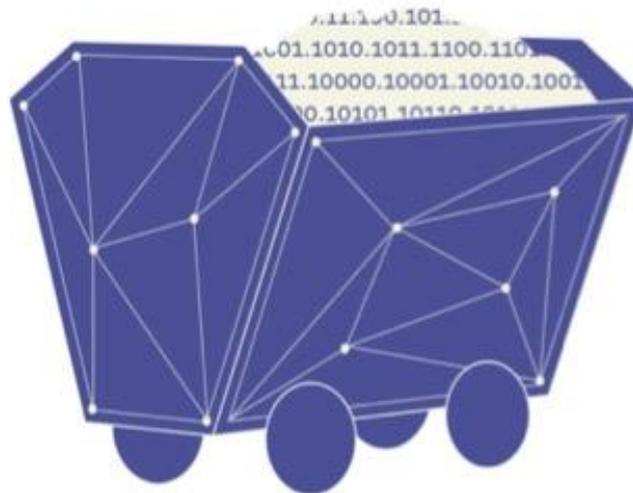


MINICURSOS

ERRC 2021
on-line



**LIVRO DOS MINICURSOS DA ESCOLA REGIONAL DE REDES DE
COMPUTADORES 2021**

ORGANIZAÇÃO E EDIÇÃO:
CARLOS MORATELLI
FÁBIO DINIZ ROSSI
GUILHERME DA CUNHA RODRIGUES
ROBERTO IRAJÁ TAVARES DA COSTA FILHO
VINÍCIUS TAVARES GUIMARÃES

CARLOS MORATELLI
FÁBIO DINIZ ROSSI
GUILHERME DA CUNHA RODRIGUES
ROBERTO IRAJÁ TAVARES DA COSTA FILHO
VINÍCIUS TAVARES GUIMARÃES

**MINICURSOS DA XIX ESCOLA REGIONAL DE REDES DE
COMPUTADORES (ERRC 2021)**

Porto Alegre
Sociedade Brasileira de Computação

2021

Dados Internacionais de Catalogação na Publicação (CIP)

E74 Escola Regional de Redes de Computadores (19. : 27-29 out. 2021 : Online)

Minicursos da XIX Escola Regional de Redes de Computadores (ERRC 2021) [recurso eletrônico] / organização: Carlos Moratelli ... [et al.]. Dados eletrônicos. – Porto Alegre : Sociedade Brasileira de Computação, 2021.

63 p. ; PDF ; 2.50MB

Evento online

ISBN 978-65-87003-98-6 (e-book)

1. Ciência da Computação. 2. Redes de computadores. I. Moratelli, Carlos. II. Rossi, Fábio Diniz. III. Rodrigues, Guilherme da Cunha. IV. Costa Filho, Roberto Irajá Tavares da. V. Guimarães, Vinícius Tavares. VI. Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense. VII. Sociedade Brasileira de Computação. VIII. Título.

CDU 004.7

Ficha catalográfica elaborada por Jéssica Paola Macedo Müller – CRB-10/2662

Biblioteca Digital da SBC – SBC OpenLib

Índice para catálogo sistemático:

1. Comunicação entre computadores : Redes de computadores 004.7

MENSAGEM DA COORDENAÇÃO DOS MINICURSOS

A XIX Escola Regional de Redes de Computadores (ERRC 2021) é um evento anual promovido pela Sociedade Brasileira de Computação (SBC). A escola foi realizada entre os dias 27 e 29 de outubro de 2021, sendo organizada de forma online pelo Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense (IFSul) - Câmpus Charqueadas, com a colaboração de diversas outras instituições de ensino, pesquisa e extensão.

Dentre os diversos objetivos da ERRC, um destes consiste em qualificar profissionais da região sul nas diversas áreas de Redes de Computadores. Com este intuito, todos os anos são selecionados minicursos em tópicos atuais de interesse da comunidade. Não diferente, neste ano, foram selecionados dois minicursos, os quais foram incluídos como capítulos para este volume. Os minicursos aqui representados, apresentam tópicos de ponta da área de Redes de Computadores, os quais irão, certamente, fortalecer e colaborar com a aprendizagem dos participantes do evento.

O primeiro minicurso apresentou conceitos necessários ao entendimento das atuais arquiteturas de serviços e computação em borda nas redes 5G, voltados para o desenvolvimento de redes virtuais privadas sobre o Core 5G. O segundo curso apresentou uma introdução teórica e prática sobre LoraWAN, um padrão que de rede de baixa potência e longo alcance que está sendo largamente utilizado para fornecer conectividade em paradigmas atuais, como internet das coisas e redes veiculares.

Os coordenadores dos minicursos agradecem aos autores, por compartilharem seus conhecimentos através da submissão de minicursos de alto nível para esta edição da escola, aos coordenadores e organizadores da ERRC, pelo apoio dado na seleção dos minicursos e realização do evento.

Carlos Roberto Moratelli (UFSC) e Fábio Diniz Rossi (IFFAR)
Coordenadores dos Minicursos da ERRC 2021

Índice

1 Arquitetura de Serviços e Computação de Borda nas redes 5G para o desenvolvimento de redes virtuais privadas sobre o Core 5G.....	6
2 Fundamentos de LoraWAN – Teoria e Prática.....	39

1 Arquitetura de Serviços e Computação de Borda nas redes 5G para o desenvolvimento de redes virtuais privadas sobre o CORE 5G

Adão Boava, Christian Mailer e Dener Kraus

Resumo: Com a necessidade de reestruturação das arquiteturas das redes de comunicação para atender aos novos requisitos das aplicações, novas propostas de arquiteturas de redes têm surgido com o objetivo de tornar viável economicamente e tecnicamente as implementações para atender as aplicações. Inicialmente o minicurso irá apresentar os requisitos estratégicos para o desenvolvimento do 5G, conforme a recomendação do ITU, fazendo uma relação entre requisitos, cenários e aplicações. Essa proposta de minicurso tem o objetivo de apresentar a utilização das redes móveis 5G formada de acesso e core segundo o padrão definidos pelo 3GPP. Serão apresentadas a arquitetura baseada em serviços que separa os planos de dados e controle, e a arquitetura de computação em borda que podem ser utilizadas para criar redes privadas para as indústrias.

O minicurso proposto possui características teórica e prática, os autores apresentarão a implementação da arquitetura de serviços e computação em borda através do código aberto (projeto free5GC).

Também será apresentado pelos autores a evolução das redes de acesso e core das redes móveis 5G. No acesso será dada ênfase às técnicas MIMO e Beamforming, enquanto no CORE serão enfatizadas a Virtualização e o Fatiamento da rede.

1. INTRODUÇÃO

As redes de comunicações estão diante de uma mudança de um cenário que no primórdio era focado somente em voz para um cenário que deve suportar trilhões de dispositivos conectados transmitidos dados, voz e vídeo.

Em função desse novo cenário, a maneira de prestar serviços na área de telecomunicações móveis está passando por uma grande transformação. Esses serviços possuem requisitos que exigem novas arquiteturas de telecomunicação para garantir a qualidade dos serviços prestados. O projeto, o desenvolvimento e a implantação dessas arquiteturas precisam evoluir para atender de forma rápida e flexível esses requisitos rigorosos de comunicação. Para atender a essas exigências, existe uma grande expectativa nas redes móveis de quinta geração (5G). Com o 5G espera-se ser possível, por exemplo, a prestação de serviços para as indústrias inteligentes (Indústria 4.0), telemedicina, cidades inteligentes, fazendas inteligentes, Blockchain, veículos autônomos e a Internet das Coisas (IoT - Internet of Things) [1,3].

Além dos aspectos tecnológicos, as redes 5G também introduzem mudanças no modelo de negócios das empresas. Tradicionalmente, operadores de sistemas de comunicação móvel sem fio se focavam em usuários finais como principais fontes de receita. Em redes 5G, a intenção é ampliar (ou até migrar) o foco para ter **indústrias como principais clientes** [4, 5, 6]. Para dar suporte a essa ampliação do modelo de negócio das empresas de telecomunicações, as redes 5G devem atender aos pré-requisitos de quatro principais cenários, como visto na Figura 1: comunicações massivas entre máquinas (mMTC, massive machine type communications) relacionadas as comunicações V2X, logística, cidades inteligentes (smart cities) e Big Data; comunicações móveis de altas taxas (eMBB, extreme mobile broadband), cujas aplicações são acesso multimídia em alta resolução, realidade aumentada, realidade virtual e transferência massiva de dados; comunicações ultra confiáveis de baixa latência (URLL, ultra-reliable low latency) ligadas à indústria 4.0, aplicações de caráter crítico (healthcare, ambientes hostis, missões de resgate, etc.), veículos autônomos e controle a longa distância; e comunicações de longo alcance relacionadas a cobertura de áreas remotas e/ou não conectadas. As aplicações mMTC demandarão principalmente capacidade de rede para gerenciar conexões massivas de dispositivos, atingindo milhares de dispositivos por setor. Já as comunicações eMBB necessitarão de sistemas com maior capacidade, com taxas de transferência de dados de 10 Gbps por setor. Enquanto as comunicações URLL exigirão robustez e baixa latência, com latência fim-a-fim inferior a 1ms [7].

A fim de suportar as tendências do mercado e para se harmonizar com a explosão de tráfego móvel, os seguintes aspectos devem ser considerados nas redes 5G de acordo com o “REPORT ITU-R M.232” [29, 30]:

- a. Vazão Média: A vazão média das redes móveis deve aumentar significativamente para suportar o crescimento mais que exponencial do tráfego gerado pelas novas aplicações;
- b. Experiência do Usuário: Na nova rede 5G deve ser pelo menos mantida independentemente da localização do usuário e condições de tráfego da rede;
- c. Escalabilidade: O número de terminais móveis por estação radiobase aumentará significativamente devido aos novos serviços como, IoT e M2M;
- d. Latência: A qualidade das experiências dos usuários pode ser muito melhorada, reduzindo a latência no estabelecimento da conexão e entrega dos pacotes;
- e. Eficiência Energética: baixo consumo de energia tanto para rede como para os dispositivos móveis;
- f. Eficiência dos custos: necessidade de diminuir o CAPEX e OPEX para motivar a expansão e melhoria das redes além de reduzir o custo de uma assinatura móvel;
- g. Flexibilidade: topologia flexível juntamente com serviços sem fio complexos e integração com diferentes tecnologias de acesso via rádio (RAT);
- h. Serviços não Tradicionais: novos serviços esperados como vídeo móvel em alta definição, comunicação M2M, serviços baseados na localização, computação em nuvem;
- i. Espectro: mais espectro pode ser necessário para acomodação do grande aumento do tráfego móvel. Espera-se arranjos de frequência e compartilhamento com outros serviços. A harmonização do espectro pode reduzir o custo dos recursos de tecnologia.

De acordo com a recomendação ITU-R M.2083-0 [11], oito parâmetros são considerados REQUISITOS fundamentais para o desenvolvimento de 5G: Taxa de dados de pico, Taxa de dados experimentada pelo usuário, Latência, Mobilidade, Densidade de conexão, Eficiência energética, Eficiência espectral e Capacidade de tráfego por área.

A relação entre os cenários (eMBB, URLLC e mMTC) e os oito requisitos acima deu origem a figura 1 abaixo.

Figura 1: Cenário x Requisitos.



