer os updates necessários e solucionar problemas que atigem outros usuários, por meio de fóruns no site citado. Apesar de todos os problemas encontrados durante a instalação do framework, com o tempo foi possível fazer com o mesmo funcionasse.

CAPÍTULO 5

RESULTADOS E DISCUSSÕES

5.1 –Funcionamento da rede

O sistema operacional *Contiki* disponibiliza alguns códigos de aplicações para exemplificar casos de uso de uma rede baseada no RPL. Estes códigos podem servir de base para aplicações mais elaboradas. Um exemplo interessante, e que aqui foi usada para demonstrar o funcionamento da rede, é o uso de bibliotecas para a alocação de memória, abstrações de comunicação e mecanismos de redes de rádios de baixa potência. Um nó no *Cooja* é um sistema *Contiki* real compilado e executado. O cenário utiliza de base a distribuição de nós dentro de uma sala onde há vários dispositivos sendo usados, considerando uma situação real onde existe diversas pessoas e objetos conectados. As Figuras 10 , 11 e 12 ilustram as topologias de rede sem o uso de nó malicioso, dentro do simulador *Cooja*. Os dispositivos serão os nós da rede, sendo a comunicação feita através destes. Esta comunicação é estabelecida cada vez que um nó encontra-se dentro do alcance de outro nó e vice versa, os nós de cor amarela representam os sensores (nós filhos), os de cor verde representam o nó raiz (nó pai) e por fim o nó de cor roxa é o nó malicioso. A Figura 13, 14 e 15 ilustram as mesmas topologias, porém com o uso do nó malicioso entre os nós sensores.

No primeiro momento tem-se o nó malicioso que realiza o ataque conhecido como Hello-Flood, onde temos que um nó atacante envia mensagens de Hello, que para o RPL são os DIO, para vários outros nós, com valores altos de potência de modo a ser escolhido como a opção para o roteamento, como a maioria dos nós normais não possui uma potência de sinal igualmente forte, suas mensagens não serão transmitidas com sucesso. Ele gera uma grande quantidade de tráfego através de mensagens DIS, o que faz com que os nós dentro de seu alcance enviem as mensagens DIO e acabem redefinindo seus temporizadores (que aumenta conforme a rede se estabiliza) o que provavelmente aumenta seu consumo de energia, atrapalhando a função do RPL de baixo consumo de energia. Mesmo que seja usado uma mensagem DIS segura, este ataque pode ser executado ao usar um nó comprometido. Este é considerado um ataque direto. É esperado que não haja mudanças no DAG mas que o consumo de energia seja elevado.

Logo após tem-se o nó malicioso que realiza o ataque conhecido como *Versioning Attack* (Ataque de versão), que consiste em aumentar o número da versão (que é normalmente uma responsabilidade da raiz do DODAG's quando um reparo global for executado), levando a uma reconstrução de gráficos desnecessárias. Tem-se que a raiz recebe o DIO com um número de versão inválido, o mesmo irá atualizar e redefinir o seu temporizador de gotejamento, então irá reenviar uma nova mensagem DIO. Assim tem-se que os nós normais, irão iniciar um reparo global, ou seja, eles removem os nós pais e usam o DIO recebido para atualizar o seu novo vizinho.

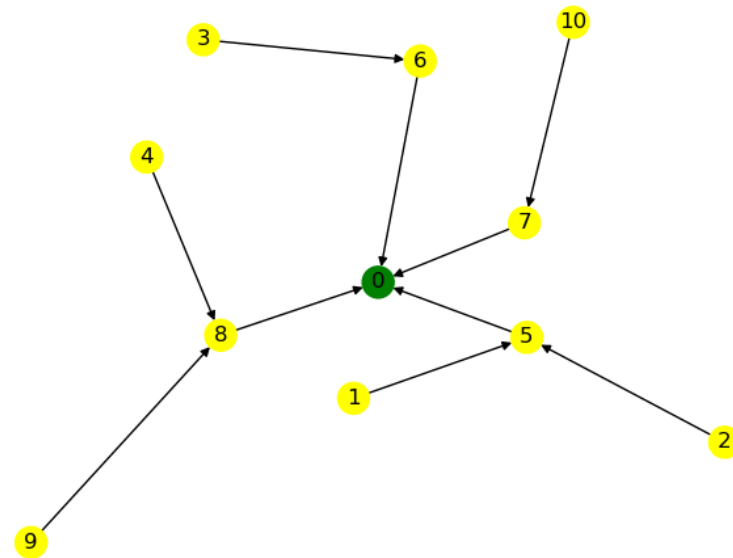
5.2 – Resultados dos testes

Para o teste de qualidade proposto, observa-se o *PowerTracker* ou seja os valores de Rx (recepção), Tx (transmissão) e On (tempo "ligado") de cada nó contido dentro da topologia de rede utilizada. Todas as medições foram feitas a nível de transporte e foram obtidas através do simulador *Cooja*.

O parâmetro de consumo de energia é um indicador relativo, ou seja, não pode ser convertido de forma precisa para um valor de potência absoluto. Usando o simulador *Cooja*, alteramos a conectividade entre os nós simulados na rede RPL. Assim, simulamos uma inundação HELLO, permitindo que um nó malicioso tenha a capacidade de enviar dados para todos os outros nós da rede. Porém, somente os nós que estão próximos ao atacante têm a capacidade de responder. Vimos nas Figura 13, que o nó malicioso interage com os nós 4, 1, 8. Na Figura 14 é possível ver que o nó malicioso se relaciona com os nós 9, 1, 11, 3. E por fim, na Figura 15 o nó malicioso se conecta com os nós 13, 2, 8, 14, 12, 11. A mensagem HELLO é a mensagem inicial que um nó envia ao entrar em uma rede, ao ser transmitida com um poder de sinal forte e com uma métrica de roteamento favorável, o invasor pode se apresentar como vizinho de muitos nós, possivelmente de uma rede inteira, no entanto em alguns nós vizinhos do invasor, ao tentar se comunicar com o mesmo podem ver suas mensagens perdidas pois o invasor pode estar fora do alcance de rádio. Nas Figuras 10, 11, 12 e 13, 14, 15 é possível observar a diferença do consumo de energia, com e sem o nó malicioso.

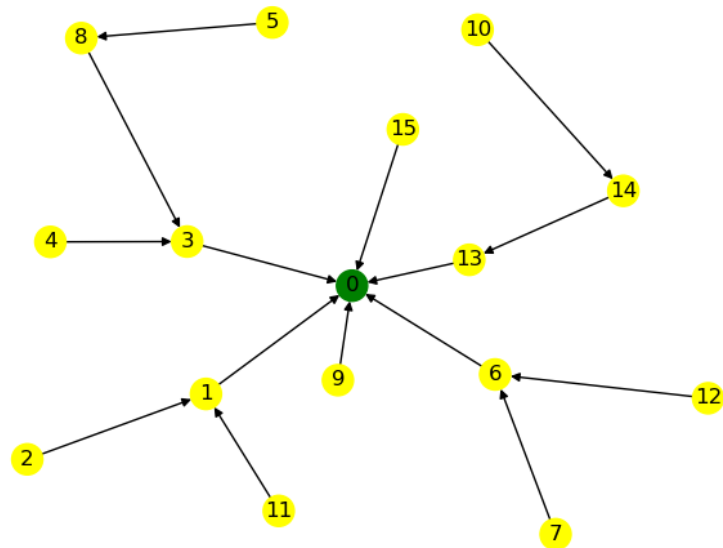
A seguir tem-se as imagens correspondentes as topologias que não possuem o nó malicioso, que posteriormente será feito o ataque Hello-Flood:

Figura 10 - Topologia com 10 nós sensores sem nó malicioso.



