



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA

Júlio Gusmão Carlos de Mendonça

Viabilizando Técnicas de Gerenciamento Baseadas em Fluxos para LoRaWAN

João Pessoa
2022

Júlio Gusmão Carlos de Mendonça

Viabilizando Técnicas de Gerenciamento Baseadas em Fluxos para LoRaWAN

Monografia apresentada ao curso de Engenharia de Computação do Centro de Informática da Universidade Federal da Paraíba, como requisito para a obtenção do grau de Bacharel em Engenharia de Computação

Orientador: Dr. Ewerton Monteiro Salvador

João Pessoa
Janeiro de 2022

Catálogo na publicação
Seção de Catalogação e Classificação

M539v Mendonça, Julio Gusmão Carlos de.

Viabilizando técnicas de gerenciamento baseadas em fluxos para LoRaWAN / Julio Gusmão Carlos de Mendonça. - João Pessoa, 2021.

50 f.

Orientação: Ewerton Monteiro Salvador.
Monografia (Graduação) - UFPB/CI.

1. Redes de longa distância. 2. Gerenciamento de redes. 3. Internet das coisas. 4. LoRa. 5. LoRaWAN. 6. Netflow. I. Salvador, Ewerton Monteiro. II. Título.

UFPB/CI

CDU 004.735

AGRADECIMENTOS

Antes de tudo, eu agradeço a Deus pela minha existência, pela proteção e conforto em tempos de dificuldade. Aos meus pais e minha irmã, por todo o amor, cuidado e educação, sendo inspiração no meu dia a dia e a fundação que permitiu a segurança na busca por conhecimento que veio a contribuir na minha formação. À minhas tias-avós e toda a família que acreditou e me apoiou ao longo da jornada. À minha namorada, por ter uma paciência enorme, mas especialmente pelo companheirismo em todos os momentos. Aos meus amigos, pela compreensão nas ausências e por continuarem fazendo questão da presença.

Ao meu orientador, sempre presente com seus direcionamentos e fazendo de tudo para que eu conseguisse finalizar a graduação. Sua contribuição e atenção para ajudar com as minhas dificuldades, sempre solícito às necessidades que eu apresentei e insistindo para que, no final, tudo desse certo, me salvando de inúmeras dores de cabeça quanto aos processos finais na universidade. À American Tower do Brasil, pelo acesso à rede LoRaWAN através do projeto IoT Labs firmado com a universidade. À empresa NLT Telecom, pela disponibilização do conjunto de dados utilizado nos testes. Ao aluno de Iniciação Científica Mateus Freitas Correia, pelo apoio na realização dos testes com os dados fornecidos pela empresas.

“O medo é o assassino da mente.”
(Frank Herbert)

RESUMO

Redes sem fio têm sido amplamente empregadas para diversas aplicações IoT (Internet-of-Things), como cidades inteligentes. Também, LPWAN (Low Power Wide Area Networking) é um paradigma de rede IoT que vem ganhando espaço no mercado e que possui características de grande importância para esse tipo de aplicação, sendo elas o baixo custo, implantação em larga escala e alta eficiência energética. De todas as tecnologias LPWAN à disposição, a LoRa vem se destacando, visto que traz um padrão que permite a construção de uma rede LPWAN sem necessitar de uma infraestrutura terceirizada. Ainda, o monitoramento de fluxos tem se tornado o método padrão para monitoramento de tráfego em redes TCP/IP, sem que exista uma alternativa equivalente para o uso no contexto LoRaWAN. Dessa forma, esse trabalho apresenta uma abordagem adaptada da definição de fluxos e o processo de modelagem de um sistema para o monitoramento desse tipo de rede. A solução foi validada com dados de tráfego de uma rede LPWAN real, e os resultados demonstram a eficácia da abordagem proposta para o gerenciamento de redes LoRaWAN. Assim, poder utilizar o conceito de análise de fluxos numa LoRaWAN é significativo para o aprimoramento do gerenciamento desse tipo de rede.

Palavras-chave: Gerenciamento de Redes, Internet das Coisas, LoRa, LoRaWAN, Net-Flow.

ABSTRACT

Wireless networks have been widely used for a vast number of IoT (Internet-of-Things) applications such as smart cities. Also, LPWAN (Low Power Wide Area Networking) is a paradigm of IoT network that has been gaining space in the market and that has characteristics of great importance for this type of application, namely the low cost, implementation in large scale and high energy efficiency. Of all the LPWAN technologies available, LoRa has been standing out as it brings a standard that allows the construction of a LPWAN network without the need for an outsourced infrastructure. Also, flow monitoring has become the standard method for monitoring traffic in TCP/IP networks, without an equivalent alternative for use in the LoRaWAN context. Thus, this work presents an adapted approach to the definition of flows and the modeling process of a system for monitoring this type of network. The solution was validated with traffic data from a real LPWAN network, and the results demonstrate the effectiveness of the proposed approach for managing LoRaWAN networks. Finally, being able to use the concept of flow analysis in a LoRaWAN is significant for improving the management of this type of network.

Key-words: Internet of Things, LoRa, LoRaWAN, NetFlow, Network Management.

LISTA DE FIGURAS

1	Quadro de LoRa.	23
2	Cabeçalho do fluxo Netflow.	24
3	Demonstração de mensagem JSON.	29
4	Estrutura do fluxo.	31
5	Modelagem tabela fluxos do banco de dados.	34
6	Dashboard Netflow. Fonte: [83].	34
7	Fluxograma de execução.	35
8	Módulo LoRaONE . Fonte: [84].	36
9	Proporção da quantidade de mensagens em um mesmo fluxo.	41
10	Número de mensagens nos fluxos de 8192b10f.	42
11	Proporção entre os tipos de mensagens.	43
12	Dispositivos que mandaram mais mensagens de uma vez.	43
13	Porcentagem de fluxos envolvendo o gateway 647a5ddda575f0bb.	44
14	Comunicação do dispositivo 4971d4c2ec2df8ba.	45
15	Conceito de <i>dashboard</i> com dados apresentados.	47

LISTA DE TABELAS

1	RSSI mínimo para decodificação com diferentes SFs. Fonte: [40].	20
---	---	----

LISTA DE ABREVIATURAS

API - Application Programming Interface

CSS - Chirp Spread Spectrum

EUI - Extended Unique Identifier

ED - End Device

FSK - Frequency Shifting Keying

FTDI - Future Technology Devices International

GPIO - General-Purpose Input/Output

IA - Internet Accounting

IETF - Internet Engineering Task Force

IoT - Internet of Things

IP - Internet Protocol

IPFIX - IP Flow Information eXport

ISM - Industrial Sientific and Medical

JSON - JavaScript Object Notation

LoRa - Long Range

LoRaWAN - Long Range Wide Area Networking

LPWAN - Low Power Wide Area Networking

MAC - Medium Access Control

MIC - Message Integrity Checks

NSSE - Network Synchronization and Scheduling Entity

OTAA - Over-The-Air Activation

RSSI - Received Signal Strenght Indicator

RTFM - Realtime Traffic Flow Measurement

SF - Spreading Factor

TDMA - Time-Division Multiple Access

UART - Universal Asynchronous Receiver-Transmitter

URL - Uniform Resource Locator

WG - Working Group

Sumário

1	INTRODUÇÃO	15
1.1	Objetivos	17
1.2	Estrutura da monografia	18
2	CONCEITOS GERAIS E TRABALHOS RELACIONADOS	19
2.1	Redes LoRa	19
2.2	Panorama da LoRaWAN	20
2.3	Fluxo em redes	23
2.4	Trabalhos Relacionados	25
2.5	Considerações finais	26
3	METODOLOGIA	27
3.1	Planejamento e Escolha de Ferramentas	27
3.2	Comunicação e Definição dos Fluxos	29
4	FLUXO LoRaWAN	32
4.1	Criação de Fluxos	32
4.2	Controle de Tempo	32
4.3	Cache	33
4.4	Banco de Dados	33
4.5	Módulo LoRaONE	35
4.6	Everynet e testes	37
5	RESULTADOS E DISCUSSÃO	40
5.1	Caracterização dos dados e configurações	40
5.2	Análise dos dados	40
6	CONSIDERAÇÕES FINAIS	46
	REFERÊNCIAS	47

1 INTRODUÇÃO

Aplicações da Internet das Coisas (Internet of Things – IoT), como em casas e cidades inteligentes, tornaram-se mais pervasivas, o que resultou no constante aumento da densidade e escala da implantação de sensores conectados através da Internet [1, 2, 3]. As previsões atuais indicam que 25 bilhões de dispositivos existirão em 2021 [4], o número de aparelhos IoT conectados pelo 5G atingirá 4,1 bilhões em 2024 [5], e é esperado que 500 bilhões de dispositivos IoT estarão conectados à Internet em 2030 [6]. As aplicações IoT empregam sensores que poderão monitorar o ambiente, comunicar-se com outros dispositivos e pessoas e realizar decisões inteligentes. Para conectá-los, redes sem fio são necessárias para promover operações robustas e maior cobertura levando em conta uma maior eficácia energética [7]. Tendo em vista a necessidade de uma rede IoT de maior densidade, novos desafios apresentam-se, buscando uma reavaliação de toda a conectividade global [8, 9] para que seja possível lidar com modelos de planejamento em tempo real.

Em sua maioria, esses dispositivos funcionam através de bateria. Espera-se que eles possam operar por um tempo longo de cinco a dez anos sem precisar de manutenção [10]. Esses dispositivos IoT finais também devem cobrir uma área geográfica abrangente. Eles comunicam-se através de pequenas porções de dados que transmitem informações importantes, como o nível de umidade, temperatura e outras variáveis, através de longas distâncias e por meio de múltiplos saltos entre esses aparelhos espalhados.

A necessidade de tais dispositivos operarem da maneira supracitada deu origem a um novo campo da conectividade IoT chamado Low Power Wide Area Networking (LPWAN), uma rede de dispositivos que se conectam através de longas distâncias com baixo consumo energético. Essa busca surgiu porque as redes IoT convencionais operavam sob tecnologias que cobriam apenas áreas menores como, por exemplo, o Bluetooth [11, 12, 13].

LPWAN é parte importante da tecnologia IoT [14]. Diversas soluções existentes nesse ramo, como SigFox [15] e LoRa (Long-Range) [16] estão sendo amplamente promovidas e aderidas. Dois dos principais objetivos presentes na concepção dessa tecnologia são o de minimizar o custo energético e, também, o total de um sistema, incluindo implantação, operação, manutenção, etc.

Diversos analistas e administradores de redes estão desenvolvendo diferentes ferramentas para lidar com problemas existentes no gerenciamento de vários aspectos dos dispositivos IoT, como a segurança da sua conectividade [33, 34]. Essa é apenas uma, entre outras necessidades, que poderiam ser administradas pelas técnicas existentes de gerenciamento de redes TCP/IP através da abordagem de fluxos [35]. Partindo desse princípio, se construída uma definição do que seria um fluxo em uma LoRaWAN, seria

possível, então, aplicar as técnicas existentes de análise e gerenciamento de fluxos nesse tipo de rede.

Desse modo, tradicionalmente, um fluxo é definido como um conjunto de pacotes IP passando por um ponto de observação na rede durante um certo intervalo de tempo, tal que todos os pacotes pertencentes a um fluxo em particular tenham um determinado conjunto de propriedades em comum [35]. Propriedades essas que podem incluir campos do cabeçalho como endereço IP da fonte e destino e os números de portas, conteúdos de pacotes, metadados e outros. Wireshark, OpenWRT e NetFlow são algumas das mais populares ferramentas de análise de rede que auxiliam na captura do tráfego entre diferentes dispositivos através de pontos de acesso sem fio.

Nesse sentido, técnicas para o isolamento e configuração de dispositivos IoT são propostas para preservar essas redes contra ataques [36] e, também, algumas estratégias baseadas em protocolos de identificação de porta são empregadas para detectar qualquer atividade maliciosa na rede [37, 38]. A Cisco agiu como divisora de águas no campo da análise de tráfego de rede com a técnica para caracterização de um fluxo para aplicações baseadas no protocolo IP [39]. A ferramenta NetFlow, pertencente à Cisco, traz diversos serviços de monitoramento e análise de fluxos de rede que lidam com diversas questões, incluindo a segurança. Dessa maneira, é perceptível que haveria sentido em investigar a possibilidade da aplicação da abordagem de fluxos em uma LoRaWAN tendo o entendimento desses seus benefícios, além das necessidades de segurança e administração citadas anteriormente.

Ainda, no caso de um operador de uma rede, se ele desejar saber quais são os padrões de tráfego que ela possui, isso seria possível com o auxílio da ferramenta Netflow. Entender esses padrões é o ponto de partida para a criação de sistemas mais complexos que poderiam utilizar tecnologias como a inteligência artificial para analisar os dados e auxiliar no processo de tomada de decisão dentro de uma empresa. Nesse sentido, seria possível, após a definição de um fluxo numa LoRaWAN, criar modelos preditivos para definir diferentes perfis de clientes e, com isso personalizar o serviço fornecido, causando impacto positivo em sua qualidade.

Um dos muitos desafios do gerenciamento de redes é entender o que está passando pela mesma, por exemplo, identificando qual a relação, em termos de porcentagem, entre diferentes tipos de protocolos, quais seriam as entidades que mais enviam mensagens nessa rede ou mesmo quais os destinos mais comuns desses dados. Todas essas são informações valiosas e possíveis de serem obtidas utilizando ferramentas de análise de fluxos.