Fonte: [9]

As primeiras redes 5G atenderão aos serviços eMBB e as suas especificações iniciais, estão contidas na Release 15 do grupo 3GPP (3rd Generation Partnership Project). A definição das faixas de frequências destinadas à operação das redes 5G integra os requisitos estabelecidos pelo padrão IMT-2020. A previsão é que diferentes sistemas 5G operem em faixas de frequências distintas para atender as necessidades particulares dos diversos serviços. O 3GPP especificou duas potenciais faixas de frequências para o desenvolvimento e operação de sistemas 5G: 450 a 6.000 MHz (FR1) e 24,25 a 52,60 GHz (FR2). As frequências baixas apresentam melhores condições de propagação e baixa capacidade, por isso são destinadas às aplicações IoT e cobertura de grandes áreas e regiões remotas. A faixa de frequências médias adequa-se melhor a cobertura de centros urbanos, provendo maior capacidade. A faixa de frequências em ondas milimétricas possui limitações em cobertura devido à intensificação dos fenômenos de atenuação, entretanto apresenta maior disponibilidade de espectro. Para mitigar os efeitos de atenuação nessa faixa de frequência, os sistemas 5G dispõem de algumas tecnologias como redes de cobertura ultradensa, arranjos de antenas com alto ganho com reconfiguração de feixe e sistemas MIMO [7].

As redes de cobertura ultradensas utilizam um número maior de células com tamanhos reduzidos (micro, pico e femtocélulas). Para sistemas 5G em ondas milimétricas, células com menor raio de cobertura reduzem a atenuação no espaço livre, compensando o aumento na atenuação total derivado de outros fenômenos. Outra forma de compensar a maior atenuação em ondas milimétricas é utilizar arranjos

de antenas com alto ganho. Entretanto, a mobilidade do usuário introduz a necessidade de uso de arranjos de antenas. Assim, torna-se possível direcionar espacialmente o lobo principal do arranjo para uma região específica de acordo com a concentração espacial de usuários. Sistemas MIMO se beneficiam do fenômeno de propagação por multipercursos e do uso da diversidade de antenas em seus rádios transmissores e/ou receptores como forma de mitigar o desvanecimento oferecido pelo canal. Os sistemas MIMO e arranjos reconfiguráveis podem ser utilizados em conjunto para prover maior capacidade e cobertura, minimizar interferências e reduzir de forma eficiente o custo e o consumo de energia em redes 5G [7].

Com o objetivo de subsidiar a Anatel com dados práticos e reais, a empresa Weg em parceria com a Anatel, ABDI e Nokia desenvolveram testes pilotos em um laboratório que foi criado dentro do ambiente da Weg, esse laboratório chama-se “OPEN LAB 5G WEG-V2COM”. Os resultados serão utilizados pela Anatel para regulamentar as redes privadas sobre 5G para uso industrial no Brasil. O objetivo é testar a conectividade de diversos dispositivos IoT à rede 5G, contribuindo para o desenvolvimento de soluções economicamente viáveis para a indústria utilizando a tecnologia 5G [18,19].

Foram realizados testes avaliando o desempenho e a convivência de dispositivos e antenas com a tecnologia 5G em ambiente real, para reunir informações sobre faixas de frequência, latência, potência e outras características necessárias às aplicações industriais e que irá contribuir para a Consulta Pública realizada pela Anatel. Os primeiros resultados já mostram que a tecnologia 5G proporciona níveis de segurança, qualidade de tráfego, estabilidade e alta velocidade, superiores aos oferecidos anteriormente pelo 3G e o 4G[18,19].

Todos os testes aconteceram com a utilização de mais de uma rede 5G ao mesmo tempo, uma rede provida por operadora de telecom e outra privada com infraestrutura local, a fim de levantar dados e informações à Anatel e apoiar no processo de definição dos requisitos e condições de uso de faixas de frequência, para a regulação e outorga das redes privadas para uso industrial no Brasil [18,19].

Para a Anatel, os resultados práticos até esse momento contribuem para a avaliação da potencialidade do 5G, especialmente em relação às características da operação, observados os limites de irradiação de potência definidos em regulamentação[18,19].

De acordo com a ABDI, esses testes não só vão ajudar a Anatel com a regulamentação, mas vão contribuir para que as empresas avaliem a viabilidade econômica do uso de 5G em redes privadas. Além disso, os testes vão gerar dados para novos modelos de negócios. Com esse projeto, a ABDI cumpre sua missão de tornar acessível ao setor produtivo às tecnologias existentes, em busca da maior eficiência, produtividade e competitividade”. No final do documento são apresentados alguns resultados alcançados no Weg [18,19].

O projeto Free5GC é um projeto baseado em código aberto que torna a rede 5G disponível para todos. Esse projeto implementa o núcleo da rede 5G (5GC) de acordo com as especificações do lançamento 15 3GPP e suas atualizações. O objetivo do minicurso é apresentar a teoria básica da rede 5G através das implementações de funções de rede como serviço e computação de borda para formação de rede privada, através do uso prático do Free5GC.

Todo o processo de instalação, configuração, preparação do ambiente e o desenvolvimento de software no contexto do Free5GC é apresentado, descrito e disponibilizado para os participantes do minicurso.

O documento está organizado da seguinte maneira. A Seção 1 apresenta a introdução ao 5G; a Seção 2 apresenta os fundamentos das redes 5G que é a separação entre o plano de controle e usuário; a Seção 3 apresenta a rede de acesso 5G com ênfase no NR 5G; a Seção 4 apresenta o CORE 5G com destaque para as funções de redes; a Seção 5 apresenta o desenvolvimento da arquitetura de serviço e computação em borda com o free5GC; e finalmente a Seção 6 apresenta a conclusão do trabalho

2. FUNDAMENTOS

Com o decorrer dos tempos houve várias alterações e a introdução de diversas tecnologias nas arquiteturas das redes celulares móveis. No entanto, a maior parte dos conceitos introduzidos no início das redes celulares ainda continuam em suas arquiteturas, tais como, handoff, conceito de células, reuso de frequências, etc.

Com o objetivo de melhorar a flexibilidade, a escalabilidade e o gerenciamento da infraestrutura do 5G é necessário considerar a separação entre o plano do usuário e o plano de controle, e a redefinição dos limites entre o CORE de rede (CN) e a rede de acesso RAN. Requisitos como latência e outros atributos de desempenho tornam necessária a movimentação de algumas funções do CN para extremidade da rede, assim como funções tipicamente da rede de acesso estejam mais centralizadas [8].

Seguindo os conceitos apresentados pelo órgão regulador 3rd Generation Partnership Project (3GPP), para promover a facilidade de dimensionar, implementar e adaptar a rede, o 3GPP por meio do Release 14 (3GPP, 2018) introduziu a estratégia da utilização de uma arquitetura plana com a divisão entre o plano de controle - Control Plane (PC) e o plano de usuário- User Plane (PU) criando o conceito de Control and User Plane Separation (CUPS) aplicada na rede core do 4G, chamada de Evolved PacketCore (EPC). Essa independência fornece um suporte mais eficiente ao aumento de tráfego, pois permite uma ampliação nos elementos do PU sem a necessidade de ampliar elementos do PC [8].

3. REDE DE ACESSO 5G

Para atender os diversos cenários de serviços previstos na rede 5G, é necessário que a rede de acesso se adapte a uma grande gama de serviços com características diversas, como diferentes tipos de canais e diversas condições de propagação. Uma nova interface de rádio e de arquitetura de rede de acesso foram desenvolvidas a partir do zero, levando em conta os **requisitos** e buscando as melhores tecnologias que estarão disponíveis quando a implantação do 5G for iniciada [10].

3.1. NOVO RÁDIO

A 3GPP definiu que a tecnologia de rádio 5G será chamada de New Radio (NR) e que contará com diversas novidades comparadas às suas antecessoras, com suporte a recursos como massive Multiple Input Multiple Output (mMIMO), beamforming, novas bandas de frequência, novas técnicas de modulação de canal e Carrier Aggregation (CA) [2,11,13].

No 5G, as bandas de frequência suportadas variam de 450 MHz até 6 GHz (classificada como Frequency Range 1, FR1, mid/low band ou sub 6 GHz bands) e de 24,250 GHz até 52,600 GHz (classificadas como Frequency Range 2, FR2, high band ou above 6 GHz bands) [11,13]. A grande inovação são as altas frequências, também chamadas de millimeter wave (mmwave), capazes de transferir uma grande quantidade de dados com uma latência muito baixa. No entanto, quanto maior a frequência de transmissão, menor é o alcance das ondas, ficando suscetíveis a perda de sinal causadas por objetos comuns, como portas e janelas, e até mesmo pela presença de indivíduos ou animais entre o emissor e o receptor. Com o intuito de se aumentar a eficácia da transmissão, tecnologias como beamforming e MIMO podem ser empregadas [2,11].

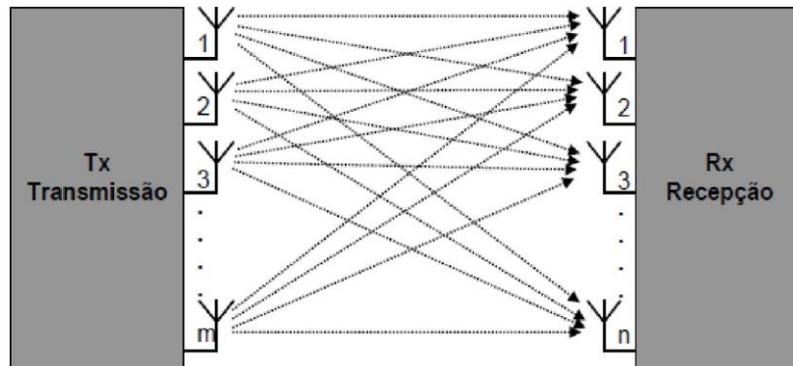
A utilização de frequências acima de 10 GHz permite a obtenção de altas taxas de transmissão já que oferece uma banda contígua cerca de 10 vezes maior que a banda atual disponível. Entretanto, essas frequências altas, especialmente as frequências acima de 30 GHz, fornecem uma cobertura reduzida. As frequências baixas e altas podem ser utilizadas no modo de agregação de portadora, aumentando tanto a cobertura quanto a taxa de transmissão. Dessa formação possível suportar uma grande variedade de serviços, seja para uma cobertura IoT ou para serviços de alta capacidade de banda larga por meio de uma cobertura de múltiplas camadas [10].

3.2. MIMO MASSIVO

Para aumentar a eficiência de sistemas sem fio móveis ou fixos, foi desenvolvido ao longo dos últimos anos o conceito de múltiplas interfaces aéreas, conhecido como MIMO (Multiple Input Multiple Output). O MIMO consiste basicamente no uso simultâneo de várias antenas ou interfaces de RF como é mostrado na figura 2. O

sistema MIMO da figura é constituído de “m” antenas de transmissão e “n” antenas de recepção visando o aumento da eficiência do sistema sem fio [28].

Figura 2: Sistema MIMO



O acesso à Internet móvel, difundido por meio das redes 3G, desencadeou um crescimento considerável no tráfego de dados em sistemas de comunicação digital sem fio. Tecnologias com maior vazão de dados tornaram-se cruciais para atender à crescente demanda, assim como mecanismos para aperfeiçoar o desempenho do sistema em termos de taxa de erro de bits (BER, bit error ratio) e mitigar fenômenos de propagação por multipercursos, inerentes em ambientes de propagação sem fio. Segundo o teorema de Shannon-Hartley, a capacidade de um canal, na presença de ruído depende basicamente da largura de banda disponível (B) e da relação sinal-ruído (SNR, signal-to-noise ratio), conforme a equação 1 abaixo. Portanto, exigem-se maior potência de transmissão e/ou largura de banda para obter sistemas digitais com maior vazão. O aumento da potência de transmissão resulta em redução na eficiência de energia do sistema. Sistemas SISO (single-input and single-output) com maior largura de banda são susceptíveis ao desvanecimento multipercurso. Além disso, a atribuição de porções do espectro eletromagnético a diferentes sistemas de comunicação o tornou escasso. Modulações digitais de alta ordem propiciam altas taxas de transferência de dados aos sistemas de comunicação digital sem fio. Todavia, necessitam de maior relação sinal-ruído para manter seu desempenho similar ao de modulações de ordem inferior sob uma determinada BER. Desta maneira, estabelece-se uma relação de compromisso entre eficiência espectral e potência de transmissão [7].

$$C=B.\text{Log}_2(1+\text{SNR}) \quad (1)$$

A diversidade de antenas, por sua vez, proporciona aumento na capacidade de sistemas de comunicação e dispensa o uso de maiores recursos de banda e potência. Múltiplas antenas em rádios transmissores e/ou receptores fornecem mecanismos para combater os efeitos ocasionados pelos fenômenos de multipercurso. Tem-se melhorias na BER e maior confiabilidade e eficiência espectral [7].