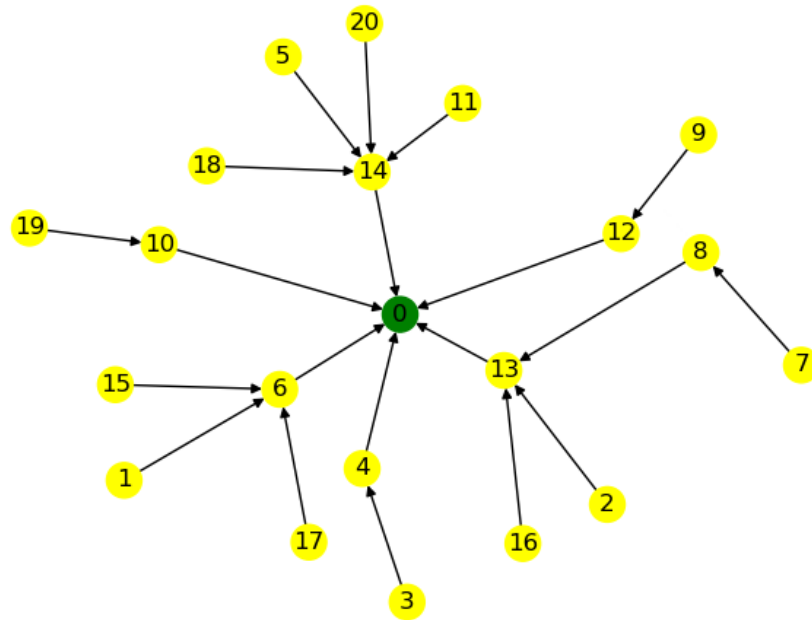
Fonte: Software *Contiki*.

Figura 11 - Topologia com 15 nós sensores sem nó malicioso.



Fonte: Software *Contiki*.

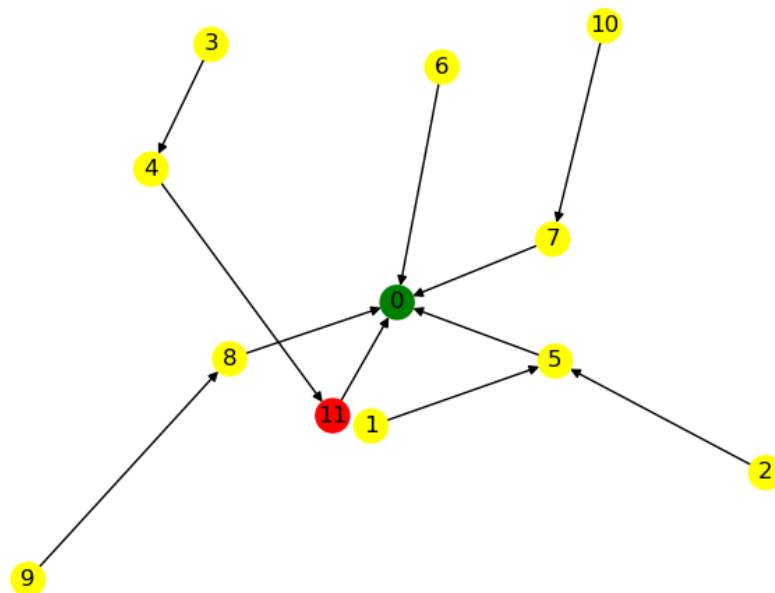
Figura 12 - Topologia com 20 nós sensores sem nó malicioso.



Fonte: Software *Contiki*.

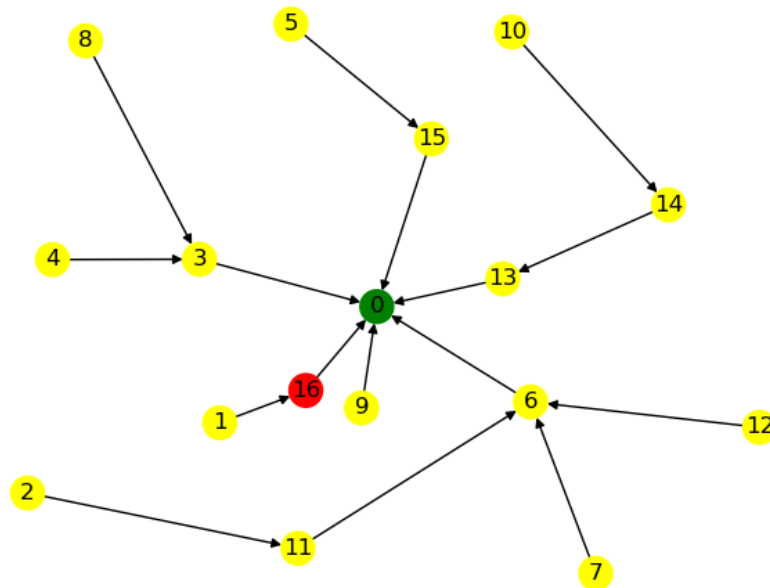
Logo abaixo temos as topologias com o nó malicioso fazendo parte da rede do ataque Hello-Flood:

Figura 13 - Topologia com 10 nós sensores com o nó malicioso.



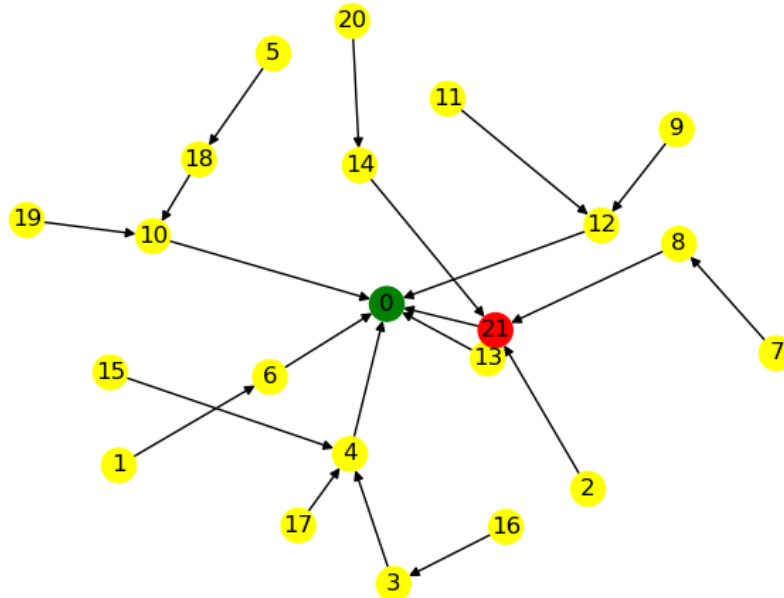
Fonte: Software *Contiki*.

Figura 14 - Topologia com 15 nós sensores com nó malicioso.



Fonte: Software *Contiki*.

Figura 15 - Topologia com 20 nós sensores com nó malicioso.



Fonte: Software *Contiki*.

Em relação à aplicação RPL, a mesma se demonstrou prática, eficiente e útil na captação de dados e comunicação entre os nós. Uma das principais vantagens do RPL neste projeto em específico é a sua flexibilidade que o permite fazer rotas de um-para-um, além de possibilitar que

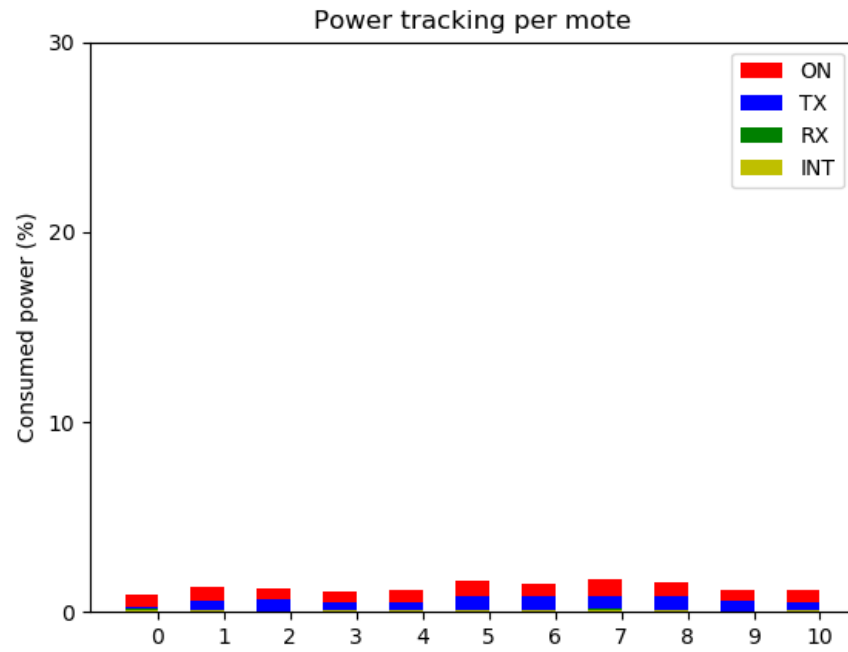
elementos de redes que possuem diferentes capacidades sejam empregados para a otimização de rotas. Também possui um modo de operação em que opera sobre o paradigma de muitos-para-muitos, o que faz dele um protocolo tão flexível que tange várias opções de operação.