Diferentes operadores de rede podem ter diferentes motivos para usar essas ferramentas. De uma perspectiva de segurança, talvez fosse desejável gerar uma base de comparação das características do tráfego da rede e então procurar por anomalias. Al-

ternativamente, poderia haver a necessidade de adicionar um novo dispositivo à rede e, antes disso, seria importante saber como o padrão de tráfego se apresenta.

Para entendermos o que se passa em qualquer sistema, é necessário perceber os elementos que o compõe e gerar informações a partir disso. É possível transformar esses dados, observando as relações existentes em seus padrões e tendências, em conhecimento. O que deriva desse processo de conhecer um sistema é a sabedoria. Com ela, é possível tomar decisões assertivas para, por exemplo, aprimorar a qualidade de diferentes serviços de uma empresa, possivelmente trazendo melhores resultados para a mesma. Tudo isso depende da coleta inicial de dados. A definição de um fluxo numa LoRaWAN, um dos objetivos desse trabalho, é a base que sustenta todo o processo inicial de aquisição e transformação de informações nesse tipo de rede.

Desse modo, seria possível a geração de um conhecimento aprofundado que possibilitaria uma gestão mais eficiente das redes LoRaWAN. Assim, inspirando-se em sistemas como o Netflow, uma ferramenta de gerenciamento de redes LoRaWAN baseada em fluxos poderia ser construída e, possivelmente, entregar, de maneira semelhante à outros gerenciadores de redes sem fio, uma melhor experiência para seus usuários e retorno para os provedores desse tipo de serviço.

1.1 Objetivos

Esse trabalho introduz os conceitos necessários para o entendimento de algumas das necessidades da tecnologia LoRaWAN, apontando que é difícil gerenciar LoRaWAN por ela ser muito nova e ainda não ter propostas sobre as especificidades de sua rede que ajudem um operador a entender o que está acontecendo nela. Para resolver esse problema, portanto, é elaborada uma definição de fluxo nesse tipo de rede, inspirada no que já existe em redes TCP/IP.

Dado o exposto, objetiva-se desenvolver um software para identificação dos fluxos em uma LoRaWAN, de modo a preparar o caminho para que os métodos de análise disponíveis no mercado possam ser empregados (ou que novos sejam criados) e os benefícios gerais dessas aplicações possam ser explorados no contexto dessa tecnologia específica.

Pode-se identificar como objetivos específicos:

- Propor uma definição de fluxos no contexto de redes LoRaWAN;
- Armazenar os fluxos numa estrutura de banco de dados.

1.2 Estrutura da monografia

Esse trabalho está dividido em 6 capítulos. No Capítulo 2 são apresentados os conceitos gerais da tecnologia LoRa e LoRaWAN, bem como o conceito de um fluxo em redes, além de analisar trabalhos que abordam o tema levantado inicialmente. No Capítulo 3, é definida a metodologia a ser seguida e o processo para a definição de um fluxo em uma LoRaWAN. No Capítulo 4, todos os processos necessários para o entendimento do funcionamento do algoritmo são mostrados, bem como os procedimentos necessários para realizar a validação dos dados obtidos a partir do recebimento das mensagens. O Capítulo 5 apresenta uma análise sobre os resultados obtidos a partir da execução do programa e pontos de dificuldades e melhorias. Por fim, o Capítulo 6 apresenta as conclusões e o caminho que poderia ser seguido para complementar a visualização das informações geradas.

2 CONCEITOS GERAIS E TRABALHOS RELACIONADOS

Esse capítulo é estruturado para apresentar a LPWAN (*Low Power WAN*) e, também, discutir em específico a tecnologia LoRaWAN, por causa de suas vantagens quanto ao alcance de comunicação entre dispositivos e eficiência energética. Ainda, um de seus protocolos de comunicação que será abordado é o LoRa, desenvolvido pela Semtech e a LoRa-Alliance [22]. Além disso, será feita uma revisão da utilização da análise de fluxos. Assim, é apresentado um panorama dos conceitos-chave para esse trabalho.

2.1 Redes LoRa

LoRa (*Long Range*) é uma tecnologia LPWAN desenvolvida pela Semtech Corporation [40]. Diferente de sistemas que têm como base a utilização de modulações FSK (*Frequency Shifting Keying*) para gerar baixo consumo energético, LoRa é baseada na modulação CSS (*Chirp Spread Spectrum*) [40]. Essa tecnologia mantém as mesmas características da concorrente, FSK, mas aumenta significativamente o raio de alcance da comunicação [40]. A modulação CSS foi concebida originalmente na década de 1940 para entregar uma forma de comunicação militar [41]. A LoRa, contudo, é uma implementação de baixo custo que visa o uso comercial dessa tecnologia [42].

O principal conceito por trás da tecnologia CSS é o de que um sinal senoidal, cuja frequência varia linearmente no tempo, chamado chirp, possa ser utilizado para disseminar informações através de um espectro mais amplo do que normalmente precisaria ocupar [40]. Essa distribuição uniforme de um dado através dessa maior largura de banda fornece uma resistência contra ruídos e interferências pelo custo de uma baixa eficiência de espectro [43]. Então, uma mensagem LoRa é um sinal chirp que transmite, em longo alcance, essa banda de frequência. Ainda, diferentes taxas de dados podem ser selecionadas ao mudar o seu *Spreading Factor* (SF) [22].

LoRa opera em uma banda ISM (*Industrial Scientific and Medical*) sub-GHz sem licenciamento (915MHz no Brasil, 900MHz nos Estados Unidos e 860MHz na Europa) [17]. Com 125, 250 e 500 KHz de largura de banda, pequenas porções de dados, que vão até 250 bytes, podem ser transmitidas através de distâncias que variam de 5 a 15 Km e tal sistema pode durar de 5 a 10 anos consumindo pouca bateria [17].

Um sistema LoRa é composto por dispositivos finais, gateways e servidores de rede e aplicação, onde os dispositivos finais tem a função de coletar informações e enviá-las para os gateways [44]. Esses que, por sua vez, retransmitem as mensagens entre os dispositivos finais e os servidores de rede previamente configurados. Então, essas mensagens são direcionadas para os servidores de aplicação apropriados para que os mesmos possam realizar o processamento desses dados.

Em uma rede LoRa, dispositivos finais transmitem seus pacotes em modo *broadcast* enquanto os gateways permanecem esperando pela transmissão, ouvindo em todos os canais disponíveis e em todos os SFs (*Spread Factors*) [44]. Uma transmissão de um dispositivo final é recebida com sucesso num gateway se sua potência for maior que o parâmetro RSSI (*Received Signal Strength Indicator*). O RSSI mínimo para a recepção do sinal em diferentes SFs é mostrado na Tabela 1. Os gateways, organizados em turnos, enviam os pacotes decodificados para um servidor de rede central utilizando conexões de internet banda larga, onde os pacotes duplicados são detectados e removidos. Se um pacote não for decodificado corretamente por um gateway (caso de colisões, por exemplo), ainda há uma chance de que ele seja decodificado por outro gateway, resultando em melhores transmissões. O número de gateways que podem ouvir a transmissão de um dispositivo final depende apenas de seu alcance, que por sua vez está ligado diretamente à sua potência de transmissão e o SF utilizado [45].

SF	RSSI (dBm)
7	-123
8	-126
9	-129
10	-132

Tabela 1: RSSI mínimo para decodificação com diferentes SFs.

Fonte: [40].

Por fim, LoRa é o protocolo de camada física desenvolvido e patentado pela Semtech, já o protocolo da camada de enlace é o LoRaWAN, de código aberto. Por conta da patente da LoRa, as informações sobre seu design e implementação não são disponibilizados pela Semtech. Porém, a implementação de seu protocolo já é bem compreendida.

2.2 Panorama da LoRaWAN

LoRaWAN é uma tecnologia desenvolvida por uma associação sem fins lucrativos chamada LoRa Alliance, que definiu as camadas superiores à camada física da LoRa e a arquitetura de rede LoRaWAN [46]. Suas soluções podem ser divididas em três categorias: as baseadas na infraestrutura celular [17, 18], redes que utilizam uma infraestrutura terceirizada [19] e redes LPWAN autônomas sem uma infraestrutura terceirizada [20]. A LoRa Alliance promove uma padronização dessa tecnologia que foca no controle da sua implantação em larga escala, visando um ecossistema de baixo custo e longo ciclo de vida de bateria, com comunicação bidirecional, taxas de dados adaptáveis e esquemas de segurança confiáveis [22].

Ainda, algumas considerações precisam ser abordadas sobre as soluções LoRaWAN. Em primeiro lugar, a tecnologia de redes de celular existente tem uma cobertura ampla, mas sua eficiência energética não se encaixa nos requisitos da LPWAN, já que não foi desenvolvida para comunicação automatizada entre dispositivos finais e a infraestrutura de transporte de dados que os mesmos precisam. As redes de celular já são densas o suficiente e, por isso, não seria possível lidar com a quantidade crescente de novos dispositivos IoT, que causariam demasiada interferência. Em segundo lugar, alguns provedores, como SigFox [19], não podem se adaptar à ambientes diferentes porque não são de código aberto. Finalmente, uma rede LoRa [21] é amplamente utilizada para aplicações LPWAN precisamente porque ela é uma tecnologia de código aberto e permite uma configuração de rede autônoma a um baixo custo.

Nesse contexto, as características da LoRa fazem dela uma excelente escolha para diversas aplicações IoT [20]. Sua arquitetura típica consiste em dispositivos finais que coletam informações e as mandam para gateways, responsáveis por retransmiti-las para o servidor de rede. Por fim, esse servidor é configurado para direcionar as mensagens ao servidor de aplicação adequado, onde é realizado o processamento de dados.

Baseando-se na camada física da LoRa, a LoRa Alliance [22] estabelece o padrão do protocolo de acesso ao meio (*Medium Access Control* - MAC) LoRaWAN (*LoRa Wide-Area Network*)[23,24]. Muitas das aplicações IoT citadas anteriormente requerem as propriedades encontradas na LoRa [25, 26]. Por isso, a indústria e a academia enxergam grande potencial na LoRa para aplicações IoT baseadas em LPWAN [27, 28].

Também, a LoRa apresenta-se como uma solução atrativa para aplicações IoT industriais graças à sua robustez [29]. No entanto, o protocolo MAC da LoRaWAN [30, 31] é feito para comunicações esporádicas entre os dispositivos. Por essa razão, a LoRaWAN adota o protocolo ALOHA, que não provém o necessário para suportar o volume de dados em tempo real das aplicações industriais [40]. Contudo, uma abordagem onde um nó central administra o acesso ao meio, de acordo com uma ordem predefinida, seria mais apropriado para lidar com tantos dados em tempo real. Dessa maneira, alguns trabalhos [32] propuseram o Industrial LoRa, um protocolo MAC centralizado para topologias em estrela para serem usados na LoRa e que permite o suporte às comunicações em tempo real para as aplicações industriais. Essa abordagem é baseada na topologia de rede de comunicação móvel, onde aparelhos são conectados à uma estação base através do meio sem fio e diversas estações bases são interconectadas à uma estação base controladora através de conexões cabeadas [47]. Isso mostra o interesse e possibilidade de adaptação para incorporar essa tecnologia nos mais diferentes sistemas.

São quatro os elementos que compõem a arquitetura LoRaWAN: dispositivos IoT, gateways LoRaWAN, servidores de rede e servidores de aplicação [48]. Dessa maneira, essa estrutura garante que haja um único salto entre um dispositivo IoT e um gateway através

de diversos canais, retirando a necessidade de construir e manter uma rede de múltiplos saltos mais complexa. O gateway se comunica com o servidor de aplicação através de uma rede IP onde as taxas de dados variam entre 0,3 kbps e 50 kbps e seus esquemas de segurança implementam algoritmos para garantir a autenticação, integridade dos pacotes e uma criptografia de ponta a ponta [22].

A LoRaWAN propõe três classes de operação para atender a diferentes requisitos das suas muitas aplicações [40]. Dispositivos finais comumente implementam um modo de operação de Classe A. Alternativamente, existem outros modos opcionais, como o de classe B e C. Os dispositivos finais que operam nos modos de classe A e B são alimentados por uma bateria, enquanto que os que operam na Classe C recebem a alimentação de uma matriz externa.

Na classe A, depois de enviar mensagens confirmadas, os dispositivos ficam na espera de um reconhecimento do servidor de rede durante dois espaços de tempo pré-estabelecidos, o que aprimora a robustez da transmissão das mensagens, e eles não esperam respostas do servidor quanto às mensagens não confirmadas. Além disso, os de classe A consomem menos energia que os demais. O modo de operação de Classe B coloca à disposição janelas para recepção de mensagens adicionais que são programáveis por gateways através de pacotes de referência. Por fim, o modo de Classe C não tem restrição de downlink e pode receber diversas dessas mensagens a qualquer momento, desde que não esteja em transmissão.

A LoRaWAN incorpora custos adicionais nos pacotes para que seja possível habilitar os recursos MAC provenientes de suas classes de dispositivos. Esses custos incluem cabeçalhos MAC, comandos e um MIC (*Message Integrity Checks*) para avaliar a integridade dos dados [87].

O protocolo de comunicação LoRaWAN possui dois tipos de mensagens, up-link e down-link. As mensagens de up-link são enviadas por dispositivos LoRa, passando por gateways, em direção ao servidor de rede. O quadro de mensagens LoRa é constituído por um preâmbulo, o cabeçalho da camada física, o cabeçalho CRC (Cyclic Redundancy Check) e pela carga útil da camada física, que inclui o seu próprio CRC no fim da mensagem para garantir a proteção da integridade de todo o pacote, sendo o cabeçalho e o CRC opcionais. O cabeçalho, quando utilizado, tem como função indicar o tamanho da carga útil, sendo disponibilizado apenas um byte, limitando a carga apenas para o tamanho de 255 bytes [87].