Sistemas de comunicação sem fio com múltiplas antenas transmissoras e receptoras são denominados sistemas MIMO. Esses sistemas se beneficiam do fenômeno de propagação por multipercursos, inerentes a ambientes de propagação sem fio, e emprego de técnicas de diversidade para prover maior vazão de dados, cobertura e uso eficiente do espectro de frequências. Os sistemas MIMO tem sido incorporados às redes de telefonia móvel 4G e aos padrões de rede WiFi (Wireless Fidelity). Além de serem amplamente difundidos para aplicações nas futuras redes de telefonia móvel 5G [7].

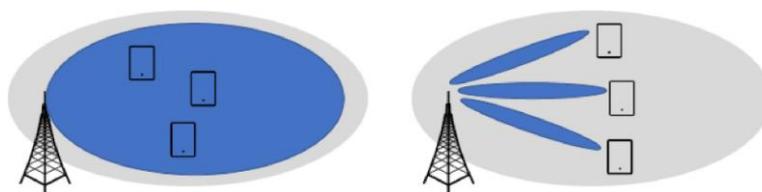
O MIMO massivo estende o conceito do MIMO, permitindo que centenas de antenas sejam implementadas nas estações rádio base assim como no equipamento do usuário. Com MIMO massivo é possível a transmissão de múltiplos fluxos de dados visando uma alta eficiência espectral. Também possibilita a utilização de técnicas de mitigação de interferência como o direcionamento de feixes (beamforming), realizando o direcionamento de potência para o usuário que a demanda e reduzindo para usuários com menor necessidade. Assim há redução de interferência entre os usuários, possibilitando um incremento de capacidade e qualidade de serviço da rede. A utilização de ondas milimétricas possibilita diminuir as dimensões dos elementos que formam as antenas, aumentando ainda mais o número de elementos de antena [10].

3.3.BEAMFORMING

Tecnologia de beamforming consiste na formação de um feixe direcional espacial entre um transmissor e um receptor. Esta técnica pressupõe conhecimento do canal através do CSI (Channel State Information). No beamforming o transmissor com múltiplas antenas faz uma pré-codificação dos dados, ajustando fase e ganho em cada antena. Conseguem-se desta forma maior alcance, redução de interferência e maior vazão total [28].

Beamforming é uma técnica que visa direcionar a transmissão diretamente para o receptor ou esperar a transmissão vinda da direção do emissor. Essa abordagem permite melhorar a relação sinal-ruído, signal-to-noise ratio (SNR), atingindo-se maior taxa de transmissão [11]. Ainda dentro de beamforming, pode-se utilizar técnicas estendidas como multibeam, em que uma antena possui múltiplos feixes (beams) dinamicamente controlados que otimizam a conexão para um dispositivo específico. A Figura 3 ilustra a aplicação de single beam e multibeam.

Figura 3: Single Beam e Multibeam, respectivamente



Fonte: [11]

3.4. CLOUD RAN:

A fim de atender aos requisitos e diversos cenários da rede 5G se fará necessário a virtualização do núcleo e da borda da rede. Uma nova arquitetura de rede de acesso via rádio Cloud RAN (C-RAN) foi proposto para dar suporte à implantação das futuras redes 5G.

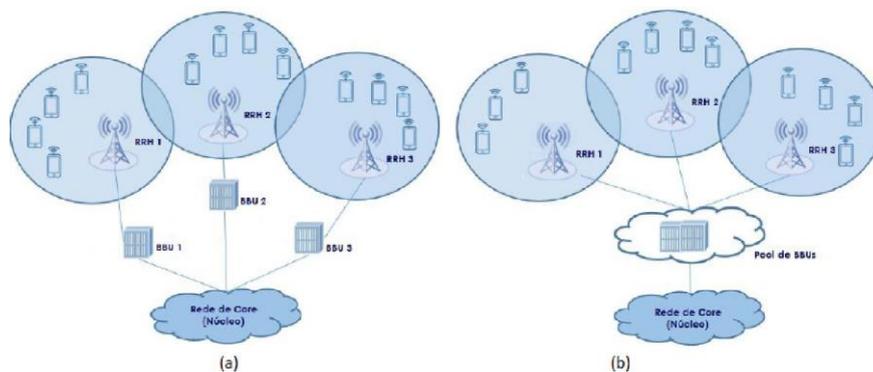
A fim de atender aos requisitos de gerenciamento de recursos do 5G é fundamental uma nova arquitetura para Redes de Acesso via Rádio (RAN).

Em arquiteturas RAN tradicionais, cada ERB realiza o processamento da banda base através de uma Unidade de Banda Base (BBU, BaseBand Unit) e de rádio frequência através de uma Unidade Remota de Rádio (RRH/RRU, Remote Radio Head) localizada na ERB [9,10].

A arquitetura C-RAN (Cloud Radio Access Network), conhecida como Rede de Acesso via Rádio Centralizada ou em Nuvem, é caracterizada por centralizar os recursos de processamento de banda base, em um “pool” de BBUs de modo a simplificar a gestão e otimizar os recursos de rede, sendo este pool compartilhado entre as ERBs. Já os recursos de processamento de rádio permanecem nas bordas da rede [9,10].

A Figura 4 [10] ilustra a arquitetura RAN tradicional em comparação com a arquitetura C-RAN, com os RRHs localizados nas ERBs e as BBUs centralizadas no pool de BBUs.

Figura 4: Evolução para a C-RAN

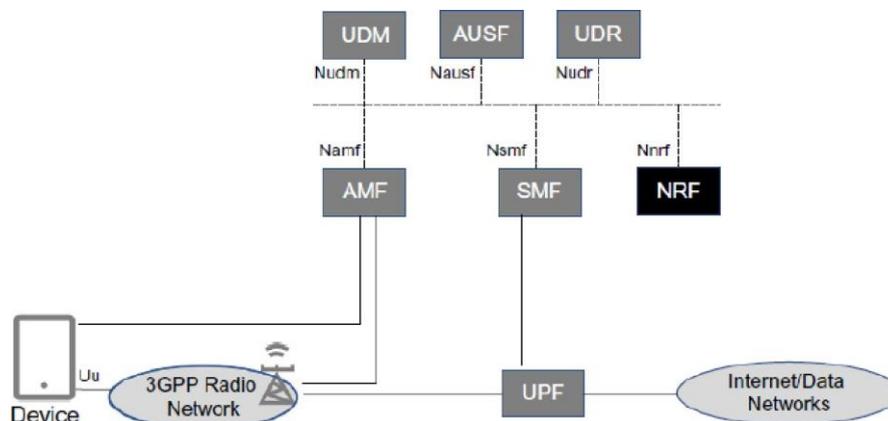


4. O CORE 5G

O Core é a estrutura responsável por gerenciar os recursos de rede e as conexões dos usuários, como autenticação, QoS, segurança, roteamento, encaminhamento de pacotes, mobilidade, interceptação legal e outros. Juntamente com a interface de acesso via rádio, Radio Access Network (RAN), compõe a rede móvel de telecomunicação. No 5G, o Core foi concebido tendo-se como foco uma arquitetura baseada em serviços, ou seja, uma estrutura flexível em que suas funcionalidades são divididas em nodos ligados a um barramento comum que interagem entre si de forma independente. Essa abordagem permite a utilização de recursos em nuvem, de orquestração e de edge computing para escalonar a rede e adaptá-la continuamente a variações na carga de processamento, atingindo-se, assim, maior eficiência e estabilidade. Para comunicação entre os nodos, chamados de Network Functions (NFs), a 3GPP definiu uma API (Application Programming Interface) que funciona sobre o protocolo HTTP (Hypertext Transfer Protocol) e que segue o modelo REST (Representational State Transfer). O modelo REST, proposto em 2000 [14], conta com amplo suporte nas linguagens de programação e é muito empregado na Web, onde muitas empresas disponibilizam uma API em REST para interação com seus produtos.

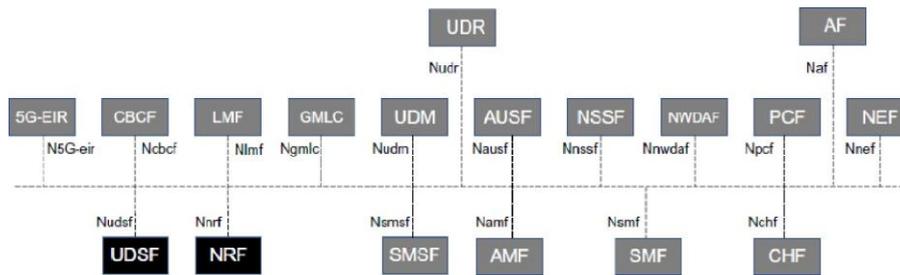
As funções de rede essenciais do Core 5G (mostradas na Figura 5) são: AMF, UDM (Unified Data Management), AUSF (Authentication Server Function), UDR (Unified Data Repository), SMF (Session Management Function), NRF (Network Repository Function) e UPF [11]. São elas que gerenciam a autenticação, estabelecimento de sessão, roteamento, interface com a rede de rádio, interface com o equipamento do usuário (UE), mobilidade, estabelecimento de túnel com a rede de dados ou Data Network (DN), entre outras funcionalidades. No entanto, há ainda funções de rede adicionais que estendem as capacidades do Core (mostradas na Figura 6), como a PCF (Policy Control Function), NSSF (Network Slice Selection Function), N3IWF (Non-3GPP Inter Working Function), AF (Application Function), NEF (Network Exposure Function), SMSF (Short Message Service Function), LMF (Location Management Function) e outras [11].

Figura 5: Elementos essenciais de uma rede 5G



Fonte: [11]

Figura 6: Funções de rede de um Core 5G



Fonte: [11]

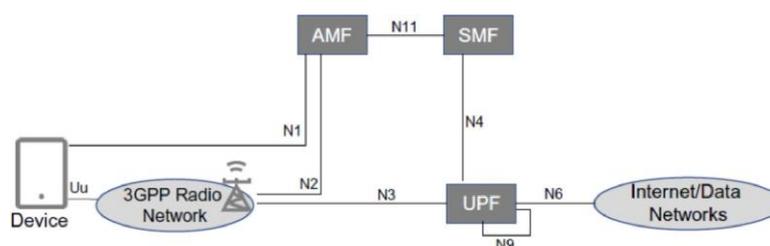
O conjunto de protocolos NAS (Non Access Stratum), composto por 5GMM (5GS Mobility Management) e 5GSM (5GS Session Management), intermedia a conexão entre o UE e o Core. Para as mensagens de controle entre UE e AMF, destinadas ao gerenciamento do registro de dispositivos, mobilidade e segurança, o protocolo 5GMM é utilizado, enquanto que para as mensagens entre UE e SMF, como as destinadas ao gerenciamento das sessões PDUs (Protocol Data Unit), utiliza-se o protocolo 5GSM [11]. As próximas subseções descrevem as principais funções de rede do Core 5G.