Os testes feitos demonstram que com o ataque de Hello-Flood, as topologias que possuem o nó malicioso obtiveram um consumo de energia mais elevado em relação as que não possuíam o nó atacante. Observa-se nas Figuras 19, 20 e 21, onde os nós que estão dentro do alcance de rádio do nó malicioso e seus vizinhos mais próximos consomem muito mais energia que os nós que estão relativamente mais afastados 3, 6 e 10, que não estão sendo atingidos diretamente então consomem menos energia que os demais. Ao comparar as Figuras 16, 17 e 18 com as Figuras 19, 20 e 21 é possível notar a grande diferença do consumo de energia, visto que até mesmo os nós que não são atingidos substancialmente pelo nó malicioso nas topologias com a sua presença, apresentam consumo de energia ainda mais reduzida nos gráficos correspondentes as topologias sem o nó atacante. Isso se deve ao fato do nó malicioso manter os nós atacados sempre em estado ligado (ON), o que faz que os mesmos estejam em estado de alerta mesmo quando não estão recebendo (Rx) ou transmitindo (Tx) dados, afetando também os nós vizinhos deles mesmo que indiretamente.

No entanto, pode-se ver que esses nós atacados diretamente não são afetados no tempo de Tx. Isso acontece com a transmissão do DIS, pois após a recepção de um DIS os nós são capazes de redefinir seus temporizadores de gotejamento, mas não enviam de imediato a mensagem DIO, devido à natureza de multicast do DIS enviado.

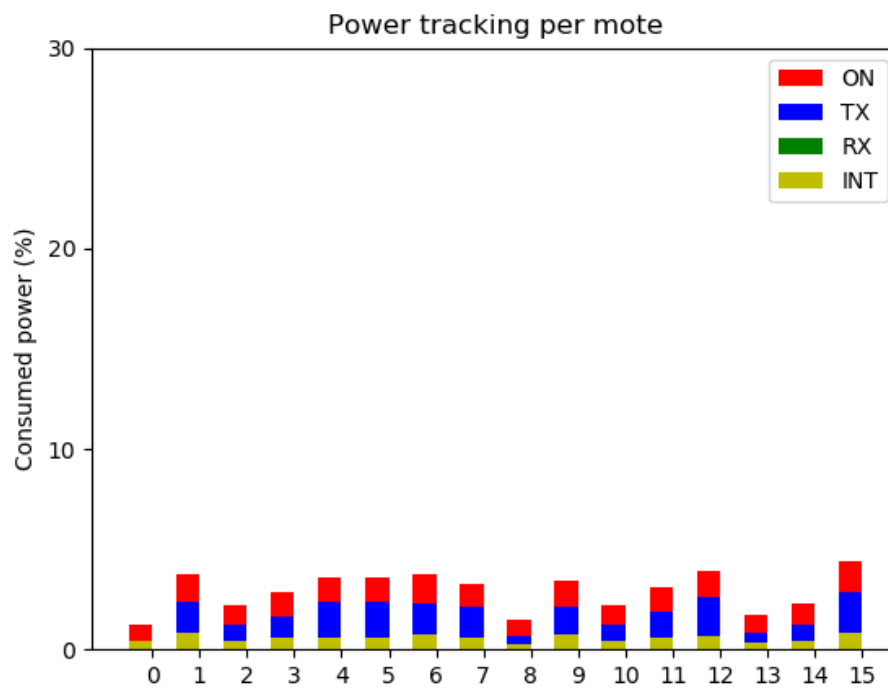
A seguir tem-se as Figuras relacionadas anteriormente e seus resultados gráficos.

Figura 16 - Consumo de energia por nó, topologia com 10 nós sensores sem nó malicioso.



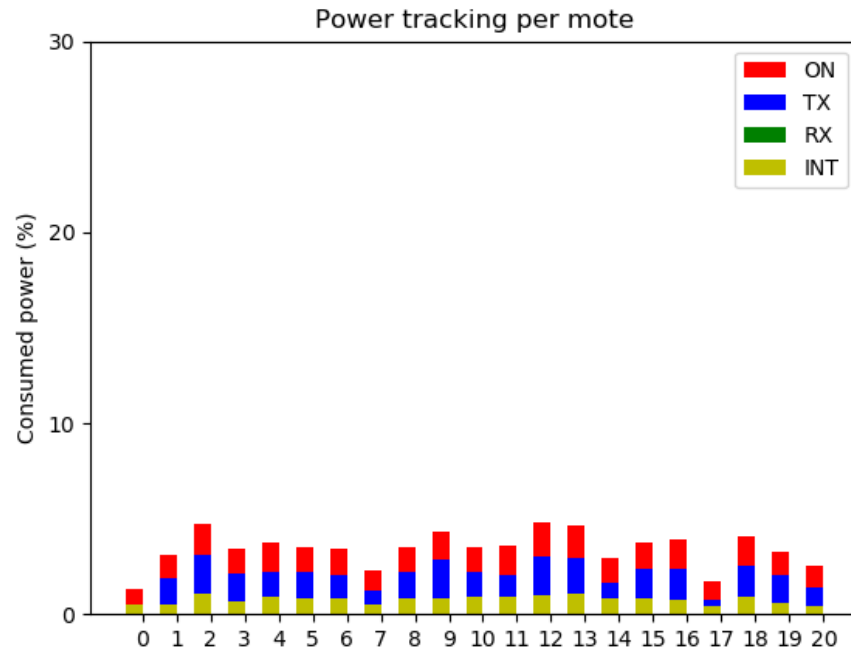
Fonte: Software *Contiki*.

Figura 17 - Consumo de energia por nó, topologia com 15 nós sensores sem nó malicioso.



Fonte: Software *Contiki*.

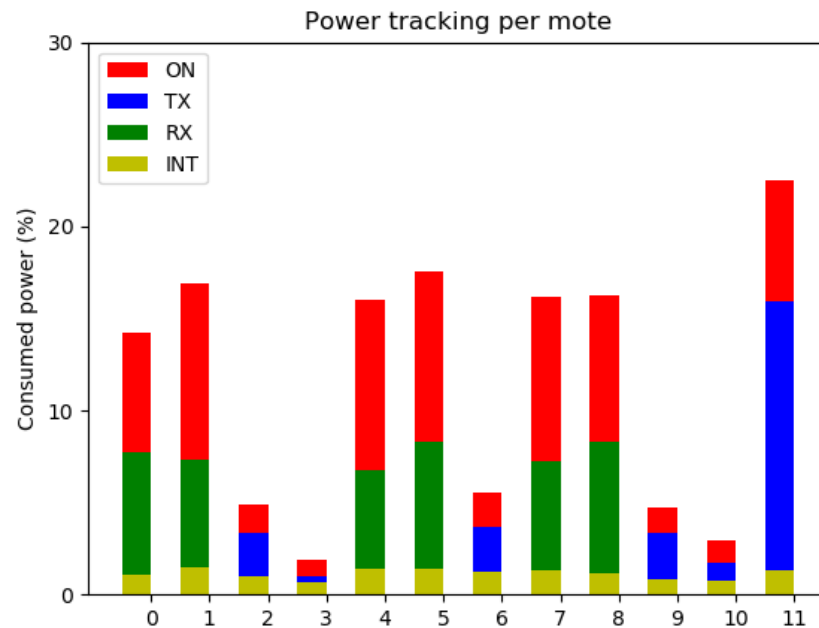
Figura 18 - Consumo de energia por nó, topologia com 20 nós sensores sem nó malicioso.



Fonte: Software *Contiki*.