Preâmbulo	Cabeçalho da camada física	Cabeçalho CRC	Carga útil da camada física	CRC
-----------	----------------------------	---------------	-----------------------------	-----

Figura 1: Quadro de LoRa.

Diversos trabalhos acadêmicos que avaliam a performance da LoRaWAN foram publicados na última década [49, 50, 51, 52, 53, 54] e todos trazem diferentes perspectivas sobre os aspectos da tecnologia. Ainda, introduzem as vantagens e limitações da tecnologia quando aplicada em diferentes cenários de escalabilidade, dispersão geográfica e padrões de transmissão de dados.

2.3 Fluxo em redes

Um fluxo é definido como um conjunto de pacotes IP passando por um ponto de observação numa rede durante um certo intervalo de tempo tal que todos os pacotes pertencentes a ele tenham um conjunto de propriedades em comum [35]. Essas propriedades em comum podem incluir os campos de cabeçalho dos pacotes, como o endereço IP da fonte e destino, número de portas, conteúdo dos pacotes e metadados.

Trabalhos na área de tratamento de fluxos datam da década de 1990 [55] e tornaram-se a base para os protocolos modernos como NetFlow e IPFIX (*IP Flow Information eXport*) [56]. O primeiro design do conceito de fluxo foi feito como parte do Internet Accounting (IA) Working Group (WG) of the Internet Engineering Task Force (IETF) [55]. Esse WG foi encerrado já em 1993 por conta de falta de um interesse comercial e, também, pela crença de que a Internet deveria ser livre, o que significa que nenhuma captura de tráfego deveria ser feita que levasse a casos de monitoramento, responsabilização de indivíduos e seus atos e etc [55]. Contudo, em 1995, o interesse em exportar o fluxo de dados para análise de tráfego ressurgiu [57]. Então, em 1999, o WG publicou um framework genérico para medição de fluxo, o RTFM Traffic Measurement System, com maior flexibilidade em relação às definições de fluxo e suporte a fluxos bidirecionais [58].

Em paralelo à RTFM, a Cisco trabalhou em sua própria tecnologia de exportação de fluxos, o NetFlow, que foi originalmente concebida para operar em switches. Nesse tipo de fluxo, as informações são mantidas numa cache de fluxo e as decisões posteriores são feitas apenas no plano de controle de um dispositivo de rede e apenas para o primeiro pacote de um fluxo. Os demais pacotes são tratados exclusivamente no plano dos dados

[59]. NetFlow foi patenteada pela Cisco em 1996 e sua primeira versão a ser amplamente adotada pelo público foi a NetFlow v5 [60], disponível ao público apenas em 2002.

Desde então, a tecnologia evoluiu em versões e funcionalidades [61, 62, 63, 64, 65, 66]. Assim, nota-se que o termo NetFlow é altamente carregado dentro de sua literatura, fazendo referência a diferentes protocolos de exportação de fluxos, dos quais ainda há implementações de terceiros. Refere-se, também, à uma tecnologia de exportação de fluxos, possuindo uma série de implementações de captura de pacotes e medição de fluxos.

Além de serem bem adaptados para o uso em redes de alta velocidade, os protocolos de exportação de fluxos e suas tecnologias trazem diversas outras vantagens quando comparados com o método padrão de captura de pacotes. Ainda, é possível verificar que eles são amplamente implantados principalmente por conta de sua integração em dispositivos de encaminhamento de pacotes de última geração, como roteadores, switches e firewalls [67]. A exportação de fluxos é bem compreendida já que é, também, amplamente utilizada para casos de análise de segurança, planejamento de capacidade e processo de perfilamento numa rede [68, 69].

A arquitetura de um sistema de monitoramento de fluxos é composta por diversos estágios [70]. O primeiro estágio é a Observação de Pacotes, onde pacotes são capturados de um certo Ponto de Observação e pré-processados. Esses Pontos de Observação podem ser as interfaces dos dispositivos de encaminhamento de pacotes, por exemplo.

O segundo estágio é o de Medição e Exportação de Fluxo. No processo de medição, os pacotes são agrupados em fluxos, como definidos anteriormente [35]. Depois que é considerado que um fluxo acabou, um registro de fluxo é exportado, o que quer dizer que esse registro é colocado em um datagrama do protocolo de exportação de fluxo implantado. Os registros dos fluxos são definidos como uma informação sobre um fluxo específico que foi observado em um dado ponto de observação [35]. Isso pode incluir tanto as propriedades características de um fluxo (como endereço IP) quanto as propriedades medidas (contadores de pacotes e de byte). Esse conceito pode ser melhor entendido se comparado a registros em uma tabela de um banco de dados, com uma coluna diferente para cada propriedade em específico.

Formato de cabeçalho de um Fluxo (Netflow v.5)			
versão	contador	tempo de inicialização do sistema	
unix em segundos		unix em nano segundos	
sequência de fluxo	tipo do mecanismo	id do mecanismo	intervalo de amostragem

Figura 2: Cabeçalho do fluxo Netflow.

O terceiro estágio é o de Coleta de Dados, cuja principal função é a de receber, armazenar e pré-processar os fluxos de dados gerados pelos estágios anteriores. As operações de pré-processamento mais comuns incluem agregação, filtragem e compressão de dados.

O estágio final é o de Análise de Dados, que possui frequentemente uma natureza exploratória e é comum ser feito manualmente. Já em ambientes operacionais, as funções de análise são geralmente integradas ao estágio de Coleta de Dados, o que faz com esse processo possa ser parcialmente automatizado. As funções mais comuns de análise incluem correlação e agregação, perfilamento de tráfego, classificação e detecção de anomalias. Todos esses estágios são conceituais e podem ser integrados de diferentes maneiras a depender do tipo de construção do sistema.

2.4 Trabalhos Relacionados

Trabalhos recentes trataram das capacidades LoRaWAN [73, 74, 75], desempenho [76] e configuração de parâmetro [77, 78] para aplicações de monitoramento. Por exemplo, existe a plataforma (LoMM) para gerenciamento de redes IoT baseadas em LoRaWAN [79]. O design modular do LoMM permite distribuir o gerenciamento e controle de EDs de uma estrutura backend remota para um backend de borda, aliviando a carga da arquitetura de backend descentralizada de tarefas que podem realmente ser realizadas de uma forma descentralizada, adaptando, assim, a configuração da rede para a real forma como foi implantada.

Pesquisadores e especialistas podem facilmente utilizar esta ferramenta para gerenciar sua própria plataforma de testes LoRaWAN [80, 81] e implementar algoritmos específicos e políticas de gerenciamento de suas redes enquanto monitoram os efeitos no desempenho das mesmas. Além disso, a capacidade de acessar as informações da rede em implantações reais permite aplicar métodos inteligentes de análise de dados, eventualmente alimentados por motores de aprendizagem de máquina, com o objetivo de desenhar percepções sobre, por exemplo, comportamentos e anomalias para que contramedidas sejam tomadas prontamente.

Em todas as abordagens, as estratégias implementadas possuem limitações quanto às informações obtidas sobre a rede e divergem principalmente na aplicação de análise de fluxos para o monitoramento da mesma. A solução trazida nesse trabalho traz uma base para que ferramentas de análise de fluxos possam compilar um conjunto maior de informações sobre uma LoRaWAN, possibilitando que estratégias de gerenciamento mais complexas desse contexto possam ser aplicadas.

2.5 Considerações finais

Esse capítulo trouxe um entendimento sobre os conceitos do tema desse trabalho. Partindo de LoRa, explicando sua definição, depois mostrando como é feita sua modulação e uma breve descrição sobre o funcionamento do seu SF. Em seguida, é definido o conceito de LoRaWAN, passando por sua topologia, pilha de protocolos e, finalmente, a definição das classes de dispositivos. Também, é discutido o conceito de fluxo em redes, suas origens, suas funções e a conceituação de sua arquitetura. Por fim, uma breve análise sobre os trabalhos existentes que tocam o tema é feita.

3 METODOLOGIA

Esse capítulo é estruturado para explicar como a pesquisa proposta por esse trabalho será realizada, seus métodos e encaminhamento para chegar aos resultados obtidos.

3.1 Planejamento e Escolha de Ferramentas

O tipo de pesquisa utilizada no presente trabalho foi experimental e exploratória em relação aos objetivos [1]. A metodologia empregada envolve uma investigação sobre a melhor seleção de campos de uma mensagem numa LoRaWAN com intuito de identificar uma formatação genérica que possa identificá-los como parte de um fluxo nesse tipo de rede.

Assim, se selecionarmos determinados campos do escopo de uma mensagem numa rede LoRaWAN, poderemos utilizá-los para classificá-los e determinarmos se os mesmos pertenceriam a um fluxo de dados específico. Assim, seria definido um fluxo nesse tipo de rede e métodos de análise de fluxo poderiam ser aplicados na mesma. Por extensão, os benefícios encontrados atualmente na utilização de análise de fluxos poderiam ser trazidos para essa rede específica. Portanto, se campos puderem ser escolhidos dentro do escopo de uma mensagem LoRaWAN e esses atenderem à definição inspirada em fluxos de redes TCP/IP, então, a análise de fluxo poderia ser aplicada sobre redes LoRaWAN.

Os procedimentos de investigação citados anteriormente visam a coleta de mensagens numa LoRaWAN e posterior experimentação com esses dados, procurando validar a hipótese levantada. Essa coleta será feita através de uma comunicação WebSockets, dado que essa é uma forma de comunicação presente em soluções comerciais para repasse de mensagens nesse tipo de rede. Ao longo da construção da pesquisa, o primeiro procedimento realizado diz respeito à coleta de dados de uma LoRaWAN específica. Em seguida, serão avaliados os campos que cada mensagem LoRaWAN possui para melhor entendimento deles com intuito de facilitar a etapa de seleção e teste. A etapa de seleção propõe uma análise sobre as características que mensagens têm em comum (como fonte e destino) que podem se adequar ao conceito de um fluxo. Os dados extraídos dos campos das mensagens serão formatados para que possa ser realizada uma análise sobre os mesmos.

Será realizada a criação de um banco de dados para armazenar as informações dos fluxos. A modelagem do banco de dados irá representar a estrutura que receberá os dados gerados a partir da classificação aplicada. Todas as restrições associadas ao banco de dados foram inseridas na própria interface, como por exemplo a formatação do dado. O CRUD (create (criar), read (ler), update (atualizar) e delete (deletar)) representa as 4 operações fundamentais de um banco de dados relacional. No caso específico desse

trabalho, porém, a aplicação apenas criará o dados porque sua função final é apenas de reter as informações gerada pelo software construído.

Por fim, foi buscada uma linguagem de programação para englobar o ambiente de desenvolvimento e um sistema de gerenciamento de banco de dados. O principal critério para a escolha de uma linguagem de programação para o desenvolvimento da proposta desse trabalho foi que ela fosse orientada a objeto, visando o reuso do código. Sua seleção passou pelo critério de popularidade no mercado e familiaridade com a mesma. A linguagem escolhida foi, portanto, Python. É uma linguagem de programação popular e que comporta uma variedade de bibliotecas pertinentes para o objetivo final. Sua portabilidade e facilidade de compreensão são pontos importantes que contribuirão na decisão de sua adoção. Quanto ao banco de dados, o escolhido foi o PostgreSQL por ser popular e cujas funções podem todas serem consideradas gratuitas.

Com base no problema exposto, e com intuito de otimizar a realização do projeto, o planejamento de seu desenvolvimento foi dividido em etapas. Cada uma dessas é detalhada a seguir:

1. Revisão da literatura do protocolo LoRaWAN: A etapa tem como objetivo o domínio do uso do protocolo LoRaWAN e a manipulação de mensagens através de uma série de testes para validação e instrumentação do LoRaWAN.
2. Funcionamento da rede específica: Para extrair os dados necessários à análise dos campos dos pacotes, será necessário compreender o funcionamento da rede específica de forma completa.
3. Dados da rede específica: Direcionamento do estudo para a identificação dos dados que trafegam nessa rede para o entendimento do arranjo de tudo o que está sendo gerado nela para melhor escolher e aplicar a estratégia de definição de um fluxo nela.
4. Uso da LoRaWAN: Desenvolvimento de um de software com base nos parâmetros de uso da classificação de um fluxo e os campos dos dados que foram considerados relevantes. Definição de uma estrutura de banco de dados para armazenamento das informações sobre os fluxos.
5. Teste de validação: Apenas após a estruturação dos campos relevantes dos pacotes da rede específica observada, é possível dar início aos testes para validação da hipótese levantada inicialmente. Espera-se que essa seja uma etapa crítica, porque os testes têm o objetivo de validar a estrutura dos campos das mensagens que foram selecionados para serem utilizadas.
6. Documentação: Os procedimentos serão documentados durante todo o período de desenvolvimento. Cada uma das partes citadas na proposta será exposta de forma detalhada.

Esse capítulo trouxe aprofundamento sobre as etapas que serão utilizadas para realizar o planejamento do desenvolvimento do presente projeto. Todos os tópicos apresentados aqui servem como base para o entendimento da metodologia utilizada nos procedimentos seguintes.