4.1. FUNÇÕES DE REDE

4.1.1. AMF

O AMF (Access and Mobility Management Function) é responsável por estabelecer a conexão com a RAN, através da interface N2, e com o UE, através da interface N1 [11]. O AMF gerencia o registro, autenticação e mobilidade do UE, além de gerenciar a encriptação e integridade das mensagens NAS [13]. Também, retransmite as mensagens de gerenciamento de sessão, Session Management (SM), entre UE e SMF, as mensagens SMS (Short Message Service) entre UE e SMSF, as mensagens dos serviços de localização entre UE e LMF e entre RAN e LMF e as mensagens de política entre UE e PCF [13]. Por fim, conta ainda com suporte para entrega de mensagens de aviso públicas, Public Warning System (PWS), e interface para interceptação legal, Lawful Interception (LI) [13]. Ressalta-se que um determinado UE só poderá ser servido por um AMF em um instante de tempo [11]. A Figura 7 ilustra as interfaces utilizadas pelo AMF para interação com as demais funções de rede.

Figura 7: Interfaces de rede utilizadas pela AMF

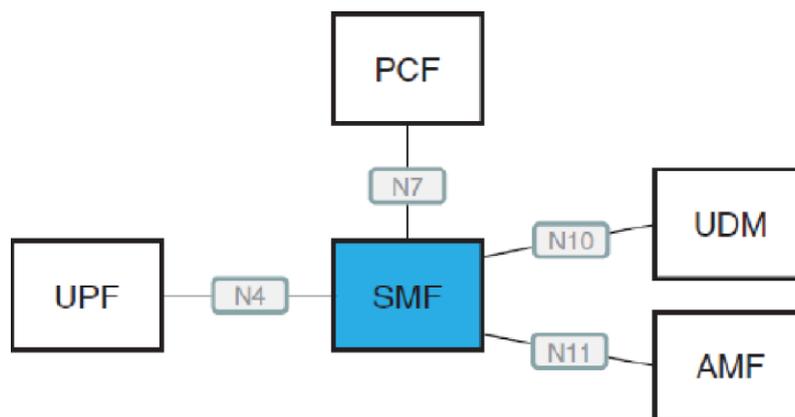


Fonte: [11]

4.1.2. SMF

O SMF (Session Management Function) estabelece e gerencia a conexão entre o UE e a rede de dados (DN) [11]. Para isso, ele seleciona o UPF de acordo com os requisitos da conexão e estabelece uma sessão PDU (Protocol Data Unit) entre o acesso e o UPF ou entre UPFs quando necessário [13]. A interface de comunicação entre SMF e UPF é chamada de N4 [12]. O protocolo PFCP (Packet Forwarding Control Protocol), desenvolvido para o 4G e evoluído para o 5G, é utilizado na camada de aplicação da interface N4 e atua em cima do protocolo UDP [11]. O estabelecimento da sessão PDU, bem como de suas características, depende dos requisitos do UE, das informações dos bancos de dados do UDM/UDR e das políticas de serviço e QoS configuradas no PCF [13]. O SMF também pode alocar endereços IPs para as sessões PDUs e possui suporte para interface de interceptação legal (LI) [11,13]. A Figura 8 ilustra as interfaces utilizadas pelo SMF para interação com as demais funções de rede.

Figura 8: Interfaces de rede utilizadas pelo SMF



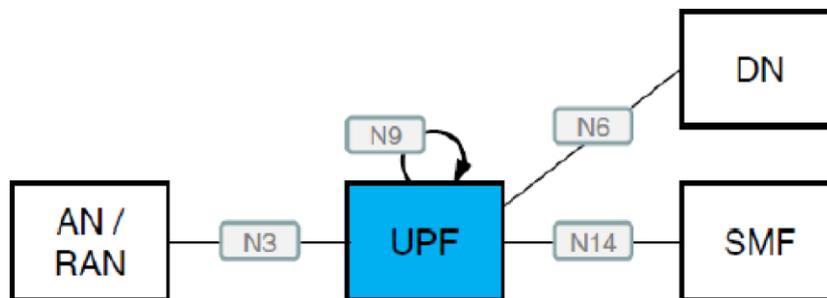
Fonte: [12]

4.1.3. UPF

O UPF (User Plane Function) é a função de rede que gerencia o tráfego do usuário. Suas interfaces de rede estão expostas na Figura 9. Ele é escolhido e controlado pelo SMF, no que tange as políticas de serviço, e serve como ponte entre a rede de dados (DN) e o UE, roteando, processando e direcionando os pacotes de acordo com as regras do SMF [11]. Desse modo, atua como ponto de ancoragem para a sessão PDU, abstraindo os eventos de mobilidade na rede e armazenando os pacotes que não podem ser entregues ao UE em um buffer

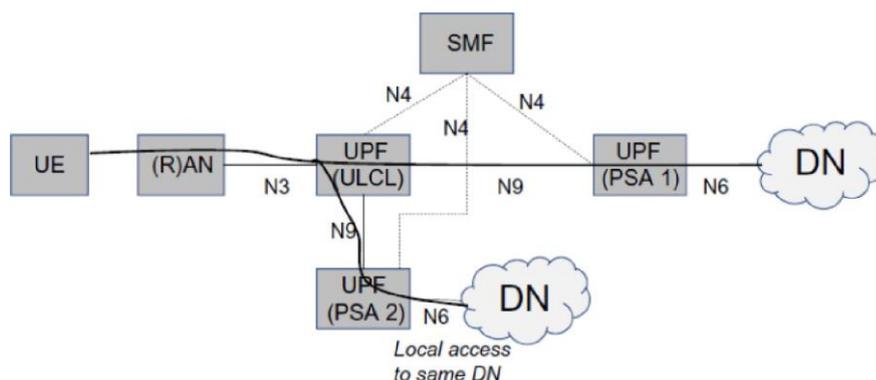
para posterior encaminhamento [11]. Pode ser disposto de forma geograficamente centralizada ou distribuída, não havendo restrições no número de UPFs servindo uma sessão PDU, conforme especificações da 3GPP [13]. Além disso, podem ser implementados em série, encaminhando-se o tráfego com base em regras de roteamento [11]. O UPF é responsável, também, por coletar dados estatísticos de tráfego, gerar relatórios, aplicar QoS de acordo com a demanda da rede e replicar o tráfego para monitoramento legal [13]. Há uma funcionalidade no UPF chamada de Up Link Classifier (UL CL) que permite direcionar o tráfego para diferentes UPFs baseado em regras de encaminhamento fornecidas pelo SMF. O UL CL gerencia o envio dos pacotes do UE para as diferentes redes e das redes de volta para o UE, podendo ser adicionado ou removido a qualquer momento pelo SMF [11]. A Figura 10 ilustra o funcionamento do UL CL.

Figura 9: Interfaces de rede utilizadas pelo UPF



Fonte: [12]

Figura 10: Direcionamento do tráfego através de UL CL



Fonte: [12]

4.1.4. NRF

O NRF (Network Repository Function) serve como repositório das funções de rede (NFs) disponíveis para o Core. Ele armazena as características que descrevem cada NF registrado e permite que outras NFs consultem seu banco de dados para obter o endereço, na rede, dos serviços desejados [11]. A grande vantagem trazida pelo NRF é a não exigência de conhecimento prévio dos endereços e perfis dos elementos do Core que compõem a rede, podendo ser requisitados no momento em que uma determinada NF necessitar de um serviço de outra NF. Cada NF, ou entidade representando a NF, é responsável por registrar-se no NRF e atualizar seu status, porém, o NRF dispõe de um mecanismo keep alive que identifica elementos que não estão mais disponíveis [11]. Mudanças na estrutura do Core ou de escalabilidade são simplesmente efetuadas através de atualizações do status da NF alterada no NRF, sem necessidade de alterar-se as configurações internas das outras NFs. Dentre as informações de perfis disponíveis no NRF, pode-se citar: tipo da NF, ID (identificação), endereço, capacidade, serviços suportados e informações de autorização [12].

4.1.5. UDM

O UDM (Unified Data Management) acessa e gerencia os dados de inscrições armazenados no UDR, envia dados relevantes para as NFs que servem o UE, como AMF e SMF, autoriza acessos e serviços, autentica usuários, gerencia identificação de usuários e oferece suporte a serviços SMS [13]. De forma geral, o UDM fornece uma interface de acesso aos bancos de dados com informações da rede, permitindo que um usuário utilize múltiplos UDMs para transações diferentes [11].

4.1.6. UDR

O UDR (Unified Data Repository) armazena e fornece acesso de dados de inscrições para o UDM, dados de políticas para o PCF e dados estruturados para exposição para o NEF [13].

4.1.7. AUSF

O AUSF (Authentication Server Function) realiza a autenticação primária e o estabelecimento de chaves entre o UE e a rede, utilizando para isso informações do UDM [11, 13]. Suas interfaces com o AMF e o UDM podem ser visualizadas na Figura 11.

Figura 11: Interfaces de rede utilizadas pelo AUSF

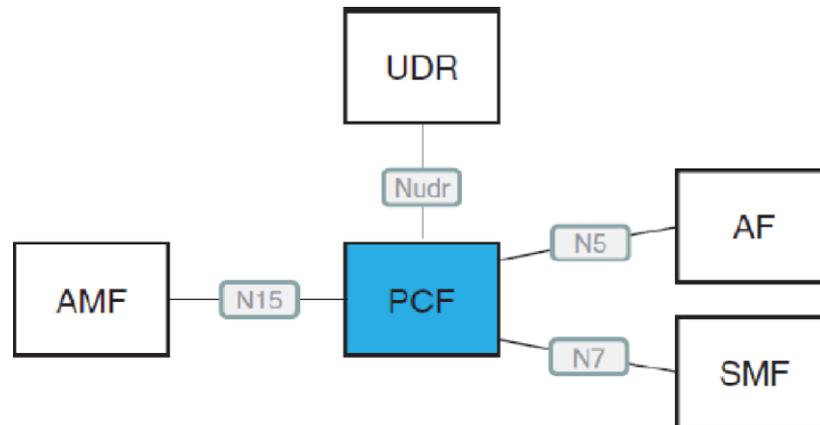


Fonte: [12]

4.1.8. PCF

O PCF (Policy Control Function) é responsável por armazenar e prover as políticas de serviço para as NFs [13]. Para o SMF, as políticas fornecidas são os níveis de QoS e regras de tráfego e cobrança, as relacionadas ao estabelecimento da sessão PDU e as relacionadas ao tráfego local que podem influenciar a escolha do UPF por parte do SMF (4). Para o AMF, o PCF fornece as regras de acesso e mobilidade, como restrições de áreas de serviço e prioridades de acesso [13]. Para o UE, o PCF fornece, através do AMF, políticas relacionadas ao acesso não 3GPP, políticas de escolha de fatias na rede (slicing), de escolha de redes de dados (DN), entre outras [13]. As interfaces de rede do PCF estão expostas na Figura 12. As políticas são determinadas considerando-se fatores como condição da rede, políticas da operadora local, requisitos de aplicativo e dados de assinatura do usuário.