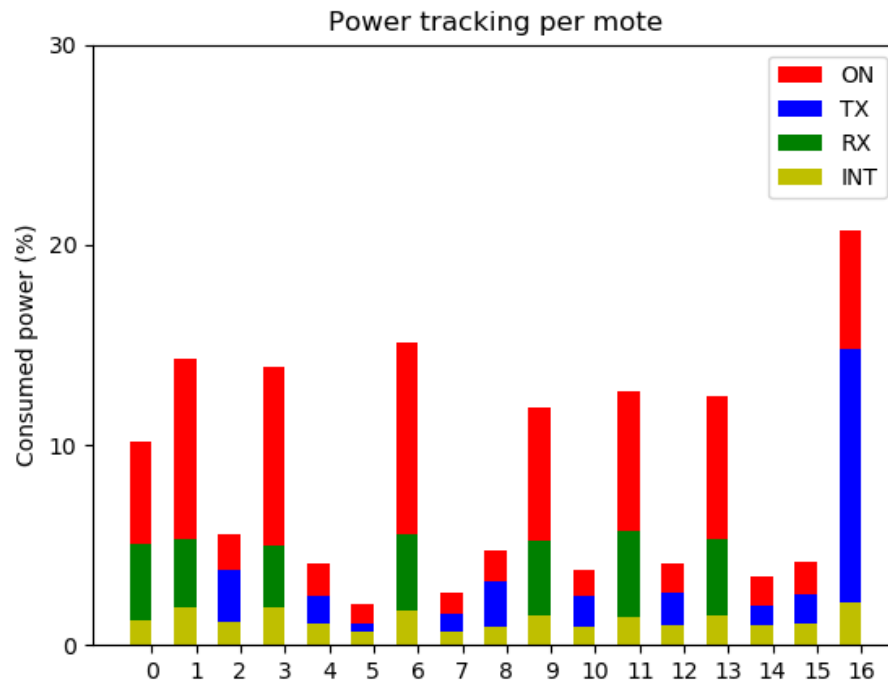
Logo em seguida tem-se os gráficos de consumo de energia gerados com a presença do nó malicioso:

Figura 19 - Consumo de energia por nó, topologia de 10 nós sensores com nó malicioso.



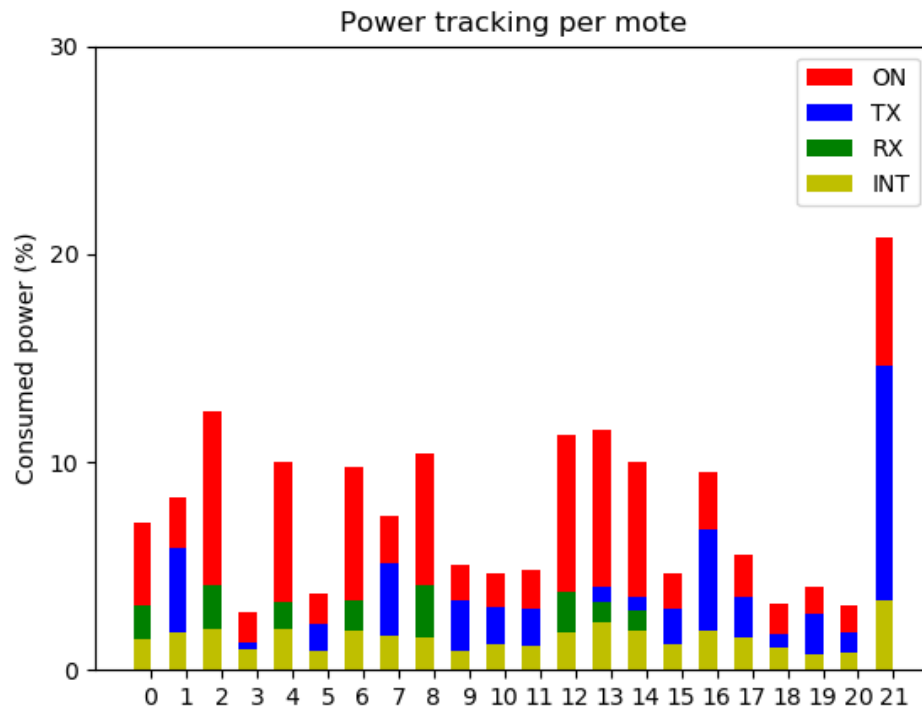
Fonte: Software *Contiki*.

Figura 20 - Consumo de energia por nó, topologia de 15 nós sensores com nó malicioso



Fonte: Software *Contiki*.

Figura 21 - Consumo de energia por nó, topologia de 20 nós sensores com nó malicioso.

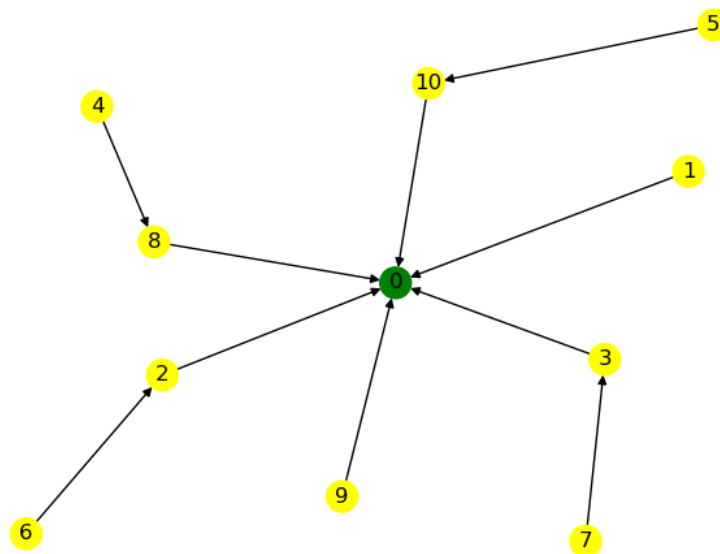


Fonte: Software *Contiki*.

O próximo ataque realizado é o Ataque de versão, simula-se um aumento de versão quando um reparo global deve ser executado, permitindo que o nó malicioso consiga fazer as alterações necessárias dentro da rede. Pode-se ver nas Figuras 26, 27 e 28 a topologia com o nó atacante, mostrando os nós e sua comunicação. Tem-se que o número de versão é geralmente usado pelo nó raiz, para fazer o controle de reparo global do RPL e garantir que todos os nós no DODAG estejam atualizados com o estado de roteamento. Toda mensagem DIO possui um número de versão para que, no caso dos nós vizinhos terem uma versão DODAG desatualizada, eles possam se juntar a um novo DODAG, recalculando sua classificação e então atualizando o seu número de versão. O nó malicioso faz com que os nós sensores façam esta atualização mais vezes que o necessário, o que cria um número a mais de gráficos que não são necessários.

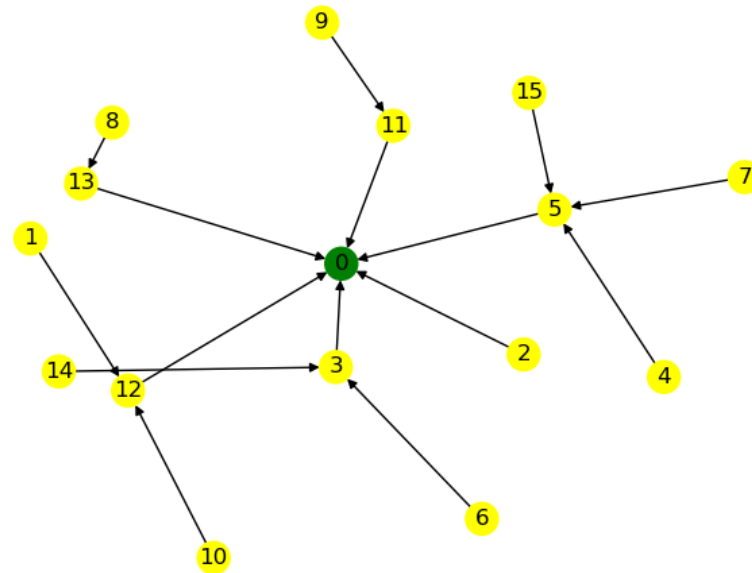
A seguir tem-se as topologias sem o nó malicioso em sua composição:

Figura 22 - Topologia com 10 nós sensores sem nó malicioso.