3.2 Comunicação e Definição dos Fluxos

A especificação WebSocket define uma API que estabelece conexões de “soquete” entre um navegador da web e um servidor. Então, é possível estabelecer uma conexão persistente entre um cliente e um servidor e ambas as partes podem começar a enviar dados a qualquer momento. Por isso, “websockets” é o nome da biblioteca que foi utilizada para construir servidores e clientes WebSocket em Python. Construída sobre “asyncio”, a estrutura assíncrona padrão do Python, ela fornece uma API elegante baseada em co-rotina. A partir dessa estrutura, é possível estabelecer uma conexão para receber mensagens LoRaWAN de uma API de dados (que é uma interface de transmissão bidirecional) baseada em WebSockets e projetada para troca de mensagens entre um servidor de aplicação e um servidor de rede em tempo real. É uma interface cujo objetivo é exclusivamente a transmissão de dados, não gravando qualquer mensagem trocada. As mensagens possuem o formato JSON, um formato de arquivo de dados que usa texto legível por humanos para armazenar e transmitir objetos de dados que consistem em pares de valor de atributo e matrizes. A figura a seguir apresenta uma mensagem JSON com campos comuns de um “downlink” da LoRaWAN.

```
{
  "end_device_ids" : {
    "device_id" : "dev1",
    "application_ids" : {
      "application_id" : "app1"
    },
    "dev_eui" : "0004A30B001C0530",
    "join_eui" : "800000000000000C",
    "dev_addr" : "00BCB929"
  },
  "correlation_ids" : [ "as:downlink:..." ],
  "received_at" : "2020-02-17T10:32:24...",
  "downlink_queued" : {
    "session_key_id" : "AXBSH1Pk6Z0G166...",
    "f_port" : 15,
    "f_cnt" : 1,
    "frm_payload" : "vu8=",
    "confirmed" : true,
    "priority" : "NORMAL",
    "correlation_ids" : [ "as:downlink:..." ]
  }
}
```

Figura 3: Demonstração de mensagem JSON.

Dessa maneira, utilizando as bibliotecas mencionadas e tendo uma URL de conexão para a API de dados, é possível receber, através da comunicação estabelecida, as mensagens JSON provenientes das trocas de mensagens entre os dispositivos finais e um servidor de rede de acordo com as definições de uma LoRaWAN. Então, para criação de um modelo de fluxos que pudesse se adaptar ao que é encontrado nos campos dessas mensagens, foi necessário retomar o conceito que diz que um fluxo em uma rede TCP/IP seria composto por um endereço IP de origem e destino, uma porta de destino e origem, o protocolo da camada de rede, o tipo de serviço e a interface lógica relacionada. Além disso, as mensagens possuem um tipo que as define quanto às suas funcionalidades [86].

Olhando para esse fluxo IP, nota-se que essas mensagens essencialmente compilam informações de origem e destino que constroem um caminho pelo qual elas fluem, nesse tipo de rede, são pacotes os dados que fluem. Com esse conceito, partindo agora para o contexto das mensagens numa LoRaWAN, foi interpretado, então, que um gateway seria o destino desse fluxo de mensagens. Esse campo do JSON foi relacionado dessa forma por causa das análises que a abordagem permite, podendo trazer informações relevantes quanto à cobertura ou mesmo sobre a sobrecarga de um gateway quando colocamos em perspectiva o par dispositivo e gateway. Nesse sentido, levando em consideração a topologia de uma LoRaWAN, o ponto de origem de uma mensagem seria uma porta mais um dispositivo e seu endereço único na rede.

A partir das informações no JSON e tendo em mente os conceitos desenvolvidos anteriormente, os seguintes campos de uma mensagem foram escolhidos para definir um fluxo numa LoRaWAN: tipo de mensagem, id único do dispositivo, id do dispositivo na rede, número da porta, id da aplicação e o id do gateway. O tipo de mensagem, comumente de “uplink” ou “downlink”, serviria para orientar a direção do fluxo, onde que, se os demais campos são idênticos, fluxos diferentes são formados para tipos de mensagens diferentes. Assim, constituem-se as informações de origem como: id único do dispositivo, id do dispositivo na rede e o número da porta e, para as informações de destino, há o id da aplicação e o id do gateway.

Desse modo, o software recebe mensagens e executa funções que analisam o JSON e preparam um fluxo (atualizando esse fluxo na estrutura de cache). Então, ele vai receber o JSON como parâmetro, performando uma análise, e atualizando uma estrutura de dados de fluxo (um cache de fluxos). Após um determinado espaço de tempo, 30 minutos segundo as definições inspiradas no Netflow, os dados são removidos da cache e despejados num banco de dados, que vai armazenar as informações para que elas possam ser acessadas posteriormente. O que será coletado, além das informações dos campos já mencionados, são: o número de mensagens, o número de bytes e um registro de tempo para quando o fluxo foi gravado, no padrão Unix

Nome	Dado armazenado
gateway	O endereço MAC do gateway
device_addr	O endereço do dispositivo na rede (device_addr da especificação LoRaWAN)
port	Porta do pacote LoRaWAN
type_message	O tipo de mensagem para realizar uma divisão por funcionalidade
device	O identificador do dispositivo (dev_eui da especificação LoRaWAN)
application	O identificador da aplicação (app_eui da especificação LoRaWAN)
message_quantity	Quantidade paratal de meServidor de RedeageServidor de Rede nesse fluxo
size_bytes	Tamanho paratal em bytes dServidor de Aplicação meServidor de RedeageServidor de Rede no mesmo fluxo
timestamp	Tempo de registro do fluxo no banco de dados

Figura 4: Estrutura do fluxo.

4 FLUXO LoRaWAN

Nessa seção será apresentado o funcionamento do software no que se refere aos processos necessários para captura, manipulação e gravação dos dados gerados a partir da troca de mensagens numa LoRaWAN.

4.1 Criação de Fluxos

O processo do software, cujo código foi escrito em Python, começa com a criação de um cliente que se conecta à API de dados e permanece "escutando" continuamente a chegada de mensagens. Uma URL, vinda de um arquivo Python de configuração, criado para gravar os dados relevantes do problema, chamado "cache", é passada dentro de um loop para garantir que o programa esteja sempre recebendo mensagens em tempo real.

Essas mensagens JSON que chegam passam por funções que irão analisar seu conteúdo classificando-as como um novo fluxo ou como parte de um fluxo já existente. O fluxo criado, então, é adicionado à cache até que o tempo de 30 minutos acabe.

4.2 Controle de Tempo

O controle do tempo é feito pela união de dois módulos do Python, um deles é o "threading" e o outro é o "time". A biblioteca padrão do Python fornece "threading", que contém a maioria dos primitivos que serão vistos aqui.

Para a thread, é passada uma função que será executada por nela. Dentro dessa função, há uma chamada para outra função, "countdown", pertencente a um outro arquivo Python chamado "timer" que engloba as informações relacionadas ao controle de tempo do software. Uma variável "t" do arquivo "cache" contém a informação de tempo que o programa deve passar recebendo as mensagens JSON até que ele esvazie os fluxos criados a partir delas para serem registrados no banco de dados.

Em "countdown", é utilizado o segundo módulo, "time". Este módulo fornece várias funções relacionadas ao tempo. Esse é um módulo disponível para todas as plataformas, mas nem todas as suas funções estão disponíveis por conta de divergências entre interações com os sistemas operacionais. Contudo, a função utilizada nesse código é multiplataforma, podendo ele ser executado tanto em Windows como em Linux e variantes. A sintaxe da função é: `time.sleep(sec)` em que `sec` é o argumento que especifica o número de segundos durante os quais o programa deve ser suspenso antes de decrementar um contador de tempo.

O parâmetro de tempo é passado e o programa aguarda os 30 minutos até que, finalmente, ele salva o número de itens que estão na cache nesse instante de tempo e faz

uma retirada desse mesmo número de itens dela, tudo isso enquanto o programa continua recebendo novas mensagens e criando fluxos.

Quando um fluxo é criado, significa que o objeto da classe “Fluxo” foi instanciado e esse objeto tem os campos: gateway, device, device_addr, application, port, type_message, message_quantity, size, active_time. Todos referentes aos dados coletados do JSON com a exceção do campo “active_time” que será utilizado para calcular quanto tempo aquele fluxo esteve ativo. Toda vez que um fluxo é criado ou que uma nova mensagem passa por esse fluxo, o “active_time” é atualizado para registrar aquele instante de tempo. O tempo é proporcional à capacidade de armazenamento do sistema que vai executar o código.

4.3 Cache

A cache, então, é uma lista que armazena os objetos da classe “Fluxo” que são criados pela função “create_flow()”, que retira os dados dos campos do JSON e verifica a existência dos mesmos na cache para que, caso existam, o número de mensagens e bytes serem atualizados, ou que seja criado um fluxo completamente novo. Só após o tempo especificado é que essa lista é esvaziada pela função “cacheClear()”, que vai retirar esses itens, começando pelos mais antigos, e fará o registro dos fluxos considerados ativos no banco de dados através da função “reg_database()”, chamada em seu escopo. As informações contidas no arquivo “cache.py” vão desde a declaração da lista que servirá como cache para armazenar os objetos de “Fluxos” até os valores de tempo e, também, informações para conexão com a API de dados e, posteriormente, o próprio banco de dados que fará o registro final.

4.4 Banco de Dados

A modelagem lógica do banco de dados é mostrada na figura abaixo. Todos os campos dessa tabela única estão diretamente relacionados à concepção do que seria um fluxo numa rede LoRaWAN. Ela possui os pontos relacionados à origem e destino das mensagens e, também, outras informações relevantes para a análise posterior dos dados. A presença de “time_stamp” como chave primária marca a temporalidade dos dados para que os mesmos possam gerar informações concisas em uma possível *dashboard* como na Figura 6.

gateway	character varying(16)	PK
device_addr	character varying(8)	PK
port	smallint	PK
type_message	character varying(20)	PK
time_stamp	integer	PK
device	character varying(16)	
application	character varying(16)	
message_quantity	integer	
size_bytes	bigint	

Figura 5: Modelagem tabela fluxos do banco de dados.

Nessa modelagem, os campos apresentados na Figura 5 representam, respectivamente, o número identificador do gateway, o número identificador de endereço do dispositivo na rede, a porta pela qual chegou o dado, o tipo de mensagem, uma marcação de tempo que representa quando as informações chegaram no banco de dados, o número único do dispositivo final, o identificador da aplicação, número de mensagens no mesmo par origem e destino e o número de bytes gerados pelo conteúdo dessas mensagens.

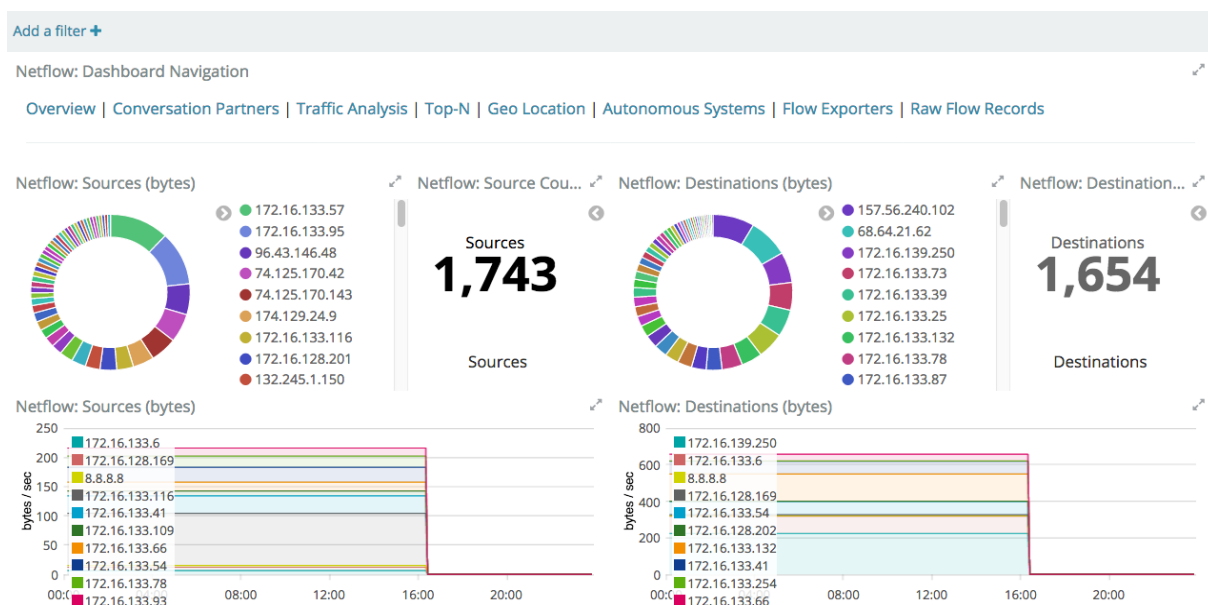


Figura 6: Dashboard Netflow.
Fonte: [83].

Ainda, o módulo Psycpg da linguagem de programação Python está sendo utilizando nessa implementação como o adaptador de banco de dados PostgreSQL. Suas principais características são a implementação completa da especificação Python DB API 2.0 e a segurança de thread (várias threads podem compartilhar a mesma conexão).

Desse modo, o primeiro procedimento que essa função irá realizar é o de estabelecer uma conexão com o banco de dados, cuja modelagem foi apresentada anteriormente.

Na sequência, tendo recebido um objeto da classe “Fluxo” como parâmetro, a função recolhe os dados desse objeto para que sejam inseridos no banco de dados somente se o fluxo não tiver passado do tempo de validade e ser considera um fluxo expirado.