Figura 12: Interfaces de rede utilizadas pelo PCF



Fonte: [12]

4.1.9. NSSF

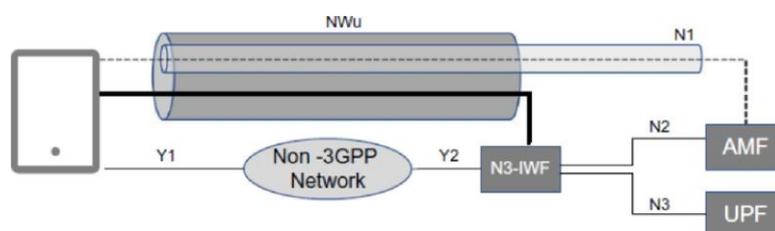
O NSSF (Network Slice Selection Function) é o elemento que seleciona as fatias de rede para serem utilizadas pelo UE (1). Tal procedimento é realizado com base no parâmetro de Single Network Slice Selection Assistance

Information (S-NSSAI), que define a fatia desejada (4). Também, o NSSF lista os AMFs que podem servir o UE, podendo consultar o NRF para isso (4)

4.1.10. N3IWF

O N3IWF (Non-3GPP Inter Working Function) permite a integração de redes que não são parte do acesso definido pela 3GPP, como WiFi, ao Core 5G [11]. Essa interconexão é feita através do estabelecimento de túneis IKEv2 (Internet Key Exchange) e IPsec (IP Security Protocol) entre N3IWF e UE [12]. A Figura 13 ilustra a utilização do N3IWF para conectar uma rede não 3GPP ao Core 5G.

Figura 13: N3IWF conectando uma rede não 3GPP ao Core 5G

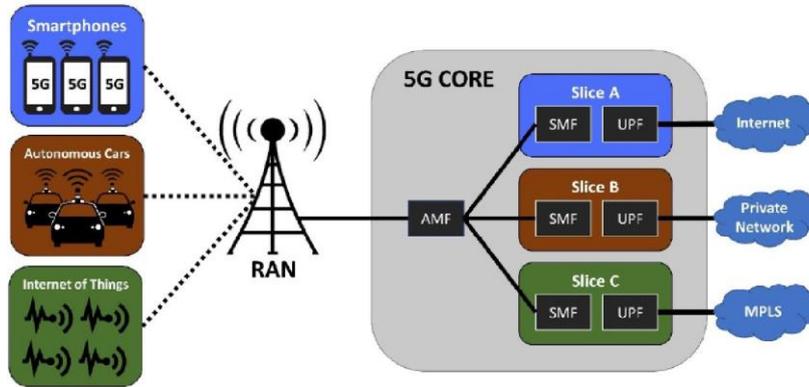


Fonte: [11]

4.2. FATIAMENTO DA REDE

Fatiamento de rede ou network slicing é um dos recursos mais importantes e inovadores do 5G, sendo essencial para modelos de negócio Network as a Service (NaaS). Ele consiste em dividir a rede física em partes (slices) lógicas e fornecer acesso diferenciado para cada tipo de dispositivo, ou seja, usuários que necessitem de comunicação em tempo real, como carros autônomos, poderão ocupar uma fatia/slice que possui maior prioridade e QoS enquanto que usuários que não dependem de latência para suas aplicações poderão ocupar outra fatia/slice de menor prioridade e, conseqüentemente, de menor custo. Além de prover serviços com prioridades otimizadas para os usuários, o fatiamento pode ser utilizado também para aumentar a segurança da comunicação, uma vez que pode manter as diferentes redes lógicas isoladas uma da outra. A Figura 14 demonstra o fatiamento sendo empregado para três conexões distintas: smartphone, carros autônomos e Internet of Things (IoT). Ressalta-se que para realizar os fatiamentos, tecnologias tais como Software Defined Networking (SDN) e virtualização são de grande utilidade.

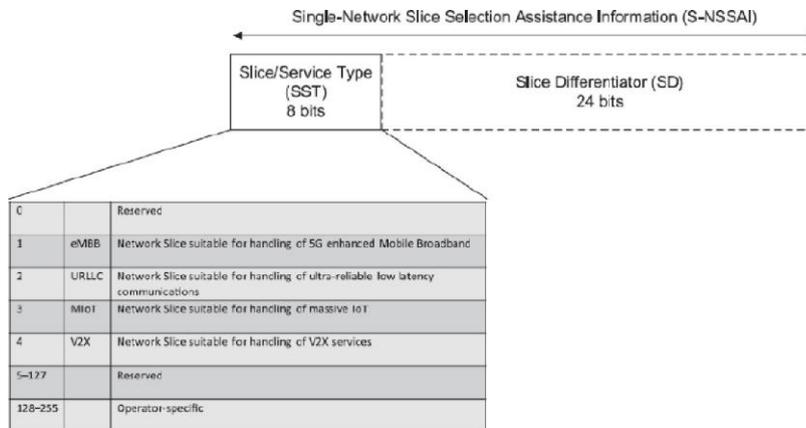
Figura 14: Diferentes dispositivos utilizando o fatiamento de rede



Fonte: [11]

Para identificar cada fatia, atribui-se a elas um valor chamado de S-NSSAI que, por sua vez, é dividido em SST (Slice/Service Type), que indica o tipo do serviço ou fatia, e SD (Slice Differentiator), parâmetro opcional que indica as diferentes fatias de um mesmo tipo (mesmo SST) [11]. A figura 15 mostra a regra de atribuição do S-NSSAI às fatias.

Figura 15: Formato do S-NSSAI



Fonte: [11]

Um dispositivo pode requisitar múltiplas fatias e utilizá-las simultaneamente. Para isso, com base no S-NSSAI, a rede irá selecionar um AMF que poderá servir

determinado dispositivo inicialmente [11]. A função de rede NSSF poderá ser utilizada para a escolha definitiva do AMF ou, caso não se utilize NSSF, o próprio AMF poderá fazer tal escolha [11].

5. DESENVOLVENDO AS ARQUITETURAS PARA O FREE5GC PARA SERVIÇOS E COMPUTAÇÃO DE BORDA

5.1. Arquitetura Baseada em Serviços – SBA

5.1.1. Metodologia

Para se realizar os serviços de Core, escolheu-se o pacote de aplicativos free5GC que implementa as principais funções de rede de um Core 5G e possui código aberto com licença Apache 2.0. Tal aplicativo é mantido pela National Chiao Tung University, de Taiwan, e sua primeira versão foi lançada em janeiro de 2019 [15]. O free5GC é a única opção de código aberto mantida ativamente com suporte a várias NFs e que possui também compatibilidade com um gerenciador de fatiamento (slicing) de rede chamado de free5gmano [16].

As funções de rede implementadas pelo free5GC são as seguintes: AMF, AUSF, N3IWF, NRF, NSSF, PCF, SMF, UDM, UDR e UPF. Além disso, há uma interface Web que permite cadastrar os dispositivos de usuário no Core e visualizar informações da conexão como IP, estado, IMEI (International Mobile Equipment Identity), entre outras. O pacote utiliza o protocolo HTTP2 para implementar a arquitetura REST e conta com criptografia TLS (Transport Layer Security) v1.2 para assegurar a privacidade dos dados trocados entre as NFs, porém optou-se por desabilitá-la para facilitar a análise do tráfego.

O aplicativo Docker foi utilizado para a implementação do Core em containers, uma vez que é a ferramenta mais popular para esse tipo de virtualização. Como são várias as funções de rede que precisam ser inseridas em containers, um aplicativo da família Docker chamado de Compose foi empregado. O Docker Compose permite o gerenciamento de múltiplos containers de forma centralizada, possibilitando que se configure o conjunto de containers através de uma única configuração e que se controle os estados de todos os containers com um único comando.

A imagem para os containers que irão executar as funções de rede foi criada a partir da compilação do pacote free5GC e dos módulos necessários. De modo a tornar a imagem mais enxuta, criou-se um ambiente de compilação e então copiou-se os binários resultantes para uma imagem Ubuntu 18.04 que foi utilizada como base para os containers. Ressalta-se que para o correto funcionamento do UPF, um módulo de Kernel chamado gtp5g precisou ser

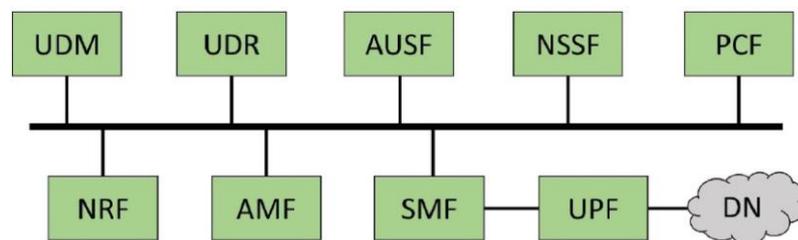
previamente instalado na máquina hospedeira. A máquina hospedeira é uma máquina virtual com sistema Ubuntu 18.04, 4 GB de memória RAM (Random Access Memory) e 2 núcleos de processamento.

Os containers contendo as funções de rede não são persistentes, ou seja, sempre que parados ou reiniciados, seus arquivos voltam ao estado da imagem base. Logo, para que se pudesse salvar de forma permanente os dados armazenados e as configurações modificadas, um volume acessível pelo hospedeiro e pelo container foi criado. Além de persistir os dados do container, o volume também evita que seja necessário gerar uma nova imagem sempre que a configuração de alguma função de rede for alterada, pois o usuário pode simplesmente atualizar o arquivo no volume e reiniciar a aplicação no container.

Cada container recebeu um IP local privado para facilitar a comunicação das funções de rede. Para o UPF, foi necessário atribuir permissão de administrador de rede para que as rotas fossem corretamente criadas durante o estabelecimento de uma conexão.

Optou-se por realizar um design básico do Core 5G sem a necessidade dos recursos providos pelo N3IWF. Essa NF, que serve como interface para redes não 3GPP, como Wi-Fi, não será utilizado pois o escopo desses experimentos limita-se a rede 5G somente. A Figura 16 ilustra o Core 5G implementado.

Figura 16: Modelo de Core implementado para o experimento



Fonte: [11]

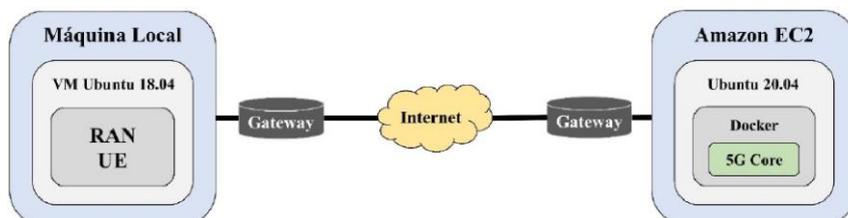
Realizou-se o cadastro das informações identificadoras do dispositivo de usuário no Core por meio da interface Web. As informações cadastradas foram: PLMN (Public Land Mobile Network), SUPI (Subscription Permanent Identifier)/IMSI (International Mobile Subscriber Identity), método de autenticação, chave K, tipo de código de operadora e código de operadora. Para simular o registro e tráfego de dados de um dispositivo de usuário no Core, adaptou-se alguns algoritmos de testes fornecidos pelo pacote free5GC com o intuito de prover melhor ambiente de coleta e análise de dados. Tal algoritmo realiza o registro de um UE simulado no Core 5G, através de um acesso via rádio

também simulado, e então executa um procedimento de ping para verificar se o túnel foi corretamente estabelecido entre o dispositivo e o UPF. Após o ping o UE será desregistrado da rede. É esperado que o acesso via rádio