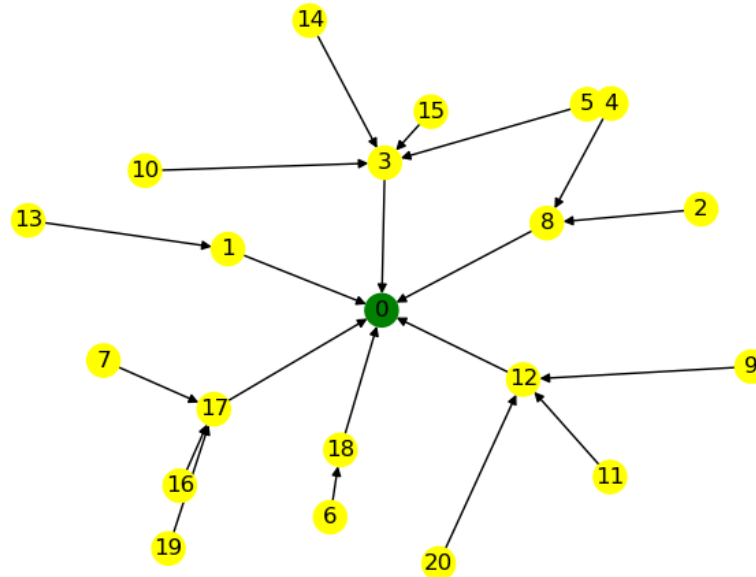
Fonte: Software *Contiki*.

Figura 23 - Topologia com 15 nós sensores sem nó malicioso.



Fonte: Software *Contiki*.

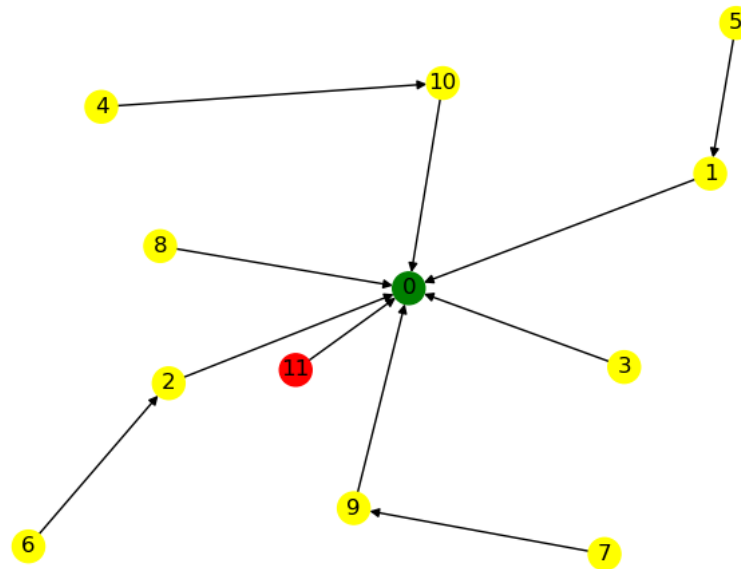
Figura 24 - Topologia com 20 nós sensores sem nó malicioso.



Fonte: Software *Contiki*.

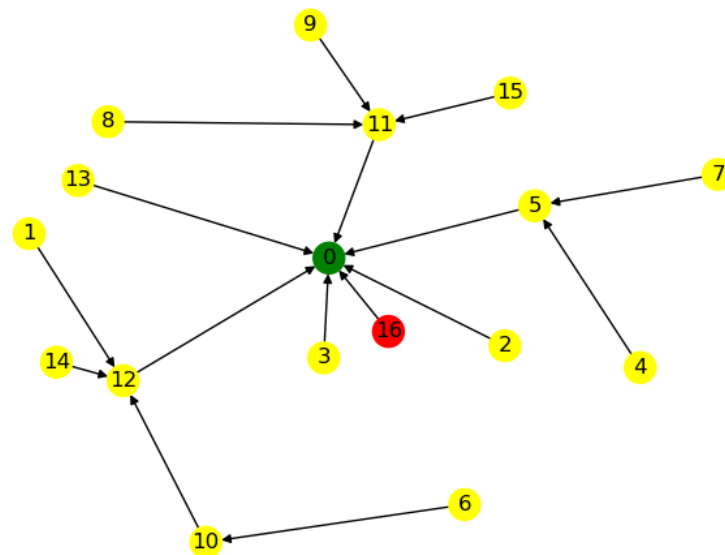
Logo abaixo pode-se ver as topologias com a presença do nó malicioso:

Figura 25 - Topologia de 10 nós sensores com nó malicioso.



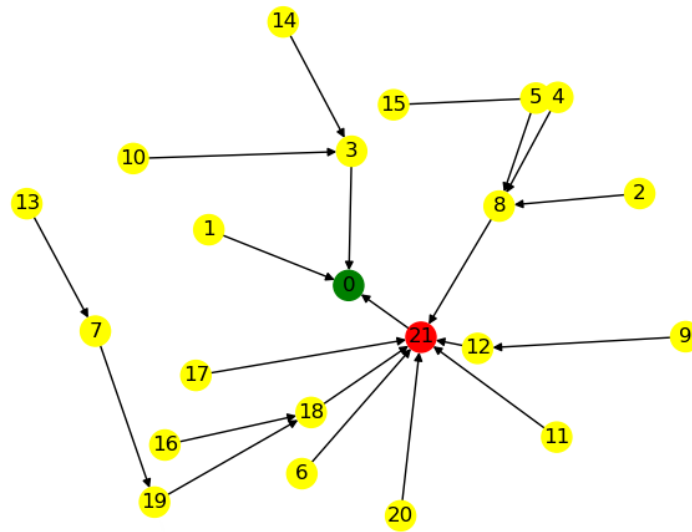
Fonte: Software *Contiki*.

Figura 26- Topologia de 15 nós sensores com nó malicioso.



Fonte: Software *Contiki*.

Figura 27 - Topologia de 20 nós sensores com nó malicioso.



Fonte: Software *Contiki*.

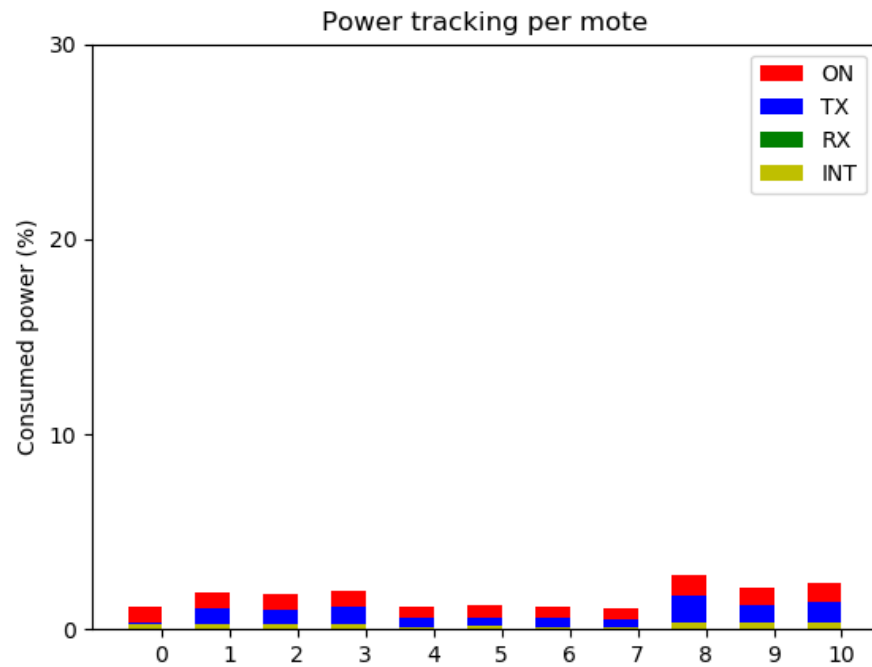
Os testes realizados com o Ataque de Versão, as topologias que possuem o nó malicioso apresentam consumo maior de energia por nó quando comparado as que não possuem o nó atacante em sua rede. A distância do nó malicioso com o nó raiz causa menos perdas de pacote.

Ao observar as Figuras 31, 32 e 33 é possível ver que esse ataque possui uma grande eficiência na rede inteira, já que ele aciona muitas mensagens ao fazer o reparo global. Logo pode-se falar que a integridade do campo de versão na mensagem DIO não é verificada mesmo no modo de operação seguro do RPL. Esse impacto causado pelo ataque do número de versão torna importante desenvolver mecanismos que possam ser usados para mitigá-lo. Mas como o RPL será comumente usado em redes de baixa potência, com dispositivos restritos, é necessário que as estratégias de mitigação de ataques sejam o mais simples possível. O aumento de consumo de energia se dá devido aos loops (pulos de um nó ao outro) que a mensagem dá até chegar ao nós raiz. Os valores de Rx são muito pequenos, por isso eles acabam não tendo visualização nos gráficos gerados por este ataque.