Então, é necessário realizar um conjunto de procedimentos que vão encerrar corretamente a conexão criada e fixar a mudança no banco de dados de fato.

Por fim, o programa apresenta as características do fluxograma da figura a seguir, com as etapas principais de tomada de decisão explicitadas.

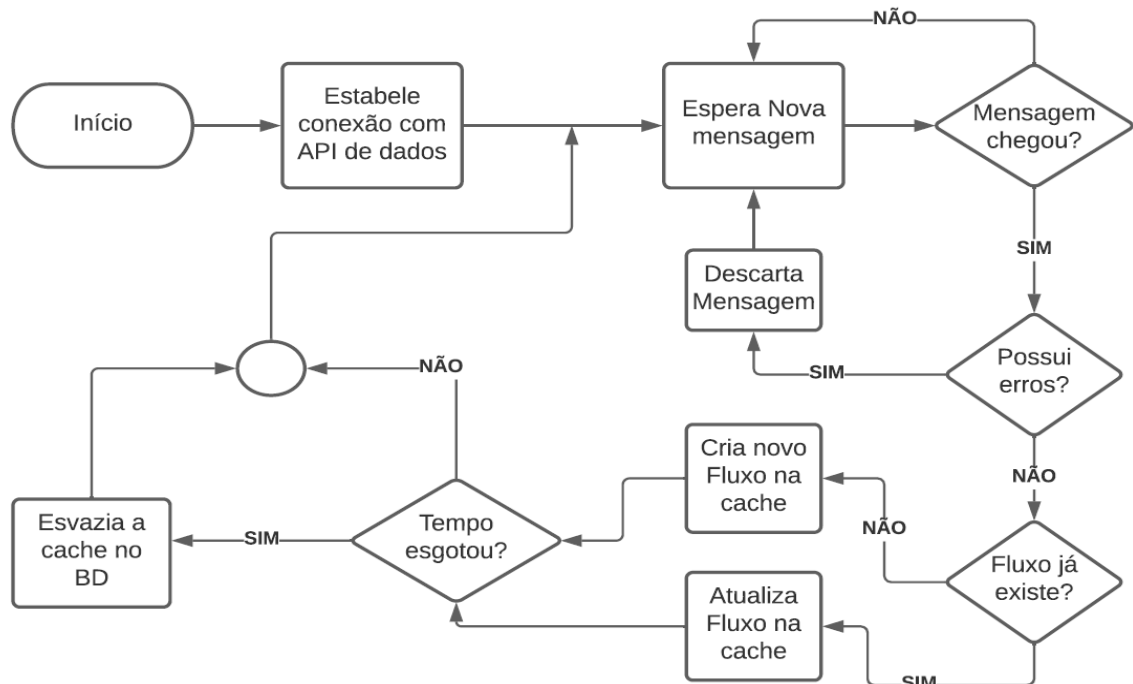


Figura 7: Fluxograma de execução.

4.5 Módulo LoRaONE

O módulo transceptor LoRaONE usa a tecnologia LPWAN LoRa para comunicação com a rede LoRaWAN.

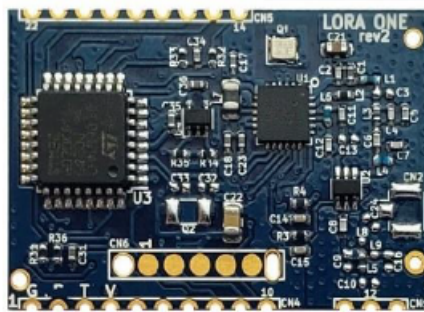


Figura 8: Módulo LoRaONE .
Fonte: [84].

O módulo LoRaONE atua como um modem de comunicação de comando AT, usando o protocolo LoRaWAN V1.0.3 e pode operar nas classes de protocolo A e C. Ele foi projetado para ser um módulo LoRaWAN de comando AT de fácil uso, tamanho pequeno e baixo consumo de energia [84].

O módulo LoRaONE possui RF, controlador e processador API e segue os perfis AU915 / LA915. O LoRaONE é certificado pela Anatel e Everynet (EhThingz), economizando custos e tempo de certificação significativos, e pode ser facilmente configurado para a rede LoRaWAN [84].

O módulo LoRaONE combina um formato pequeno de 21,5 x 33 x 2,4 mm, com 7 GPIOs para conectar e controlar sensores e atuadores. O módulo é conectado a um MCU host, por meio de um UART de 9600 bps, e pode ser facilmente usado em uma ampla variedade de aplicações IoT [84]. O módulo possui os seguintes recursos:

- Comunicação de longo alcance;
- Baixo consumo de energia;
- Tensão de alimentação 3,1 - 3,6 Volts;
- Faixa de temperatura: -40 ° C a + 85 ° C;
- Opera na Banda 915 MHZ (Perfis AU915 e LA915);
- Potência de saída ajustável até +20 dBm;
- Alta sensibilidade do receptor até -137 dBm;
- Protocolo LoRaWAN Classe A e C;
- Interface de comando AT fácil de usar em um UART de 9600 bps;
- 7 GPIO para controle, status e expansão ADC;

- Tipo de montagem PCB: Castellation SMT;
- Certificação Anatel e Everynet EhThingz;
- Favorável ao meio ambiente

Em conjunto com o módulo, há um dispositivo UART USB que é utilizado para inserir os comandos para realizar funções de comunicação entre o módulo e a rede. O FT232R, uma interface UART USB para serial, foi, então, utilizado em conjunto com o módulo LoRaONE.

Então, é preciso realizar um conjunto de procedimentos que são necessários para utilizar o dispositivo, viabilizando que sejam inseridos comandos pelo computador. Os procedimentos seguidos para tanto foram:

- Instalar o driver para o ftdi “FT232R”. É importante identificar em qual porta ele está conectado no computador.
- No caso, foi utilizado um emulador de console para porta serial chamado PuTTY. É um software gratuito para emular console de porta serial para inserir as linhas de comando do dispositivo.
- No terminal serial, é possível obter uma informação importante através do comando “AT+DEUI=?”, que é o código que representa o identificador único atribuído ao dispositivo utilizado pelo fabricante do chip.

Esses procedimentos e informação são essenciais para dar continuidade ao processo descrito posteriormente englobando a conexão com a API de dados.

4.6 Everynet e testes

As redes Everynet permitem gerar dados em tempo real a partir de sensores de baixo custo implantados em amplas áreas e alimentando sistemas analíticos de big data na nuvem [85]. Na plataforma da empresa, em seu portal de gerenciamento de servidor de rede, há uma opção que permite conectar um novo dispositivo. Selecionando essa opção, serão mostradas informações que precisam ser preenchidas para que seja feita a configuração da conexão entre a Everynet e o dispositivo, que nesse caso é essencialmente o LoRaONE. O procedimento genérico de configuração simplificada dos campos que precisam ser preenchidos é o seguinte:

- Device EUI: Obtido pelo comando mencionado anteriormente.

- Application EUJ/ Device address/ Network session key/ Application session key: geradas aleatoriamente na plataforma de gerenciamento.
- Tags: Teste-J
- Uplinks e Downlinks: ON
- Strict counters: OFF
- Activation: ABP
- Encryption: NS
- Class: A
- Counter size: 4
- Rx Window: AUTO
- Band: AU915-928A
- ADR Mode: ON
- Live stream: JOIN, UPLINK, DOWNLINK, ERROR, LOCATION, STATUS

Assim, seguindo esses passos, foi possível realizar a inicialização da conexão do dispositivo LoRaONE como descrito em [84]. E então, foram utilizados os comandos de envio de mensagens descritos em [84] para realizar os testes de software com a interceptação da comunicação de mensagens de “uplink” e “downlink”.

Os testes do algoritmo foram feitos em duas etapas. A primeira delas aconteceu conectando o módulo LoRaONE à rede da empresa Everynet. Ela possui uma plataforma de gerenciamento de dispositivos, a qual tive acesso para conectar o meu. Além disso, ela tem uma API de dados baseada em websockets que mostra, em tempo real, todas as mensagens que passam pelo seu servidor de rede, mas sem armazená-las.

Então, a própria plataforma de gerenciamento da Everynet permite criar uma URL com um token para conexão websocket com sua API de dados. Criando um, foi possível conectar o meu programa à API. Desse modo, enviando mensagens com o dispositivo LoRaONE cadastrado na rede da Everynet, foi possível receber as informações JSON através da conexão websocket estabelecida. Depois, foi avaliado se os fluxos estavam sendo criados e posteriormente registrados corretamente no banco de dados.

A segunda parte foi semelhante à primeira e contou com dados da empresa NLT Telecom, que mantém parceria com a UFPB. A NLT colheu dados referentes à 24 horas de transmissões reais de mensagens dos dispositivos em sua rede LoRaWAN. A empresa

fez previamente o tratamento desses dados, anonimizando apenas as informações sensíveis que poderiam identificar seus usuários, garantindo a segurança de todos, e os organizou em uma planilha Excel, que foi o arquivo recebido. Essa planilha contém as informações de milhares de mensagens. Em posse desse arquivo e com o auxílio de um aluno de iniciação científica da universidade, foi criado um programa em Python que utiliza os dados da NLT para transmitir continuamente os dados da planilha, simulando o servidor real e incluindo os tempos reais de transmissão de cada mensagem. Então, foi criado um servidor websocket ao qual o programa poderia se conectar através de uma URL, assim como foi feito anteriormente no caso da Everynet. Desse modo, ao invés do programa receber as mensagens de um único dispositivo, como na primeira parte, agora ele recebe dados de inúmeros dispositivos que transmitiram na rede da NLT através do servidor websocket, responsável por transmitir os dados, semelhante ao que acontecia, também, na etapa anterior. Em suma, enquanto que na primeira parte havia uma conexão websocket com a API de dados da Everynet, na segunda parte foi feita uma conexão websocket com o servidor que possuía os dados da NLT e simulava a API de dados da Everynet, mas com mensagens reais.

5 RESULTADOS E DISCUSSÃO

Nesse capítulo, será mostrado o resultado obtido depois do processamento de um conjunto de dados disponibilizados pela NLT Telecom, operadora móvel virtual que oferece conectividade LoRaWAN.

5.1 Caracterização dos dados e configurações

Os dados disponibilizados compreendem um conjunto de mensagens pertencentes a uma LoRaWAN e representam a reprodução fiel de todas as mensagens que chegaram à rede dentro de um período de 24 horas. São cento e cinquenta mil mensagens JSON cujas informações sensíveis foram removidas ou criptografadas. Os campos removidos foram *payload*, *encrypted payload*, *latitude*, *longitude*, e *altitude*. Já os campos criptografados, ou seja, que foram substituídos pela saída de uma função hash criptográfica, foram *network*, *gateway*, *device*, *device address*, e *application*.

Foi criado, então, um servidor WebSocket que realizou a simulação da comunicação real da rede da NLT Telecom durante um dia. Assim, foi possível que o programa pudesse realizar a coleta de mensagens e receber toda a comunicação, tratando as mensagens e construindo os fluxos que foram registrados no banco de dados. A simulação foi feita em tempo real e o programa passou as 24 hora, necessárias para receber todos os dados, criando fluxos e fazendo o registro periódico no banco de dados. A interação com banco de dados foi programada e aconteceu em intervalos de 30 minutos onde, após esse espaço de tempo, toda a informação presente na cache era removida para ser armazenada.

A configuração do sistema seguiu o padrão do Netflow [39] que indica que um fluxo não deve passar mais de 30 minutos na cache e, também, define um número máximo de entradas na cache. Um intervalo válido é entre 1024 e 524288 entradas, sendo o padrão de 65536, utilizado no caso dos testes. Há também um tempo máximo que um fluxo pode ficar inativo antes que ele seja removido da cache. Contudo, essa abordagem seria melhor empregada em redes de alta velocidade, o que certamente não é o caso de uma LoRaWAN. Portanto, não foi aplicada a remoção por tempo de inatividade. Caso contrário, a presença desse parâmetro ocasionaria em um número substancialmente alto de fluxos visto que a troca de mensagens pode acontecer em intervalos muito maiores de tempo.

5.2 Análise dos dados

A análise dos resultados foi feita manualmente. Do banco de dados, as informações foram exportadas para uma planilha e, utilizando a ferramenta de filtros do Excel, foi possível estabelecer relações entre os dados e criar gráficos a partir dessas observações.

Essa visualização das relações dos campos de informações proporcionou um entendimento maior sobre aspectos da rede da NLT no recorte de tempo que a empresa disponibilizou.

Finalizando o período de testes, foi observado que o sistema registrou 85991 fluxos no banco de dados. Inicialmente, foi possível notar que uma parte considerável desses fluxos possuía apenas uma mensagem, sendo que apenas 19359 do total de fluxos possuíam 2 mensagens ou mais.