simulado se comunique efetivamente com o Core e permita a criação da interface N1 que liga o dispositivo de usuário (UE) diretamente ao AMF. Para que o registro do UE seja concluído, o AMF deverá interagir com as demais funções de rede (NFs), requisitando informações do NRF quando necessário, e o SMF deverá verificar a identidade do dispositivo e atribuir uma sessão PDU para conexão, alocando o IP da sub-rede configurada. Ao final, espera-se que o túnel entre o dispositivo de usuário e o UPF seja estabelecido com êxito e que o primeiro tenha acesso a rede de dados (DN) desejada (subrede 60.60.0.0/24). Uma interface de loopback foi configurada no container do UPF com IP 60.0.0.101 e servirá para o teste de ping. O monitoramento do tráfego de pacotes entre as funções de rede, o acesso via rádio e o dispositivo de usuário foi realizado através do software Wireshark, que captura todos os pacotes de uma interface de rede escolhida e permite visualizar informações como IP de origem, IP de destino, porta, tamanho, protocolo, tempo, entre outras. Nesse experimento, escolheu-se para monitoramento a interface de rede criada pelo software Docker Compose que gerenciava os endereços IPs das funções de rede (NFs). Um filtro contendo os protocolos desejados foi aplicado para que a visualização dos pacotes fosse mais direta, sendo inserido da seguinte forma no programa: `pfcip|icmp||gtp||sctp||http2.header`. No caso do protocolo HTTP2, apenas os headers das mensagens serão exibidos para simplificar o filtro. Também, optou-se por configurar hostnames para os IPs das funções de rede de modo a facilitar a identificação na análise dos pacotes. Finalmente, após os testes locais, as configurações e a imagem base para os containers foram transferidas para a plataforma de computação em nuvem da empresa Amazon, chamada de EC2 (Elastic Compute Cloud). A instância alocada para o experimento foi a `t3a.small`, que possui 2 GB de RAM e 2 núcleos de processamento virtuais. Diferentemente da máquina utilizada nos testes locais, o sistema escolhido foi o Ubuntu 20.04, pois possui atualizações mais recentes e customizações específicas que o tornam mais eficiente nesse ambiente. Na nuvem, é possível disponibilizar globalmente o acesso ao Core, podendo ser acessado por qualquer estação de acesso via rádio e dispositivo de usuário na Internet, desde que tenham sido previamente habilitados nas configurações. Para comprovar-se a eficácia da plataforma em nuvem, conectou-se o RAN e UE simulados (instalados na máquina virtual local) no Core 5G remoto, realizando-se os mesmos procedimentos dos testes anteriores e coletando-se o tráfego dos pacotes. Ressalta-se que as portas 2152 e 38412, utilizadas pelos protocolos de transporte UDP e SCTP, respectivamente, foram abertas na nuvem, enquanto que na rede local foi habilitado o direcionamento da porta 2152 (UDP) apenas. O protocolo de encapsulamento GTP-U se beneficia da porta 2152 e da pilha UDP/IP para estabelecimento do túnel e transporte de

dados do usuário. Novamente, espera-se que o RAN e UE simulados possam interagir com as funções de rede e que o ping entre o UE (instalado localmente) e a rede de dados (DN) remota seja concluído com êxito. A Figura 17 mostra como a implementação em nuvem foi testada.

Figura 17: Arquitetura dos testes de comunicação em nuvem



Fonte: [11]

5.1.2. RESULTADOS

Após a configuração do pacote free5GC no software Docker Compose, realizou-se alguns testes para verificar a funcionalidade da plataforma implementada a partir da execução de comandos como `docker-compose up` e `docker-compose down`. Conforme o esperado, a ferramenta foi capaz de iniciar e encerrar todos os containers de forma centralizada, não apresentando nenhuma falha nos arquivos de log das funções de rede (NFs) e permitindo o acesso à interface Web pelo endereço `http://:5000/`. A Figura 18 mostra a página Web de configurações do Core 5G implementado.

Figura 18: Página Web de configuração do Core 5G



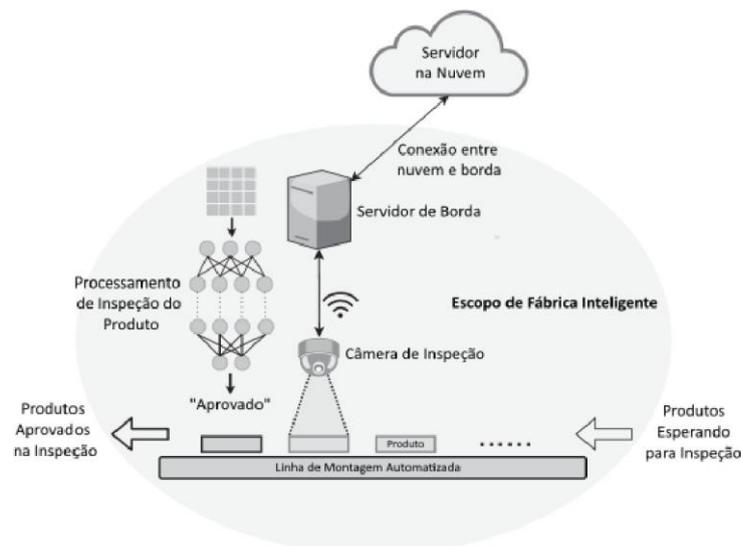
Fonte: [11]

5.2. COMPUTAÇÃO EM BORDA

Dispositivos de IoT tendem a ter um baixo custo o que implica em um baixo poder de processamento de dados, assim, para muitas das aplicações os dados

necessitam ser enviados para um servidor a fim de serem processados. Em alguns casos, a distância entre o dispositivo de IoT e o servidor torna isso inviável devido à barreiras físicas ou ao alto custo de implementação. Utilizar cabo de fibra óptica por exemplo, seria uma solução ótima, mas além do alto custo dos produtos dessa tecnologia, ela requer mão de obra especializada para a instalação e se limita a dispositivos estáticos. Assim, para aliviar os gargalos de latência e com eficiência energética, o 5G disponibiliza novos serviços. A utilização dos novos serviços possibilita a solução do paradigma da computação de borda, que se baseia em uma rede de computação em multicamadas e que pode ser introduzida em fábricas inteligentes, como mostra a Figura 19. Considere o cenário em que uma câmera de inspeção de superfície é configurada em uma linha de montagem automatizada de uma fábrica inteligente fictícia. Os produtos estão fluindo na linha de montagem automatizada, aguardando a inspeção um por um. Quando um produto chega à posição certa, a superfície dele será capturada pela câmera de inspeção, um dispositivo IoT. Então, as imagens vão ser carregadas para o servidor na borda que fará a inferência e análise quase em tempo real [20]. Por fim, as imagens também podem ser transmitidas para os servidores em nuvem para futuras análises e armazenamento de backup.

Figura 19: Linha de montagem adaptada



Fonte: Adaptada [20]

A computação de borda trata de deixar os serviços mais próximos do local onde serão entregues. Os serviços aqui incluem potência de computação e memória necessária para, por exemplo, rodar uma requisição de um aplicativo. A computação de borda, portanto, visa empurrar aplicativos, dados e poder de

computação (serviços) que se encontram longe de pontos centralizados (central de dados) para locais mais próximos do usuário (como centrais de dados distribuídas). O objetivo é atingir uma latência mais baixa e reduzir os custos de transmissão e tráfego no Core da rede. Aplicativos que usam grandes volumes de dados e/ou requerem tempos de resposta curtos, por exemplo, jogos de realidade virtual em tempo real, inspeção de qualidade por vídeo na indústria 4.0, carros autônomos, cidades inteligentes etc., são alguns dos candidatos que podem se beneficiar da computação de borda [11].

A 3GPP não especifica nenhuma solução ou arquitetura especial para computação de borda, em vez disso, a 3GPP define várias ferramentas gerais que podem ser usadas para fornecer um eficiente caminho para o usuário. Essas ferramentas não são específicas para computação de borda, mas podem ser usadas como facilitadores na sua implantação [32].

5.2.1. Seleção da UPF pela SMF

A SMF é responsável pela seleção da UPF. Os detalhes de como isso é feito não são padronizados e dependem de vários aspectos, por exemplo, aspectos de implantação relacionados à topologia de rede das UPFs implantadas, bem como, os requisitos do serviço que será entregue. Quando a SMF faz a seleção de uma UPF, um pré-requisito é que a própria SMF esteja ciente de quais UPFs estão disponíveis em suas respectivas configurações, como recursos da UPF, sequência de carregamento no caso de mais de uma UPF, etc. Uma das formas é que a SMF pode ser configurada com as UPFs disponíveis. Essa configuração pode incluir informações relacionadas à topologia para que a SMF esteja ciente sobre a localização da UPF e de que forma as UPFs estão conectadas. Isso permite que a SMF selecione UPFs adequadas, por exemplo, dependendo da localização do UE. Uma vez que a SMF sabe sobre a(s) UPF(s) disponíveis e há uma necessidade da SMF selecionar uma ou mais UPFs para uma sessão de PDU, como exemplo, no estabelecimento de sessão de PDU ou em algum evento de mobilidade, a SMF pode levar diferentes informações em consideração para selecionar uma UPF. Os detalhes não são padronizados, mas deixados para implementação e configuração do operador [11]. Algumas dessas informações são recebidas da UPF, outras são recebidas da AMF, enquanto que algumas podem ser pré-configuradas na SMF, como as informações relacionadas à topologia do plano do usuário e terminações do plano do usuário.

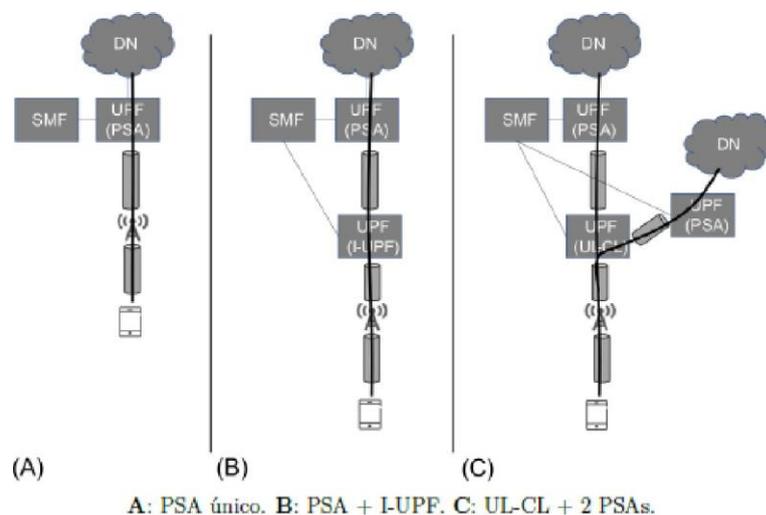
5.2.2. Formas de classificação de tráfego para a DN

Uma sessão de PDU tem no caso mais simples uma única sessão PDU âncora (PSA - PDU Session Anchor) denominada de PSA UPF e, portanto, uma única interface N6 para a DN [11]. Mas uma sessão de PDU também pode ter mais do

que uma PSA UPF e, portanto, várias interfaces N6 para uma DN conforme a Figura 19.

- a) PSA UPF: Esta é a UPF que faz a conexão com a DN através da interface N6.
- b) UPF intermediária (I-UPF): Esta é a UPF que é inserida no caminho do plano do usuário entre a AN e a PSA UPF. Ela encaminha o tráfego entre a AN e o PSA UPF.
- c) UPF com classificador de Up-Link (UL-CL) ou ponto de ramificação (BP): Esta é uma UPF que está “bifurcando” o tráfego para uma sessão de PDU na conexão ascendente e “mesclando” caminhos Up-Link descendente, fazendo funções relacionadas a QoS

Figura 20: Configuração da UPF



Fonte: Adaptada de [11]

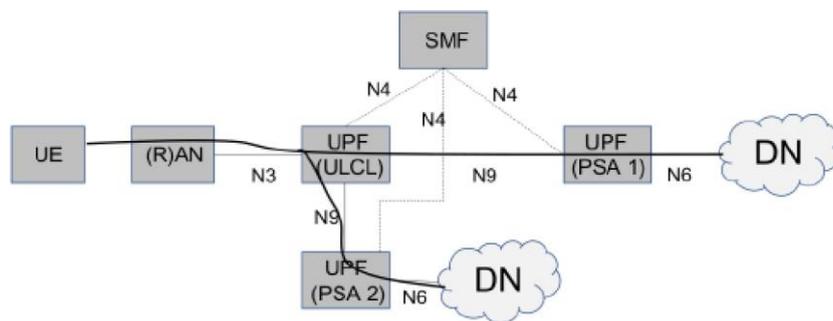
Essa última opção apresentada na Figura 20 pode ser usada para rotear seletivamente o tráfego do plano do usuário para diferentes interfaces N6, por exemplo, rotear de uma PSA UPF com interface N6 para um site periférico local e outra PSA UPF com interface N6 para um data center Remoto [11]. Essa funcionalidade é de suma importância para este trabalho, pois pode ser usada em aplicações da computação de borda 5G.