Apesar dos problemas enfrentados durante o percurso do trabalho, é notável a grande diferença no consumo de energia, o que mostra que a implementação do ataque foi realizado com sucesso. Pode-se dizer que a implementação em uma rede RPL se mostra útil para aplicações

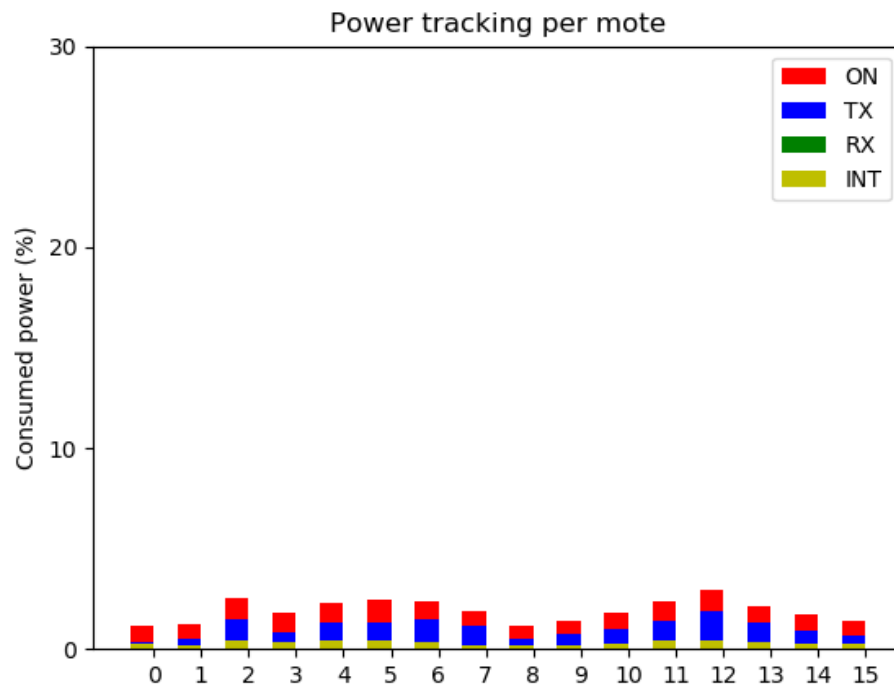
com baixas taxas de dados quando a mesma não esta sendo atacada. A seguir tem-se as Figuras relacionas anteriormente e seus resultados gráficos aqui já comentados.

Figura 28 - Consumo de energia por nó, topologia de 10 nós sensores sem nó malicioso



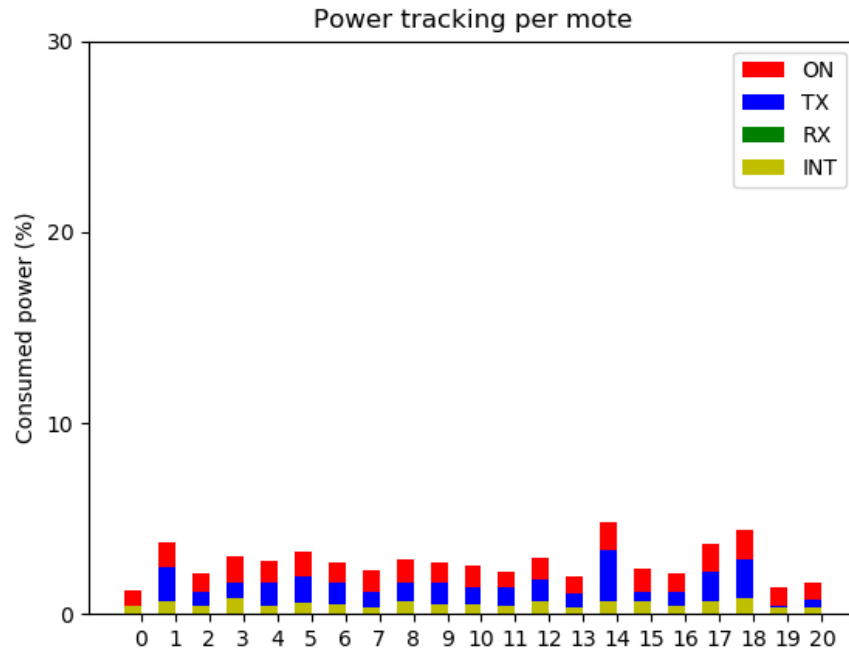
Fonte: Software *Contiki*.

Figura 29 - Consumo de energia por nó, topologia de 15 nós sensores sem nó malicioso



Fonte: Software *Contiki*.

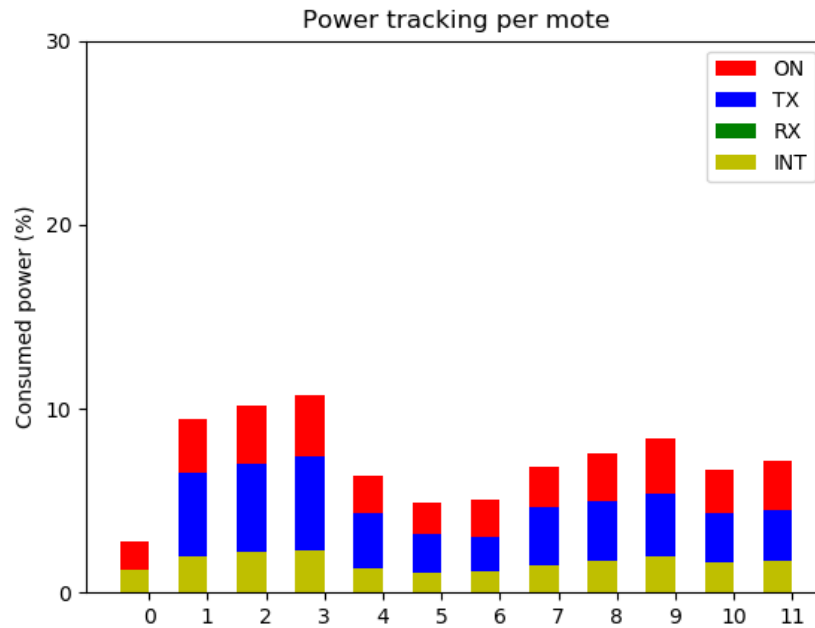
Figura 30 - Consumo de energia por nó, topologia de 20 nós sensores sem nó malicioso.



Fonte: Software *Contiki*.

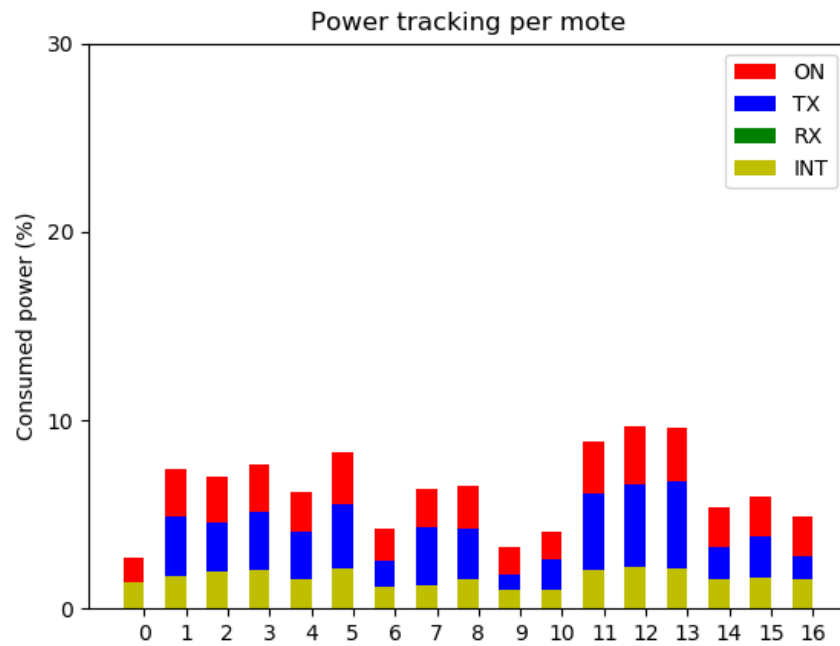
Tem-se logo abaixo os gráficos de consumo de energia com nó malicioso:

Figura 31 - Consumo de energia por nó, topologia de 10 nós sensores com nó malicioso.