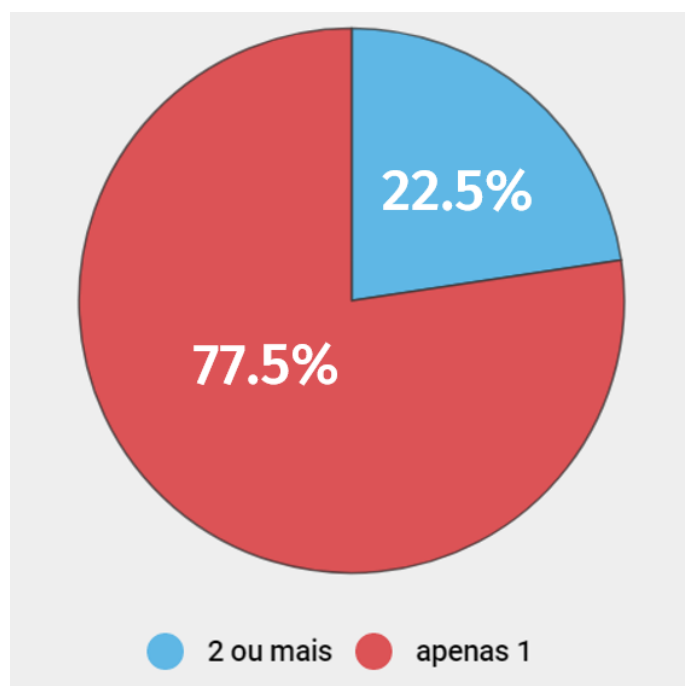


Figura 9: Proporção da quantidade de mensagens em um mesmo fluxo.

Então, foi preciso investigar o que estava acontecendo na rede porque a situação de diversos fluxos com apenas 1 mensagem não seria ideal. A suspeita inicial era de que, de fato, os dispositivos estivessem realizando transmissões esporadicamente e que isso poderia ser uma característica intrínseca dessa rede. Ao observar o campo “timestamp”, foi possível perceber que embora os fluxos fossem apenas de uma mensagem, as ocorrências de um mesmo tipo de fluxo era periódica e seguia intervalos regulares de aproximadamente 2 horas, como pode ser visto na figura seguinte.

O gráfico mostra a quantidade total de todos os tipos de mensagens que envolvem a comunicação com o endereço de dispositivo 8192b10f. No eixo horizontal do gráfico, há apenas os momentos de gravação dos fluxos no banco de dados onde houve alguma transmissão do dispositivo em questão. Assim, foi possível entender a rede da NLT Telecom possuía realmente como característica diversos dispositivos que estavam realizando poucas transmissões naquele período de tempo.

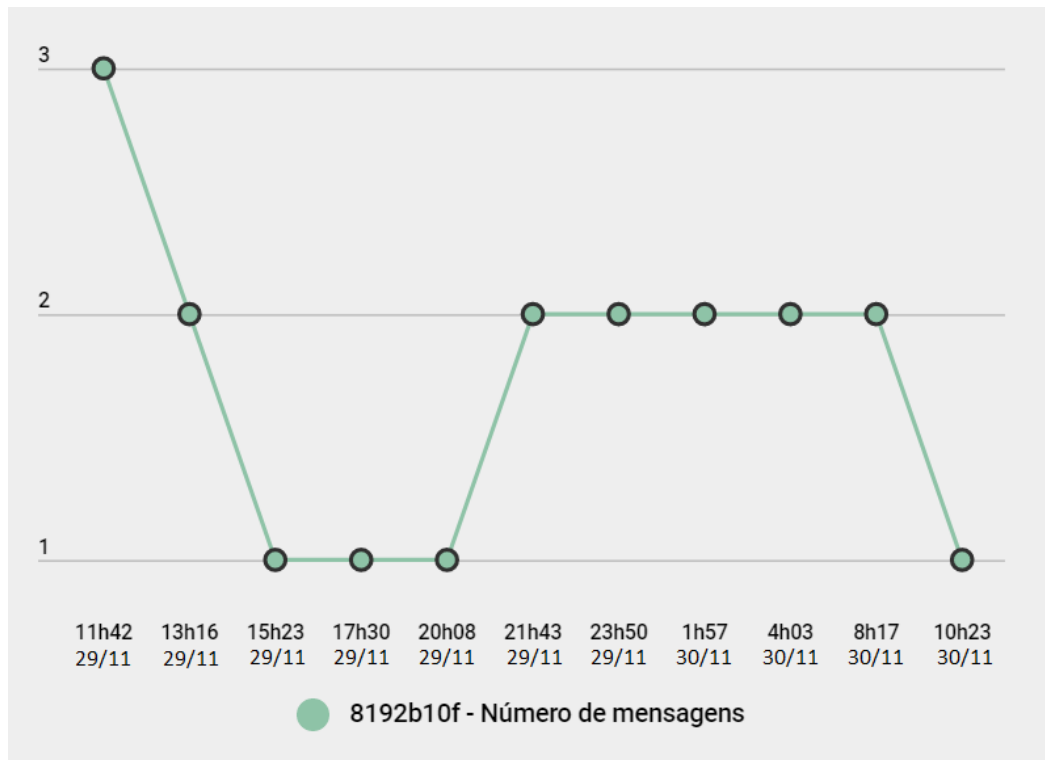


Figura 10: Número de mensagens nos fluxos de 8192b10f.

É possível notar, filtrando os dados, que as transmissões de boa parte das mensagens são separadas por um intervalo de horas e não de minutos. Desse modo, a existência de vários fluxos é reflexo da característica da própria rede. A Figura 9 demonstra essa especificidade presente na rede, com mais de 70% dos fluxos apresentando apenas mensagens unitárias. Nesse sentido, a depender da decisão do operador dessa rede, seria possível ajustar as configurações do programa para que ele esvaziasse os dados armazenados na cache após um intervalo maior de uma ou mais horas. Isso agruparia mais mensagens em menos fluxos, melhorando a visualização das informações sobre os dispositivos na rede.

Ainda, foi possível descobrir a proporção entre os tipos de mensagens na rede. Com especial atenção para os erros, com menos de 0,01% dos casos, se esse dia fosse estabelecido como um parâmetro, um controle de qualidade poderia ser estabelecido a partir de uma simples análise da porcentagem da ocorrência de mensagens de erros em 24 horas. Certamente, também, técnicas mais complexas poderiam ser aplicadas sobre esses dados, tornando esse tipo de controle ainda mais eficiente. De uma maneira geral, seria possível encontrar anomalias na rede.

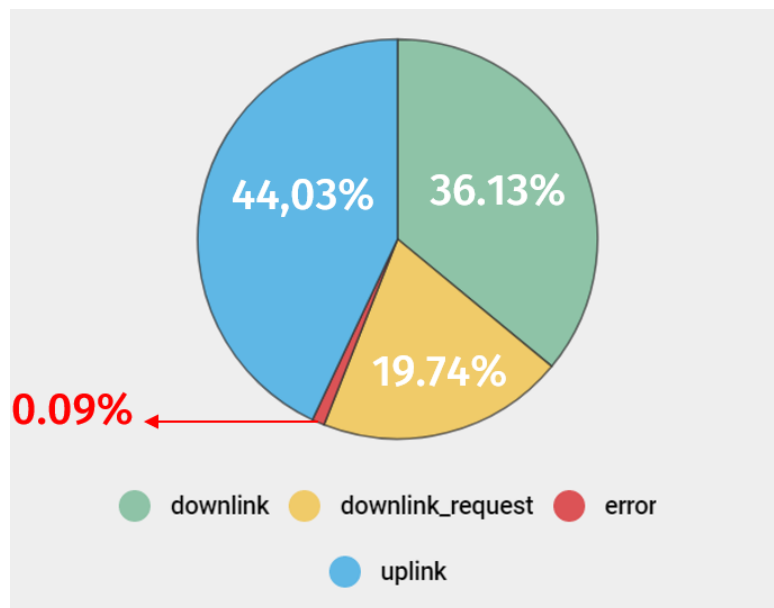


Figura 11: Proporção entre os tipos de mensagens.

Continuando o processo, foram analisados os fluxos que continham o maior número de mensagens. A maioria eram fluxos pertencentes a um mesmo endereço de dispositivo que será investigado posteriormente. Então, para base de comparação, a figura a seguir mostra os cinco fluxos de dispositivos diferentes que tiveram o maior número de mensagens seguindo uma ordem de primeira aparição após a aplicação do filtro de maior para menor número de mensagens.

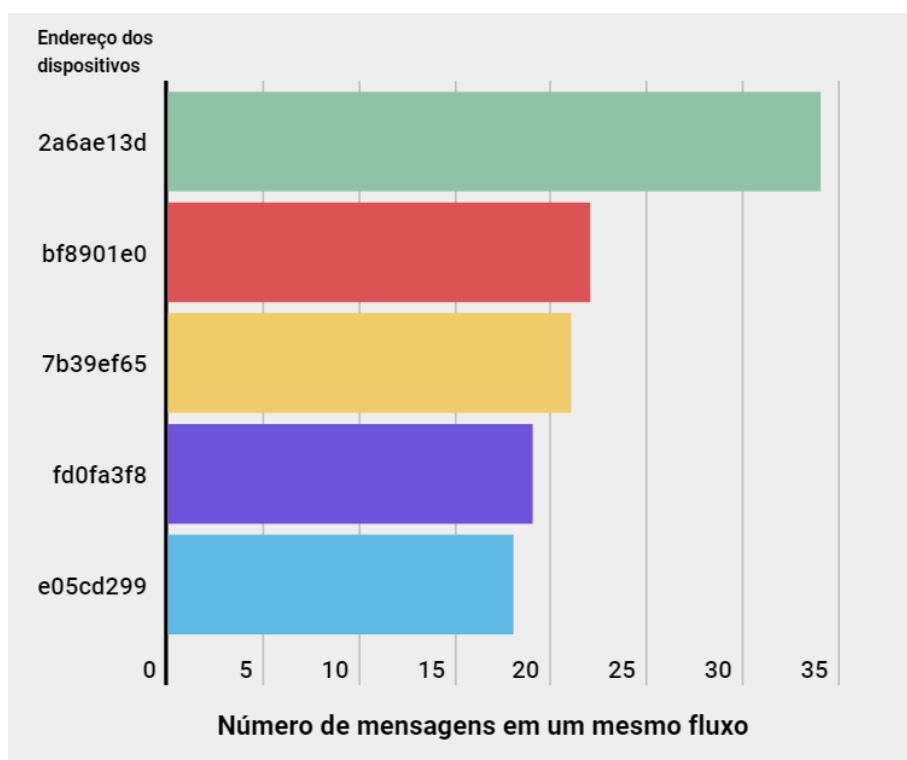


Figura 12: Dispositivos que mandaram mais mensagens de uma vez.

Prosseguindo a análise, agora sobre o dispositivo 2a6ae13d, os dados indicam que absolutamente toda a sua comunicação envolveu apenas um mesmo gateway, sendo ele o 647a5ddda575f0bb. A figura a seguir mostra a porcentagem de mensagens trocadas em fluxos envolvendo o gateway em questão e os demais dispositivos.

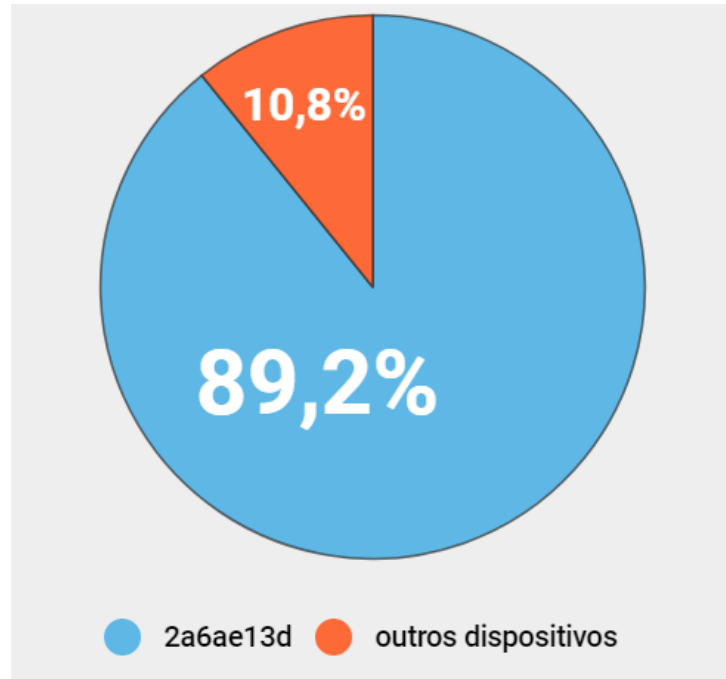


Figura 13: Porcentagem de fluxos envolvendo o gateway 647a5ddda575f0bb.

Os dados sugerem que o dispositivo esteja apenas coberto pelo gateway referido. Esse tipo de observação, acrescentada de um método mais refinado, poderia trazer informações relevantes sobre a disposição dos gateways para melhor suprir todos os dispositivos conectados à rede. Ainda, outras informações envolvendo a cobertura dos dispositivos dos usuários dessa rede poderiam ser encontradas com análises mais refinadas, podendo mostrar, aliado à algoritmos mais avançados, até mesmo a posição ótima para dispor os gateways.

Embora tenha maior quantidade de mensagens através dos fluxos, perde em número de bytes para o dispositivo 4971d4c2ec2df8ba, que tem menos mensagens. Foram analisadas, então, as porcentagens dos tipos de mensagens que esse dispositivo envia em relação ao outro dispositivo há uma diferença entre as quantidade de mensagens do tipo uplink de cada um, o que justifica uma diferença na quantidade de bytes, o dispositivo 4971d4c2ec2df8ba tem maior porcentagem de uplinks. Indo além, o dispositivo em questão é coberto por 18 gateways diferentes, como mostra a figura a seguir, sendo a maior parte da sua comunicação realizada com o gateway 84602e9548138c7e. Ainda, as informações dos fluxos também indicam que o dispositivo mudou de endereço na rede uma vez, assumindo uma nova identidade quando conectado, o que poderia significar em uma

mudança no seu padrão de envio de mensagens ou mesmo de sua localização. De todo forma, dentro os vários gateways, a sua comunicação se concentra principalmente em três, apresentando uma melhor cobertura pelos mesmos, que juntos estão envolvidos em 80% das trocas de mensagens. Os outros 15 gateways estão envolvidos na pequena parcela de 19,7% das trocas de mensagens.