5.2.3. Classificação Up-Link

Classificação de Up-Link é uma funcionalidade que é suportada por uma UPF onde a UPF desvia parte do tráfego para uma PSA UPF diferente (local) conforme é apresentado na Figura 21. O UL CL fornece encaminhamento de tráfego de ligação ascendente para diferentes Âncoras de Sessão PDU e desvia o tráfego de ligação

descendente para o UE, isto é, o desvio do tráfego de diferentes Âncoras de Sessão PDU na ligação para o UE. O UL CL desvia o tráfego com base nas regras de detecção e encaminhamento de tráfego, fazendo uso de filtros de tráfego fornecidos pela SMF. Assim, o UL CL aplica as regras de filtragem, por exemplo, para examinar o endereço IP de destino dos pacotes IP da conexão ascendente enviados pelo UE e determina como o pacote deve ser encaminhado. A UPF que suporta um UL CL também pode ser controlada pela SMF para oferecer suporte à medição de tráfego e aplicar cobranças. O uso do UL CL se aplica a Sessões de PDU do tipo IPv4 ou IPv6 ou IPv4v6 ou Ethernet, de modo que a SMF possa fornecer filtros de tráfego [11].

Figura 21: Acesso local a DN usando o UL DL



Fonte: Adaptada de [11]

Quando a SMF decide desviar o tráfego, ela insere um UL CL no caminho de dados e uma PSA adicional. Isso pode ser feito a qualquer momento durante a vida útil de uma sessão de PDU. A PSA adicional pode ser colocada na mesma UPF que o UL CL ou pode ser uma UPF autônoma. Quando a SMF determina que o UL CL não é mais necessário, ele pode ser removido do caminho de dados pela SMF [11]. O UE desconhece o desvio de tráfego por parte do UL CL e não participa na inserção e na remoção do UL CL. A solução com o UL CL, portanto, não requer nenhuma funcionalidade específica do UE.

5.2.4. Free5GC Compose

O Free5GC Compose [22] é uma versão em docker compose da 3ª versão do Free5GC [23]. O Free5GC é mantido pela Universidade Nacional Chiao Tung (NCTU - National Chiao Tung University), de Taiwan, bem como, é um projeto de código aberto com licença Apache 2.0 para redes móveis 5G. O objetivo final do Free5GC é implementar as NFs do 5GC definidas pela Versão 15 (R15) da 3GPP em diante. Atualmente, o Free5GC está na 3ª versão que foi entregue em Abril de 2020,

disponibilizando novos recursos para suportar serviços como o protocolo de rede que transmite sinais de televisão via internet (IPTV - Internet Protocol Television). Assim, o primeiro recurso disponibilizado no estágio 3 é o de operação completa do 5GC com o 5G Orchestrator para poder operar, administrar e gerenciar as NFs [23]. Foram disponibilizados na 3ª versão do Free5GC os recursos de acesso para dispositivos não 3GPP (N3IWF) e o classificador de Up-link (UL CL). As NFs da 3ª versão do Free5GC são as seguintes: NSSF, NRF, UDM, PCF, AUSF, AMF, SMF, UPF, WEBUI, N3IWF e o ULCL [23]. A interface gráfica na Web (WEBUI - User Graphic Interface), é uma interface Web disponibilizada pelo Free5GC e que foi implementada em um Docker Container no Free5GC Compose, assim como, as outras NFs da 3ª versão do Free5GC também foram implementadas em Containers [22]. A WEBUI permite cadastrar os dispositivos de usuário no Core com uma série de configurações como: Identidade Internacional de Assinante Móvel (IMSI - International Mobile Subscriber Identity), S-NSSAI, DNN e configurações de QoS. Podendo através da WEBUI também, visualizar informações da conexão como consumo de dados em tempo real, IP do assinante e entre outras funções. A plataforma Free5GC Compose utiliza o Protocolo de Transferência de HiperTexto (HTTP - HyperText Transfer Protocol) para implementar a arquitetura de representação do estado de Transferência (REST - Representational State Transfer) [22]. Além disso, o Free5GC Compose permite o gerenciamento das NFs do 5GC em containers de forma centralizada, possibilitando que se configure um conjunto de containers através de uma única configuração e que se controle os estados de todos os containers com um único comando. Para o gerenciamento e controle dos pacotes de dados em nível de camada de transporte, o Free5GC Compose necessita do GTP5G pré-instalado.

5.2.5. GTP5G

GTP5G é um módulo de kernel Linux personalizado para lidar com pacotes de dados através do protocolo de controle de encaminhamento de pacotes (PFCP - Packet Forwarding Control Protocol). Segundo a 3GPP TS 29.281 [24], o kernel GTP5G é responsável pelas camadas de baixo nível, como a camada de transporte. O protocolo de plano de usuário para tunelamento GPRS (GTP-U - GPRS tunneling protocol user plane) é identificado em cada nó com um identificador do terminal do tunnel (TEID - Tunnel Endpoint Identifier), um endereço IP e um número de porta UDP. Assim, o GTP-U é necessário para permitir o encaminhamento de pacotes entre entidades GTP-U, conforme a 3GPP TS 29.244 [25].

5.2.6. Ambiente de Teste

5.2.6.1. Criação das VMs

Para a simulação do ambiente de teste virtual de computação de borda 5G é necessário primeiramente, a criação de VMs. Optou-se em utilizar o VirtualBox (6.1.26) [26], por ser um software gratuito e de uso intuitivo. O VirtualBox foi instalado em um computador hospedeiro, que tem um processador intel i7 de 5° geração com 4 núcleos de processamento, 8 GB de memória de acesso aleatório (Random Access Memory - RAM), 256 GB de armazenamento em disco de estado sólido (SSD - Solid-State Drive) e um sistema operacional (SO) Ubuntu Desktop 20.04. Após a instalação do VirtualBox, foi criada uma VM com o SO Ubuntu Server 20.04 [27], contendo 1 núcleo de processamento, 30 GB de armazenamento e configurada com 2 adaptadores de rede, o primeiro do tipo tradução do endereço da rede (NAT - Network Address Translation) e o segundo do tipo apenas adaptador do hospedeiro (Host-only Adapter). Após a criação da VM, algumas verificações importantes foram feitas, por exemplo, o acesso a rede externa utilizando o comando Ping. Para a verificação do IP atual da VM utilizou-se o comando ifconfig, conforme o Código Fonte 3.1. Vale ressaltar que antes de executar o comando ifconfig, deve-se instalar primeiramente um pacote de aplicativos net – tools.

Quadro 1 – Configurações das VMs utilizadas no ambiente de computação de borda 5G

VM	Aplicação	VM IP/FQDN ou DNS	SO	Memória RAM	Núcleos	Armazenamento em disco
1	Free5GC Compose	192.168.56.120/ free5gc-compose	Ubuntu Server 20.04	2048 MB	1 un.	30 GB
2	UERANSIM e Cliente Netperf	192.168.56.121/ ueransim-netperf-cliente	Ubuntu Server 20.04	1024 MB	1 un.	30 GB
3	Servidor Netperf Borda	192.168.56.122/ netperf-servidor-borda	Ubuntu Server 20.04	1024 MB	1 un.	30 GB
4	Servidor Netperf Remoto na AWS	3.15.16.208/ ec2-3-15-16-208.us-east-2.compute.amazonaws.com	Ubuntu Server 20.04	1024 MB	1 un.	30 GB

5.2.6.2. Utilização do Free5Gc Compose

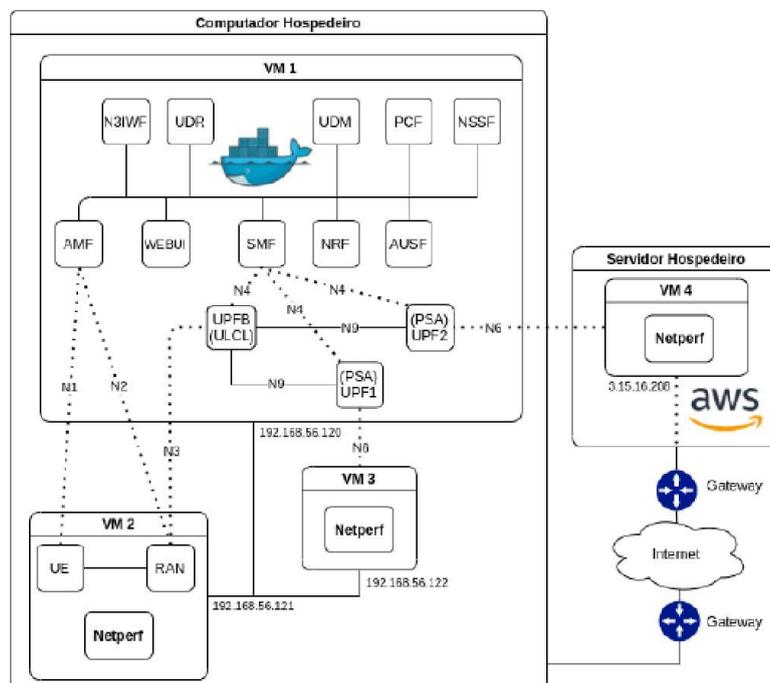
Para a simulação dos serviços do 5GC, escolheu-se o programa Free5GC Compose, o qual foi instalado a partir do repositório do github [22] na VM 1. Além disso, para o correto funcionamento das NFs do Free5GC Compose, o módulo de kernel GTP5G foi instalado. Cada container das NFs implementado no Free5GC Compose tem pré configurado um DNS de um IP local privado para facilitar a comunicação das NFs e facilitar na identificação de cada NF durante a configuração. Assim, realizou-se a verificação dos DNSs, bem como foi alterado o DNS da UPFB referente a interface N6 localizado no arquivo de configurações da SMF do Free5GC Compose para o IP da VM 1. No passo seguinte, adicionou-se comandos no arquivo de configuração do Docker Compose, conforme o Código Fonte 3.8 para realizar a tradução dos endereços que passam pela UPFB e para habilitar o

roteamento de pacotes, apresentados nas linhas 11 e 12. A porta 2152 na linha 20 do Código Fonte 3.8 foi aberta para a UPFB e a porta 38412 foi aberta para a AMF. As portas 2152 e 38412 são utilizadas pelos protocolos de transporte UDP e SCTP, respectivamente e o protocolo de encapsulamento GTP-U utiliza-se da porta 2152 e da pilha UDP/IP para o estabelecimento do túnel e transporte de dados do usuário durante a Sessão PDU. Além disso, para a NRF, AMF, AUSF, NSSF, PCF, SMF, UDM e a UDR, optou-se em manter a porta 8000 referente ao protocolo HTTP, bem como, para a WEBUI manteve-se a porta 5000.

5.2.6.3. Diagrama de simulação do ambiente de computação em borda 5G

Por fim, na Figura 24 é apresentado a arquitetura do 5GC instalado e configurado na VM 1 com o UL CL implementado pela UPFB (UPF-Bifurcada), UPF1 como primeiro ponto de ancoragem e a UPF2 como segundo ponto de ancoragem para o estabelecimento da sessão PDU com o servidor de borda (VM 3) e o servidor remoto (VM 4). Na VM 2 da Figura 22 é ilustrado o UERANSIM através das siglas UE e RAN, bem como, o Netperf para ser utilizado como cliente. Na VM 3 e VM 4 encontra-se instalado o Netperf para ser utilizado como servidor.