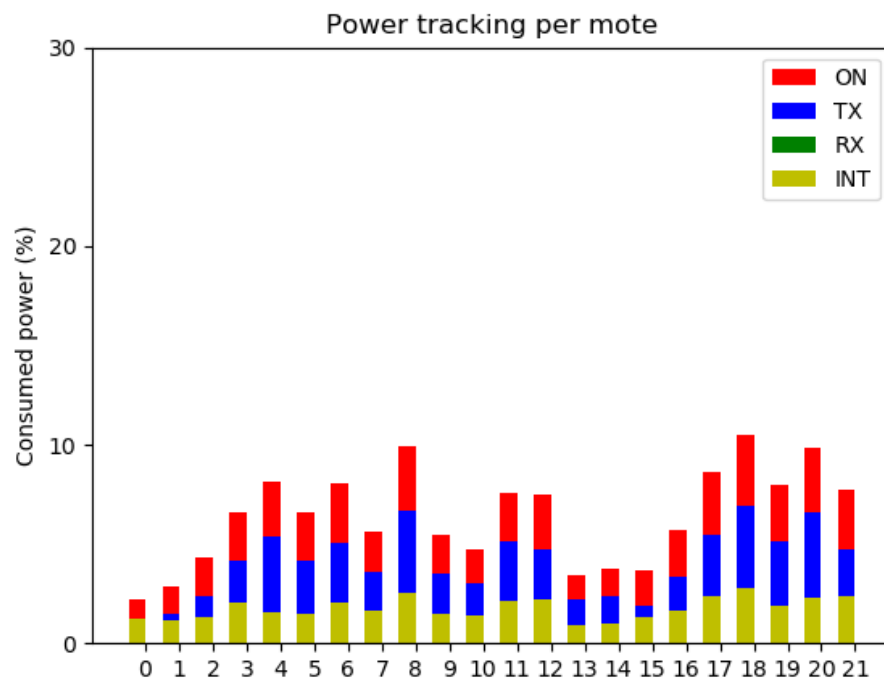
Fonte: Software *Contiki*.

Figura 32 - Consumo de energia por nó, topologia de 15 nós sensores com nó malicioso.



Fonte: Software *Contiki*.

Figura 33 - Consumo de energia por nó, topologia de 20 nós sensores com nó malicioso.



Fonte: Software *Contiki*.

CAPÍTULO 6

CONCLUSÃO E TRABALHOS FUTUROS

A internet das coisas é com certeza uma grande parte do futuro das redes de telecomunicações, inovando em várias áreas de atuação e criando novas aplicações. Para que a mesma possa ser implementada, ainda é necessário um grande estudo, implantação de tecnologias e avanços na área de segurança para que a IoT possa ser viabilizada. O trabalho tratou de alguns ataques que podem afetar de forma muito danosa as redes de IoT, mostrando que ainda há muito o que se pesquisar nessa área. Temos que o software *Contiki* aliado ao simulador *Cooja* possui uma grande variedade de aplicações na área de IoT, e possui plugins importantes para a observação dos eventos que ocorrem durante uma simulação.

Pode-se destacar também a privacidade para IoT, sendo este um dos principais pontos, e uma das barreiras que impedem a efetiva adoção da internet das coisas. Sabe-se que tanto as empresas quanto os usuários estão preocupados com a violação de seus dados e como serão usados. Logo podemos dizer que a segurança e privacidades desempenham um papel chave na implementação da IoT.

Como proposta para trabalhos futuros, tem-se a implementação de simulações que visem a mitigação dos ataques feitos. Um estudo sobre métodos usados para diminuir e até mesmo extinguir os efeitos causados por ataques nas redes de IoT, juntamente com simulações que comprovem a eficácia dos códigos implementados para a mitigação de cada ataque.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] WEBER, ROLF H. Internet of things: Need for a new legal environment? 2009, Vol. v.25, p.522-527.
- [2] GUBBI, JAYAVARDHANA, ET AL. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*. 29, 2013, pp. 1645-1660.
- [3] SINGER, T. Tudo conectado: Conceitos e representações da internet das coisas. . *SimSocial*. 2012.
- [4] MIORANDI, D, ET AL. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*. v.10, 2012, Vols. p.1497–1516.
- [5] SONAR, K., & UPADHYAY, H. A survey: DDOS attack on Internet of Things. *International Journal of Engineering Research and Development*, (2014),10(11), 58-63.
- [6] ITU, INTERNET REPORTS. *ITU Internet Reposts 2005: The Internet of Things*. s.l. : International Telecommunications Union, 2005.
- [7] ZHANG, C; GREEN, R. Communication security in internet of things: Preventive measure and avoid ddos attack over IoT network. *Procedding of the 18th Symposium on Communications & Networking*, 2015.
- [8] RAYES, A; SALAM, S. Internet of Things - From hype to reality. Chan, Switzerland: Springer International Publishing, 2017.
- [9] JARA, A; LADID, L; SKARMETA, A. The Internet of Things through IPv6: An Analysis of Challenges, Solutions and Opportunities. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 2014.
- [10] ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things:. *a survey, Comput Netw*. 2010, Vol. v.54, p.2787-2805.
- [11] KEOH, SYE LOONG, KUMAR, SANDEEP S. AND TSCHOFENIG, HANNES. Securing the Internet of Things: A Standardization Perspective. *IEEE Internt of Things Journal*. 2014, Vol. v.1, NO.3.
- [12] LEO, MARCO, ET AL. A federated architecture approach for Internet of Things security. p.1-5, 2014.

- [13] TEIXEIRA, A. F., ET AL. Defending Internet of Things against exploits. *IEEE Latin America Transactions*. 2015, Vol. 13, p. 1112-1119.
- [14] SICARI, S., RIZZARDI, A., GRIECO, L., AND COEN-PORISINI, A. Security privacy and trust in internet of things: The road ahead. *Comp.Networks*. 2015, Vol. 76, p.146-164.
- [15] CONTIKI-OS. Contiki. The Open Source OS for the Internet of Things. Disponível em: <
<http://www.contikios.org>
- [16] BRUNO, L., Security in Lossy and Low-power Networks- tools for 6LoWPAN, 2010. Monografia (Engenharia em Informática), Istituto Superiore Mario Boella, Torino, Itália
- [17] DE CASTRO, H. A., DA CUNHA, M. J., CARRIJO, R. S., Proposta de aplicações integrando o sistema operacional Contiki com a plataforma de testes unitários Cunit, 2016.
- [18] IOANNIS, A, CHRYSOSTOMOS, C AND HADJICHRISTOFI, G. Internet of Things:. *Security vulnerabilities and challenges*. p.180 - 187, 2015.
- [19] DALFOVO, O AND HOSTINS, C, A. *Delineamento para aplicação do RFID na logística de supermercado como inteligência competitiva: supermercado Hostins*. p.23-48, Blumenau : Revista Interdisciplinar Científica Aplicada, 2010, Vol. v.4.
- [20] BRADLEY, J, BARBIER, J AND HANDLER, D. *Embracing the Internet of Everything to capture your share of \$14.4 Trillion: More relevante, Valuable Connections will improve innovation, productivity, efficiency and customer experience*. s.l. : Cisco Systems Inc., 2013. CISCO Whitepaper.
- [21] AL-FUQAHA, A.; GUIZANI, M.; MOHAMMADI, M.; ALEDHARI, M.; AYYASH, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE COMMUNICATION SURVEYS & TUTORIALS*, v. 17, 2015.
- [22] ALI, A.; SHAH, G. A.; FAROOQ, M. O.; GHANI, U. Technologies and challenges in developing Machine-to-Machine applications: A survey. *Journal of Network and Computer Applications*, 2017.
- [23] ATZORI, L; IERA, A; MORABITO, G. The Internet of Things: A survey, 2010. *Computer Networks* 54 (2010), p. 2787–2805.