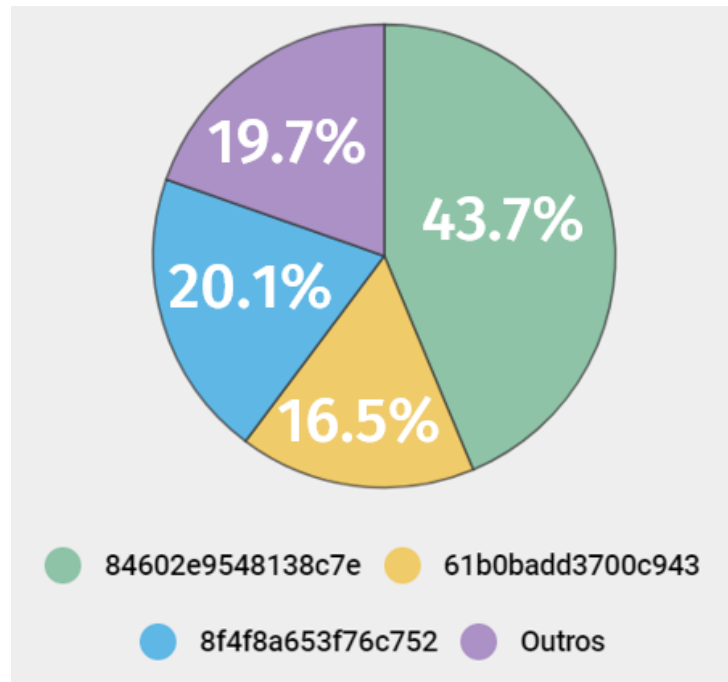


Figura 14: Comunicação do dispositivo 4971d4c2ec2df8ba.

Por fim, os resultados foram muito interessantes do ponto de vista de teste do poder da abordagem de fluxos para o gerenciamento de uma LoRaWAN. Todos os gráficos foram gerados manualmente para o propósito da apresentação dos resultados, mas eles ilustram bem a função que uma *dashborad* poderia fazer automaticamente, quando for desenvolvida futuramente.

A partir da análise dos dados exportados para uma planilha do Excel, foi possível visualizar informações que aprofundaram o entendimento do que está acontecendo na rede. É possível, então, perceber que esse trabalho oferece as bases para emissão de alertas, otimização da cobertura da rede e mesmo previsão de acontecimentos, levando em consideração que o padrão dos dispositivos de cada usuário poderia ser registrado e somado à modelos de inteligência artificial. Tudo isso sustentado pela pedra angular que é a definição dos fluxos em uma LoRaWAN, apresentada nesse trabalho.

6 CONSIDERAÇÕES FINAIS

O desenvolvimento do software foi realizado em dois grandes marcos. O primeiro marco do projeto consistiu na investigação teórica do problema para que, depois de tê-lo bem estruturado, fosse possível galgar uma solução eficiente para o mesmo. O segundo grande marco foi direcionado ao desenvolvimento do sistema propriamente dito, com a adaptação do conceito de fluxos para uma LoRaWAN já definida. O mesmo foi testado a partir de mensagens simuladas de um dispositivo próprio para que essas informações das comunicações pudessem ser interceptadas através da conexão com WebSockets. O processo de desenvolvimento seguiu o cronograma estabelecido e houve um acompanhamento frequente para validar as etapas passadas e ajustar os passos seguintes. Dentre as etapas presentes no desenvolvimento estão:

- Desenvolvimento de um cliente WebSocket que pudesse se conectar a uma API de dados.
- Realizar a extração dos dados das mensagens JSON e realizar seu armazenamento numa estrutura de dados.
- Desenvolver uma estrutura de cache para os dados e esvaziá-la depois de um certo número de mensagens serem recebidas.
- Esvaziar a cache após um intervalo de tempo.
- Limpar os dados da cache após um intervalo de tempo e armazená-los em um banco de dados.

De todos os processos para construção do software, o que demandou mais tempo foi o da realização do controle adequado de tempo e manutenção dos dados da cache em um banco de dados. É muito importante prezar pela integridade dos dados e reunir as ferramentas necessárias para isso ao desenvolver esse software precisou de cautela e pensamento crítico. A escolha da linguagem de programação foi certa, levando em consideração os parâmetros de portabilidade do código em outras plataformas, simplicidade na programação propriamente dita e o auxílio prestado por diferentes módulos do Python. Todos esses elementos contribuíram para uma solução mais direta para o problema.

Assim, os objetivos propostos foram alcançados e a aplicação está funcional, atingindo o objetivo maior de realizar a coleta de dados e gravação dos mesmo em um banco de dados relacional. Diferente, então, dos outros trabalhos citados anteriormente, pôde-se constatar que a abordagem para coleta de dados através de software trouxe informações diferentes das propostas em outros trabalhos e é possível que, com base na especificidade

dos dados que são coletados, *dashboards* possam ser criadas a partir deles. Essas *dashboards* poderiam apresentar, em tempo real, o estado da rede para que operadores pudessem obter um entendimento mais aprofundado sobre o seu estado considerando o que é visto em sistemas como o Netflow. Um imagem conceitual de uma *dashboards*, reunindo alguns dos gráficos presentes nesse trabalho, é apresentada na imagem seguinte. No entanto, essa questão não foi desenvolvida nesse trabalho.

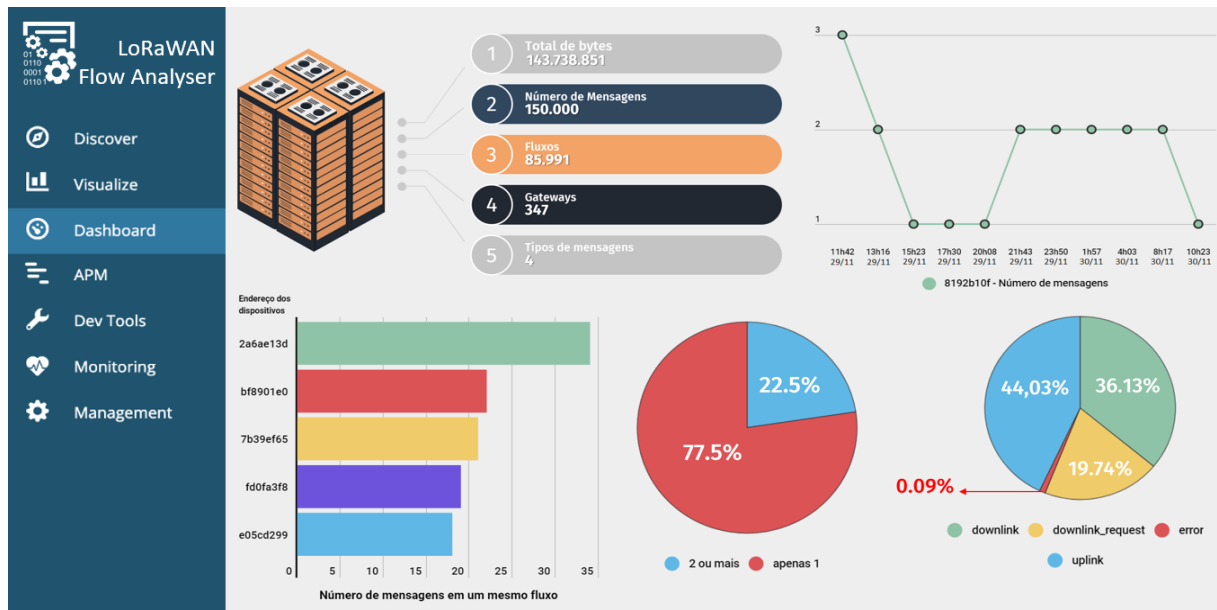


Figura 15: Conceito de *dashboard* com dados apresentados.

Para esse trabalho, todos os conhecimentos adquiridos ao longo do curso, direta ou indiretamente, foram utilizados para resolução do problema levantado. Foram necessários conhecimentos em linguagem de programação Python, o que por si só se relaciona com diferentes aprendizados adquiridos durante a formação, banco de dados, levantamento de requisitos, engenharia de software e comunicação serial.

Os resultados obtidos através da conclusão desse projeto se mostraram satisfatórios uma vez que os objetivos foram atingidos e um conjunto de habilidades e ferramentas adquiridas ao longo da formação puderam ser utilizadas para tal. O tratamento das mensagens e os registros dos dados no banco de dados aconteceu como esperado. Como trabalho futuro, desenvolver uma *dashboards* que pudesse dispor as informações presentes no banco de dados trazendo, por consequência, uma interface gráfica para o usuário, seria o mais desejado para aprimoramento do que já foi realizado.

REFERÊNCIAS

- [1] STANKOVIC, J. A. Research Directions for the Internet of Things. **IEEE Internet of Things Journal**, [s. l.], v. 1, ed. 1, p. 3-9, fev. 2014.
- [2] 5G AMERICAS. 5G Americas LTE progress leading to the 5G massive Internet of Things. **5G Americas**, [s. l.], dez. 2017. E-book (75p.).
- [3] NORTHSTREAM. Massive IoT: different technologies for different needs. **Northstream White Paper**, [s. l.], 29 jun. 2017. E-book (18 p.).
- [4] Gartner Identifies Top 10 Strategic IoT Technologies and Trends. [S. l.], 7 nov. 2018. Disponível em: <<https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends>>. Acesso em: 25 novembro 2021.
- [5] CERWALL, Patrik et al. Ericsson Mobility Report. **Broadband IoT set to overtake 2G and 3G**, [s. l.], p. 11, jun. 2021. E-book (36 p.).
- [6] EVANS, Dave. A Internet das Coisas. **Como a próxima evolução da Internet está mudando tudo**, [s. l.], abr. 2011. E-book (13 p.).
- [7] LIN, J.; YU, W.; ZHANG, N.; YANG, X.; ZHANG, H.; ZHAO, W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. **IEEE Internet of Things Journal**, [s. l.], v. 4, ed. 5, p. 1125-1142, out. 2017.
- [8] BUZZI, S. et al. BUZZI, S. A Survey of Energy-Efficient Techniques for 5G Networks and Challenges Ahead. **IEEE Journal on Selected Areas in Communications**, [s. l.], v. 34, n. 4, p. 697-709, abr. 2016.
- [9] ČOLAKOVIĆ, Alem; HADŽIALIĆ, Mesud. Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. **Computer Networks**, [s. l.], v. 144, p. 17-39, 24 out. 2018.
- [10] SINHA, R. S.; WEI, Y.; HWANG, S. A survey on LPWA technology: LoRa and NB-IoT. **ICT Express**, [s. l.], v. 3, ed. 1, p. 14-21, mar. 2017.
- [11] AL-FUQAHA, A. et al. Internet of Things: A survey on enabling technologies, protocols, and applications. **IEEE Communications Surveys & Tutorials**, [s. l.], v. 17, n. 4, p. 2347-2376, 2015.
- [12] GAZIS, V. A survey of standards for machine-to-machine and the Internet of Things. **IEEE Communications Surveys & Tutorials**, [s. l.], v. 19, n. 1, p. 482-511, 2017.

- [13] CHEN, S. et al. Machine-to-machine communications in ultra-dense networks—A survey. **IEEE Communications Surveys & Tutorials**, [s. l.], v. 19, n. 3, p. 1478-1503, 2017.
- [14] VASHI, S. et al. Internet of Things (IoT): A vision, architectural elements, and security issues. **2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)**, [s. l.], p. 492-496, 2017.
- [15] SIGFOX. [S. l.], 2021. Disponível em: <<http://www.sigfox.com/en/>>. Acesso em: 30 maio 2021.
- [16] LORA Alliance. [S. l.], 2021. Disponível em: <<https://www.lora-alliance.org/>>. Acesso em: 31 maio 2021.
- [17] WANG, H.; FAPOJUWO, A. O. A Survey of Enabling Technologies of Low Power and Long Range Machine-to-Machine Communications. **2017 IEEE Communications Surveys & Tutorials**, [s. l.], v. 19, n. 4, p. 2621-2639, 2017.
- [18] FLORE, D. 3GPP Standards for the Internet-of-Things. **Smart Summit Singapore**, [s. l.], v. 25, nov. 2016. E-book (17 p.).
- [19] ZUNIGA , J. C.; PONSARD, B. Sigfox system description. **LPWAN@ IETF97**, [s. l.], v. 25, nov. 2016.
- [20] VANGELISTA, L.; ZANELLA, A.; ZORZI, M. Long-range IoT technologies: The dawn of LoRaTM. **Proc. Future Access Enablers Ubiquitous Intell. Infrastruct.**, [s. l.], p. 51-58, 2015.
- [21] BOR, M. C. et al. Do LoRa low-power wide-area networks scale?. **Proc. Int. Conf. Model. Anal. Simulat. Wireless Mobile Syst.**, [s. l.], p. 59–67, 2016.
- [22] LORA ALLIANCE. White paper: A technical overview of lora and lorawan. **The LoRa Alliance**, [s. l.], p. 7-11, 2015.
- [23] CROCE, D. et al. Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance. **IEEE Communications Letters**, [s. l.], v. 22, n. 4, p. 796-799, abr. 2018.
- [24] DE CARVALHO SILVA, J. et al. LoRaWAN — A low power WAN protocol for Internet of Things: A review and opportunities. **2017 2nd International Multi-disciplinary Conference on Computer and Energy Science (SpliTech)**, [s. l.], p. 1-6, 2017.

- [25] LO BELLO, L.; MIRABELLA, O.; TORRISI, N. Modelling and evaluating traceability systems in food manufacturing chains. **13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises**, [s. l.], p. 173-179, 2004.
- [26] BELTRAMELLI, L. et al. Interference Modelling in a Multi-Cell LoRa System. **2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)**, [s. l.], p. 1-8, 2018.
- [27] CENTENARO, M. et al. Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios. **IEEE Wireless Communications**, [s. l.], v. 23, n. 5, p. 60-67, out. 2016.
- [28] BOR, M.; VIDLER, J.; ROEDIG, U. LoRa for the Internet of Things. **Embedded Wireless Systems and Networks Conference**, [s. l.], p. 361–366, 2016.
- [29] TESSARO, L. et al. Lightweight Synchronization Algorithm with Self-Calibration for Industrial LORA Sensor Networks. **2018 Workshop on Metrology for Industry 4.0 and IoT**, [s. l.], p. 259-263, 2018.
- [30] LORA ALLIANCE TECHNICAL COMMITTEE. LoRaWAN™ 1.1 Specification. **LoRa Alliance**, [s. l.], out. 2017.
- [31] LORA ALLIANCE TECHNICAL COMMITTEE REGIONAL PARAMETERS WORKGROUP. LoRaWAN Regional Parameters. **LoRa Alliance**, [s. l.], jan. 2018.
- [32] LEONARDI, L. et al. Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications. **IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society**, [s. l.], p. 4141-4146, 2018.
- [33] MUNZ, G.; CARLE, G. Distributed network analysis using TOPAS and wireshark. **NOMS Workshops 2008 - IEEE Network Operations and Management Symposium Workshops**, [s. l.], p. 161-164, 2008.
- [34] AL-SHAER, E. et al. **HONEYSCOPE: IoT Device Protection with Deceptive Network Views**, [s. l.], 1. ed. Autonomous Cyber Deception: Springer International Publishing, 2019. p. 167-181, ISBN 9783030021108, 3030021106. E-book (235 p.).
- [35] CLAISE, B.; AITKEN, P.; TRAMMELL, B. **Specification of the IP Flow Information Export (IPFIX) protocol for the exchange of flow information**. RFC 7011 (Internet Standard), Internet Engineering Task Force, 2013. Disponível em: <http://www.ietf.org/rfc/rfc7011.txt>. Acesso em: 29 maio 2021.

- [36] ROUX, J. et al. Toward an Intrusion Detection Approach for IoT Based on Radio Communications Profiling. **2017 13th European Dependable Computing Conference (EDCC)**, [s. l.], p. 147-150, 2017.
- [37] MEIDAN, Y. et al. ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis. **Proceedings of the symposium on applied computing**, [s. l.], p. 506-509, abr. 2017.
- [38] KIM, M. S.; WON, Y. J.; HONG, J. W. Characteristic analysis of internet traffic from the perspective of flows. **Comput. Commun.** **29(10)**, [s. l.], p. 1639-1652, 2006.
- [39] INTRODUCTION to Cisco IOS NetFlow - A Technical Overview. [S. l.], 29 maio 2012. Disponível em: <https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html>. Acesso em: 2 jun. 2021.
- [40] OUSAT, B.; GHADERI, M. LoRa Network Planning: Gateway Placement and Device Configuration. **2019 IEEE International Congress on Internet of Things (ICIOT)**, [s. l.], p. 25-32, 2019.
- [41] SEMTECH CORPORATION. An1200.22. LoRa Modulation Basics. **AN SEMTECH and Modulation Basics**, [s. l.], 2015.
- [42] WIXTED, A. J. et al. Evaluation of LoRa and LoRaWAN for wireless sensor networks. **2016 IEEE SENSORS**, [s. l.], p. 1-3, 2016.
- [43] LIANDO, J. C. et al. Known and unknown facts of LoRa: Experiences from a large-scale measurement study. **ACM Trans. Sensor Netw.**, [s. l.], v. 15, n. 2, p. 16, 2019.
- [44] NAOUI, S.; ELHDHILI, M. E.; SAIDANE, L. A. Enhancing the security of the IoT LoRaWAN architecture. **2016 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)**, [s. l.], p. 1-7, 2016.
- [45] ADELANTADO, F. et al. Understanding the limits of lorawan. **IEEE Communications Magazine**, [s. l.], v. 55, n. 9, p. 34-40, set. 2017.
- [46] LYU, J.; YU, D.; FU, L. Achieving Max-Min Throughput in LoRa Networks. **2020 International Conference on Computing, Networking and Communications (ICNC)**, [s. l.], p. 471-476, 2020.

- [47] TORIL , M.; WILLE, V. Optimization of the Assignment of Base Stations to Base Station Controllers in GERAN. **IEEE Communications Letters**, [s. l.], v. 12, n. 6, p. 477-479, jun. 2008.
- [48] MATNI, N. et al. Optimal Gateway Placement Based on Fuzzy C-Means for Low Power Wide Area Networks. **2019 IEEE Latin-American Conference on Communications (LATINCOM)**, [s. l.], p. 1-6, 2019.
- [49] HUH , H.; KIM, J. Y. LoRa-based Mesh Network for IoT Applications. **2019 IEEE 5th World Forum on Internet of Things (WF-IoT)**, [s. l.], p. 524-527, 2019.
- [50] POLONELLI, T. et al. Slotted aloha on lorawan-design, analysis, and deployment. **Sensors**, [s. l.], 2019.
- [51] HUANG, A. et al. A Practical Marine Wireless Sensor Network Monitoring System Based on LoRa and MQTT. **2019 IEEE 2nd International Conference on Electronics Technology (ICET)**, [s. l.], p. 330-334, 2019.
- [52] MAGNO, M. et al. Wulora: An energy efficient iot end-node for energy harvesting and heterogeneous communication. **Design, Automation Test in Europe Conference Exhibition (DATE)**, [s. l.], p. 1528-1533, 17 mar. 2017.
- [53] PETÄJÄJÄRVI, J. et al. Performance of a low-power wide-area network based on lora technology: Doppler robustness, scalability, and coverage. **International Journal of Distributed Sensor Networks**, [s. l.], 17 mar. 2017.
- [54] POLONELLI, T. et al. An accurate low-cost Crackmeter with LoRaWAN communication and energy harvesting capability. **2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)**, [s. l.], p. 671-676, 2018.
- [55] MILLS, C.; HIRSH, D.; RUTH, G. Internet accounting: Background. **Internet Engineering Task Force**, [s. l.], nov. 1991. Disponível em: <http://www.ietf.org/rfc/rfc1272.txt>. Acesso em: 3 jun. 2021.
- [56] BROWNLEE, N. Flow-based measurement: IPFIX development and deployment. **IEICE Trans. Commun.**, [s. l.], v. 94, n. 8, p. 2190-2198, ago. 2011.
- [57] CLAFFY, K. C. et al. A parameterizable methodology for Internet traffic flow profiling. **IEEE Journal on Selected Areas in Communications**, [s. l.], v. 13, n. 8, p. 1481-1494, out. 1995.
- [58] BROWNLEE, N. RTFM: Applicability statement. **Internet Engineering Task Force**, [s. l.], out. 1999. Disponível em: <https://www.ietf.org/rfc/rfc2721.txt>. Acesso em: 3 jun. 2021.

- [59] CISCO. Cisco Catalyst 6500 Architecture. **Cisco Systems, Inc.**, [s. l.], jan. 2013. E-book (28 p.).
- [60] CISCO. NetFlow Services Solutions Guide. **Cisco Systems, Inc.**, [s. l.], jan. 2007. E-book (72 p.).
- [61] CLAISE, B. Cisco systems NetFlow services export version 9. **Internet Engineering Task Force**, [s. l.], out. 2004. Disponível em: <http://www.ietf.org/rfc/rfc3954.txt>. Acesso em: 3 jun. 2021.
- [62] CISCO. Cisco IOS Flexible NetFlow. **Cisco Systems, Inc.**, [s. l.], dez. 2008. E-book (126 p.).
- [63] DERI, L. et al. Increasing data center network visibility with cisco NetFlow-Lite. **2011 7th International Conference on Network and Service Management**, [s. l.], p. 1-6, 2011.
- [64] IETF. IP Flow Information Export (ipfix). **Internet Engineering Task Force**, [s. l.], abr. 2013. Disponível em: <http://datatracker.ietf.org/wg/ipfix/charter/>. Acesso em: 8 jun. 2021.
- [65] QUITTEK, J. et al. Requirements for IP Flow Information Export. **Internet Engineering Task Force**, [s. l.], out. 2004. Disponível em: <https://www.ietf.org/rfc/rfc3917.txt>. Acesso em: 8 jun. 2021.
- [66] LEINEN, S. Evaluation of candidate protocols for IP Flow Information Export. **Internet Engineering Task Force**, [s. l.], out. 2004. Disponível em: <http://www.ietf.org/rfc/rfc3955.txt>. Acesso em: 8 jun. 2021.
- [67] STEINBERGER, J. et al. Anomaly Detection and Mitigation at Internet Scale: A Survey. **Lecture notes in computer science**, [s. l.], v. 7943, p. 49–60, 2013.
- [68] STEINBERGER, J. et al. Anomaly Detection and Mitigation at Internet Scale: A Survey. **Lecture notes in computer science**, [s. l.], v. 7943, p. 49–60, 2013.
- [69] EUROPEAN PARLIAMENT & COUNCIL. Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the Retention of Data Generated or Processed in Connection With the Provision of Publicly. abr. 2006. Disponível em: <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:105:0054:0063:EN:PDF>. Acesso em: 10 jun. 2021.
- [70] JOHN, W.; TAFVELIN, S.; OLOVSSON, T. Passive Internet measurement: Overview and guidelines based on experiences. **Computer Communications**, [s. l.], v. 33, ed. 5, p. 533-550, 15 mar. 2010.

- [71] SADASIVAN, G. et al. Architecture for IP Flow Information Export. **Internet Engineering Task Force**, [s. l.], 2009. Disponível em: <http://www.ietf.org/rfc/rfc5470.txt>. Acesso em: 10 jun. 2021.
- [72] GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. 176 p. ISBN 85-224-3169-8.
- [73] LoRa Alliance Technical Committee. **LoRaWAN™ 1.1 Specification**. LoRa Alliance, October 2017.
- [74] ADELANTADO, F. et al. Understanding the Limits of LoRaWAN. **IEEE Communications Magazine**, [s. l.], v. 55, ed. 9, p. 34–40, 2017.
- [75] BOR, M. et. al. Do LoRa Low-Power Wide-Area Networks Scale? **MSWiM 2016**, [s. l.], p. 59–67, 2016.
- [76] AUGUSTIN, A. et. al. A Study of LoRa: Long Range: Low Power Networks for the Internet of Things. **Sensors**, [s. l.], v. 16, ed. 9, 2016.
- [77] LUVISOTTO, M. et. al. On the Use of LoRaWAN for Indoor Industrial IoT Applications. **Wireless Communications and Mobile Computing**, [s. l.], v. 2018, p. 1–11, mai. 2018.
- [78] BOR, M; ROEDIG, U. LoRa Transmission Parameter Selection. **2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)**, [s. l.], p. 27–34, jun. 2017.
- [79] COSTA, C.; CENTENARO, M.; RIGGIO, R. LoMM: a Monitoring and Management Platform for LoRaWAN Experimentation. **2020 IEEE International Conference on Communications Workshops (ICC Workshops)**, [s. l.], p.1-6, 2020.
- [80] YOUSUF, A. M.; ROCHESTER, E. M.; GHADERI, M. A low-cost LoRaWAN test-bed for IoT: Implementation and measurements. **2018 IEEE 4th World Forum on Internet of Things (WF-IoT)**, [s. l.], p. 361-366, 2018.
- [81] RODRIGUEZ, I. et al. The Gigantium Smart City Living Lab: A Multi-Arena LoRa-based Testbed. **2018 15th International Symposium on Wireless Communication Systems (ISWCS)**, [s. l.], p. 1-6, 2018.
- [82] MAGNO, M. et al. WULoRa: An energy efficient IoT end-node for energy harvesting and heterogeneous communication. **Design, Automation Test in Europe Conference Exhibition (DATE) 2017**, [s. l.], 2017.
- [83] Logstash Netflow Module. Disponível em: <https://www.elastic.co/guide/en/logstash/7.16/netflow-module.html>. Acesso em: 4 nov. 2021.

- [84] Quadrar Tecnologia em Serviços de Informática Ltda. LoRaONE Long Range LoRaWAN Network Technology Low-Power Transceiver Module User Manual V1.3. Disponível em: <<https://www.quadrar.com.br/files/ugd/8fb6e0dc2141e8cc954aa0b193b9e049ae4577.pdf>>. Acesso em: 4 nov. 2021.
- [85] Everynet. Company|Everynet. Disponível em: <<https://www.everynet.com/company>>. Acesso em: 10 jun. 2021.
- [86] Everynet. Everynet NS Docs. Disponível em: <<https://ns.docs.everynet.io/data-api/messages-and-types/>>. Acesso em: 12 jun. 2021.
- [87] ARAS, E.; RAMACHANDRAN, G. S.; LAWRENCE, P.; HUGHES, D. Exploring The Security Vulnerabilities of LoRa. **2017 3rd IEEE International Conference on Cybernetics (CYBCONF)**, [s. l.], p. 1-6, 2017.