- [24] REIS, D, O. *Estudo de uma plataforma aberta para comunicações M2M*, s.l. : Tese de Mestrado Integrado. Engenharia Informática e Computação. Faculdade de Engenharia. Universidade do Port, 2013.
- [25] JUNIOR, A, J AND MORENO, E, D. *Segurança em Infraestrutura para Internet das Coisas*. p. 370-380, s.l. : Revista Gestão.Org, 2015, Vol. v.13 n.3.
- [26] CHAQFEH, M AND MOHAMED, N. *Challenges in middleware solutions for the internet of things*. s.l. : Collaboration Technologies and Systems (CTS), 2012. International Conference on IEEE. Vols. p. 21-26.
- [27] DWIVEDI, A. K. AND VYAS, O. P. *Wireless Sensor Network: At a Glance*. s.l. : INTECH Open Access Publisher, 2011.
- [28] KUMARI, P, KUMAR, M AND RISHI, R. *Study of Security in Wirelees Sensor Networks*. p. 347-354, s.l. : Proceendings of International Journal of Computer Science and Technology, 2010, Vol. v. 1 n. 5.
- [29] FERREIRA, H, G, C. *Arquitetura de Middleware para Internet das Coisas*. 2014, Vol. p.125.
- [30] UKIL, A, SEN, J AND KOILAKONDA, S. *Embedded security for Intentert of Things*. s.l. : National Conference IEEE, 2011. Emerging Trends and Applications in Computer Science (NCETACS). Vols. p. 1-6.
- [31] CASADO, L AND TSIGAS, P. *Contikisec: A secure network layer for wireless sensor networks under the contiki operating system*. 2009. In: Identife and Privacy in the Internet Age. Vols. p. 133-147.
- [32] DUNKELS, ADAM. *A big step for Contiki: built-in encryption*. 2014.
- [33] LEITÃO, L., SAMPAIO, A. AND HOLANDA FILHO, R. *NaturalCloud: Um framework para integração de Rede de Sensores sem Fio na Nuvem*. Ouro Preto : s.n., 2012. X Workshop em Clouds, Grids e Aplicações WCGA. Vols. p. 44-45.
- [34] RUSCHEL, H, ZANOTTO, M AND MOTA, W. *Especialização em Redes e Segurança de Sistemas* . Curitiba : Pontifícia Universidade Católica do Paraná., 2010.

- [35] MELL, P. AND GRANCE, T. *The NIST definition of cloud computing*. s.l. : NIST-National Institute of Standards and Technology, 2009.
- [36] ALMORSY, M, GRUNDY, J AND IBRHAHIM, A, S. *Collaboration-based cloud computing security management framework*. . s.l. : IEEE, 2011. Cloud Computing (CLOUD) IEEE International Conference on. Vols. p. 364-371.
- [37] NIST. *Standards for Security Categorization of Federal Information and Information Systems*. s.l. : Acesso: 11 de Jun. 2018., 2016.
- [38] SHARMA, K. AND GHOSE, M. K. *Complete Security Framework for Wireless Sensor Networks*. 2009.
- [39] ISTEPANIAN, R. S H, "The potential of Internet of Things (IoT) for assisted living applications," Assisted Living 2011, IET Seminar on , pp.1,40, 6-6 April 2011.
- [40] MORALES, E. B. "6LoWPAN Internet das coisas", 2013.
- [41] KHAN, R.; KHAN, S.U.; ZAHEER, R.; KHAN, S., "*Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges*," Frontiers of Information Technology (FIT), 2012 10th International Conference on , vol., no., pp.257,260, 17-19, 2012.
- [42] CHAVAN, A. A. , MININATH K. N., "*Secure and Cost-effective Application Layer Protocol with Authentication Interoperability for IOT* " International Conference on Information Security & Privacy (ICISP2015), 11-12, 2015.
- [43] BORMANN, C.; CASTELLANI, A. P.; SHELBY, Z. CoAP: An Application Protocol for Billions of Tiny Internet Nodes. IEEE Internet Computing Março de 2012.
- [44] SHELBY, Z.; HARTKE, K.; BORMANN, C. The Constrained Application Protocol (CoAP). Junho de 2014. Disponível em: <<https://tools.ietf.org/html/rfc7252>>. Acesso em: 10 de Junho de 2018.
- [45] YASSEIN, M. B.; SHATNAWI, M. Q.; AL-ZOUBI, D. Application Layer Protocols for the Internet of Things: A survey. *In Engineering & MIS (ICEMIS)*, Setembro de 2016.
- [46] THANGAVEL, D.; XIAOPING M.; VALERA, A.; HWEE-XIAN, T.; COLIN, K.-Y. T. Performance Evaluation of MQTT and CoAP via a Common Middleware. IEEE Ninth

International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) Symposium on Sensor Networks, 2014.

- [47] SUBRAMONI, H. ; MARSH , G. M. ; NARRAVULA, S.; PING LAI, ; PANDA, D. K. Design and evaluation of benchmarks for financial applications using Advanced Message Queuing Protocol (AMQP) over InfiniBand. *Published in Workshop on High Performance Computational Finance*, 2008.
- [48] SAINT-ANDRE, P. Extensible Messaging and Presence Protocol (XMPP): Core. Março de 2011. Disponível em: <<https://datatracker.ietf.org/doc/rfc6120/>>. Acesso em: 10 de Junho de 2018.
- [49] SANTANA, E. M. Desenvolvimento de uma plataforma de comunicação 6lowPAN para redes de comunicação wireless de curto alcance. *Faculdade Federal de Uberlândia*. 2017.
- [50] MAHMOOD, A.; JAVAID, N.; SOHAILRAZZAQ. A review of wireless communications for smartgrid. *Renewable and Sustainable Energy Reviews*, Setembro de 2014.
- [51] SAPUTRO, N.; AKKAYA, K.; ULUDAG, S. A survey of routing protocols for smart grid communications. *Computer Networks*, Abril de 2012.
- [52] THREAD (2015). Whitepaper, Thread Stack Fundamentals. Thread Group.
- [53] ROTTA, G.; CHARÃO, A.; DANTAS, D. Um Estudo sobre Protocolos de Comunicação para Ambientes de Internet das Coisas. Universidade Federal de Minas Gerais. *Fórum de iniciação Científica*, 2017.
- [54] GADDOUR, O.; KOUBÂA, A. RPL in a nutshell: A survey. *Computer Networks*, Julho de 2012.
- [55] HUI, J. W.; CULLER, D. E. IP is Dead, Long Live IP for Wireless Sensor Networks. *SenSys '08*, Novembro de 2008.
- [56] WU, P.; CUI, Y.; WU, J.; LIU, J.; METZ, C. Transition from IPv4 to IPv6: A State-of-the-Art Survey. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, 2013.
- [57] PONGLE, P. ; CHAVAN, G. Real Time Intrusion and Wormhole Attack Detection in Internet of Things. *International Journal of Computer Applications* (0975-8887)Volume 121 - No. 9, July 2015

- [58] BAHIA, J. G., CAMPISTA, M. E. M., Um Mecanismo de Controle de Demanda no Provimento de Serviços de IoT Usando CoAP, 2017.
- [59] WITTENBURG, G.; SCHILLER, J. Running real-world software on simulated wireless sensor nodes. In Proceedings of the ACM Workshop on Real-World Wireless Sensor Networks-REALWSN 06, pages 7–11, 2006.
- [60] LEVIS, P.; LEE, N.; WELSH, M.; CULLER, D. Tossim: Accurate and scalable simulation of entire tinyos applications. In Proceedings of the 1st international conference on Embedded networked sensor systems, pages 126–137. ACM, 2003.
- [61] BEN L. TITZER, DANIEL K. LEE, JENS PALSBERG. Avrora: Scalable sensor network simulation with precise timing. In Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press.
- [62] VASILOMANOLAKIS, E. ; DAUBERT, J. ; LUTHRA, M. ; GAZIS, V. ; WIESMAIER, A.; KIKIRAS, P. On the Security and Privacy of Internet of Things Architectures and Systems. International Workshop on Secure Internet of Things, 2015.
- [63] KALITA, H. K; KAR, A. —Wireless sensor network security analysis, International Journal of Next-Generation Networks, vol. 1, no. 1, pp. 87-115, 2009.
- [64] FIGUEIRA, V, P. “Internet das Coisas” –Um Estudo sobre Questões de Segurança, Privacidade e Infraestrutura. *Universidade Federal Fluminense*, 2016.