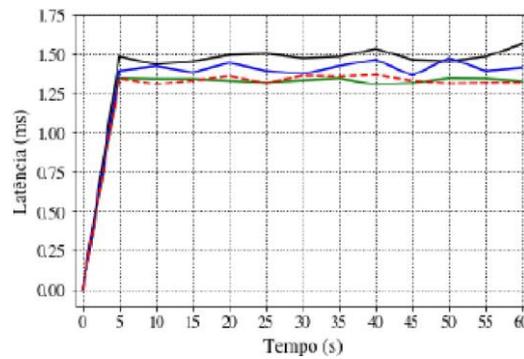
Figura 22: Ambiente configurado para a simulação de computação de borda 5G



5.2.6.4. Resultados

Para ilustrar e apresentar uma comparação dos resultados, segue na Figura 26, as latências obtidas através dos protocolos ICMP, UDP e TCP entre cliente e servidor de borda, utilizando ainda, o protocolo ICMP com o ULCL habilitado. Os dados foram obtidos durante uma leitura de 60 segundos, capturando os valores médios a cada 5 segundos, por exemplo, o valor médio entre o intervalo de 0 a 5 segundos em seguida de 0 a 10 segundos, etc.

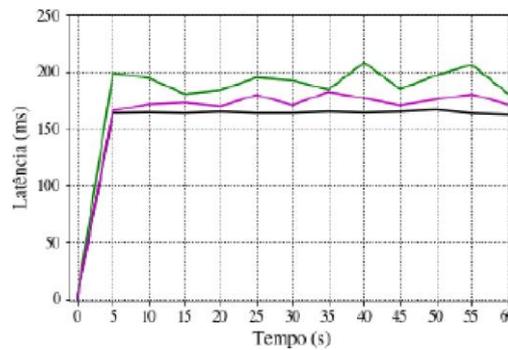
Figura 23: Gráfico das latências obtidas entre cliente e servidor de borda, onde a cor Verde=ICMP sem ULCL; Cor azul=UDP; Cor preta=TCP e Cor vermelha=ICMP com ULCL



Outro gráfico gerado foi o das latências obtidas utilizando os protocolos ICMP, UDP e TCP entre cliente e servidor remoto, conforme a Figura 24.

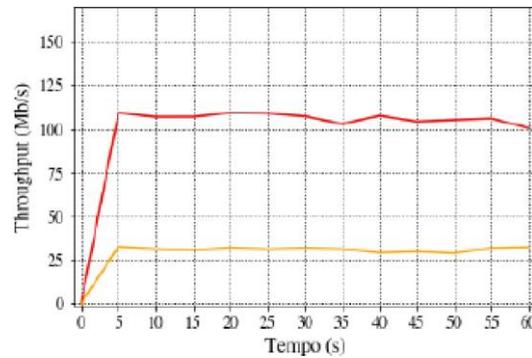
Figura 24: Gráfico das latências obtidas entre cliente e servidor remoto, onde a cor preta=ICMP;

Cor verde=TCP e Cor magenta=UDP



Para o último gráfico gerado é apresentado os throughput obtidos utilizando o protocolo TCP entre cliente, servidores de borda e remoto, conforme o Figura 25.

Figura 25: Gráfico dos Throughputs obtidos utilizando o TCP entre cliente e servidores, onde a Cor vermelho=Servidor de borda; Cor amarelo=Servidor remoto



Na Tabela 1 é apresentado de forma resumida o resultado médio das latências e Throughputs obtidos para os protocolos ICMP, UDP e TCP.

Tabela 1: Comparação dos resultados obtidos

Origem e Destino dos pacotes de dados	Latência	Latência	Latência	Throughput	Throughput
	ICMP (ms)	UDP (ms)	TCP (ms)	UDP (Mb/s)	TCP (Mb/s)
Cliente e Servidor Borda	1.32	1.41	1.56	112.44	100.53
Cliente e Servidor Remoto	162.62	171.43	180.90	43.45	32.27
Cliente e Servidor Remoto (UL CL)	1.31	-	-	-	-

Assim, com os resultados obtidos na implementação do UL CL, conforme apresentado na Tabela 1, pode-se observar que não foram obtidos dados para os protocolos UDP e TCP já que é esperado que os dados sejam os mesmos dos obtidos entre o cliente e o servidor de borda.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1]. Kraus, Dener; “Computação de borda para indústria utilizando a rede 5G”, UFSC-2021. Disponível em:<https://repositorio.ufsc.br/bitstream/handle/123456789/228613/TCC_Dener_Kraus.pdf>. Acesso em: 04/09/2021.
- [2]. Mailer, Christian; “Plataforma de CORE 5G em nuvem para disponibilização de funções de rede como serviço”, UFSC-2020. Disponível em:<https://repositorio.ufsc.br/bitstream/handle/123456789/209624/TCC_20201_ChristianMailer.pdf>. Acesso em: 26/10/2021.
- [3]. Gurjão, E. Livro de Minicursos SBRT 2020. il. [S.l.]: Instituto Federal de Ensino, Ciência e Tecnologia da Paraíba – IFPB, 2020. 142 p. ISBN 978-65-87572-23-9.
- [4]. Palattella, M. R. et al. Internet of Things in the 5G Era: Enablers, Architecture, and Business Models. IEEE Journal on Selected Areas in Communications, v. 34, n. 3, p. 510–527, 2016.
- [5]. Lema, M. A. et al. Business Case and Technology Analysis for 5G Low Latency Applications. IEEE Access, v. 5, p. 5917–5935, 2017.
- [6]. Both, Cristiano Bonato, Cardoso, Kleber Vieira, Macedo, Ciro J.A ; “Soft5G+: explorando a softwarização nas redes 5G”.
- [7]. Cesar, Evandro Vilas Boas; Arranjos de Antenas para 5G com MIMO em Ondas Milimétricas, Dissertação de Mestrado - Inatel, 2019.
- [8]. Carlos, João da Cruz de Lima, GARANTIA DE QOS NO NÚCLEO DA REDE MÓVEL CELULAR DE QUINTA GERAÇÃO UTILIZANDO REDES DEFINIDAS POR SOFTWARE; Dissertação de mestrado- UFC, 2019.
- [9]. Mendes, Hanna Felícia dos Santos (2019); Abordagem teórica da aplicação de Virtualização de funções de rede na tecnologia de comunicação 5G.
- [10]. OLIVEIRA, Lidiano A. N.; ALENCAR, Marcelo S.; LOPES, Waslon Terllizzie A. (2018). Evolução da Arquitetura de Redes Móveis Rumo ao 5G.
- [11]. ROMMER, S. et al. 5G Core Networks: Powering Digitalization. [S.l.]: Academic Press, 2019.
- [12]. PENTTINEN, J. 5G Explained: Security and Deployment of Advanced Mobile Communications. 1. ed. Hoboken, NJ, EUA: Wiley, 2019.
- [13]. CHANDRAMOULI, D.; LIEBHART, R.; PIRSKANEN, J. 5G for the Connected World. [S.l.]: Wiley, 2019.
- [14]. FIELDING, R. Architectural Styles and the Design of Network-based Software
- [15]. SCHOLL, B.; SWANSON, T.; JAUSOVEC, P. Cloud Native: Using Containers, Functions, and Data to Build Next-Generation Applications. 1. ed. Sebastopol, CA, EUA: O’Reilly Media, 2019.

- [16]. WEG-V2COM OPEN LAB. RESULTADOS PRELIMINARES WEGV2COM OPEN LAB 5G. Acesso em: 04/09/2021. Disponível em:<https://sei.anatel.gov.br/sei/modulos/pesquisa/md_pesq_documento_consulta_externa.php?eEPwqk1skrd8hSlk5Z3rN4EVg9uLJqrLYJw_9INcO6fX6o9bVPoiTHXHKDp8z4jNp1Hsw31wuTQX8J-fqjddyWo1pe5dZRrEvwZXjQvETUCBSxyrrp uXwu\EBod27a>.
- [17]. WEG. WEG completa testes práticos de conectividade à rede 5G. Disponível em:<<https://www.weg.net/institucional/BR/pt/news/produtos-e-solucoes/weg-completa-testes-praticos-de-conectividade-a-rede-5g>> Acesso em: 7 jul. 2020.
- [18]. YANG, Y. et al. Intelligent IoT for the Digital World: Incorporating 5G Communications and Fog/Edge Computing Technologies. 1. ed. [S.l.]: Wiley, 2021. ISBN 9781119593546.
- [19]. 3GPP. Universal Mobile Telecommunications System (UMTS); LTE; General UMTS Architecture (3GPP TS 23.101 version 8.0.0 Release 8). Acesso em: 26/08/2021. Disponível<https://www.etsi.org/deliver/etsi_ts/123100_123199/123101/08.00.00_60/ts_123101v080000p.pdf>.
- [20]. National Chiao Tung University. *Free5GC Compose*. Acesso em: 26/08/2021. Disponível em:<<https://github.com/free5gc/free5gc-compose>>.
- [21]. National Chiao Tung University and National Chung Cheng University. *free5GC Link The World!* Acesso em: 26/08/2021. Disponível em:<<https://www.free5gc.org/>>.
- [22]. 3GPP. *General Packet Radio System (GPRS) Tunnelling Protocol UserPlane (GTPv1-U)*. Acesso em: 26/08/2021. Disponível em:<<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1699>>.
- [23]. 3GPP. *Interface between the Control Plane and the User Plane nodes*. Acesso em: 26/08/2021. Disponível em:<<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3111>>.
- [24]. ORACLE. *Download VirtualBox*. Acesso em: 26/08/2021. Disponível em:<<https://www.virtualbox.org/wiki/Downloads>>.
- [25]. CANONICAL. *Get Ubuntu Server*. Acesso em: 26/08/2021. Disponível em:<<https://ubuntu.com/download/server>>.
- [26]. Rochol, Juergen, Sistema de comunicação sem fio, Vol.24, Editora Bookman
- [27]. Luciano, Leonardo de Almeida Maia, A 5ª Geração Móvel e o Futuro da Internet, disponível em: <<https://www.eventos.momentoeditorial.com.br/wp-content/uploads/2016/09/CRR-Leonardo-5G-ISP-CEMIG-Telecom.pdf>>.
- [28]. INTERNATIONAL TELECOMMUNICATION UNION - ITU (Swiss). Future technology trends of terrestrial IMT systems: M Series Mobile, radiodetermination, amateur and related satellite services. Geneva, 2014. 32 p. (Rep. ITU - R M.2320 - 0).
- [29]. INTERNATIONAL TELECOMMUNICATION UNION - ITU (Swiss). IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond: M Series Mobile, radiodetermination, amateur and related satellite services. Geneva, 2015. 21 p. (Recommendation ITU - R M . 2083 - 0).
- [30]. HUO, Y.; DONG, X.; XU, W. 5G Cellular User Equipment: From Theory to Practical Hardware Design. 2017.

2 Fundamentos de LoRaWAN – Teoria e Prática

A. F. Pastório, J. Rossato, J. P. C. A. Sá, F. A. Spanhol, L. A. Rodrigues, E.

T. Camargo

ABSTRACT

Low Power Wide Area Network (LPWAN) is a category of wireless network widely used to provide connectivity to the Internet of Things paradigm and have been widely used to support the construction of Smart Cities. Generally, an LPWAN network allows to periodically send only a few kilobits per second (Kbps), but provides longer battery life for end devices (sensors, actuators) while being able to cover areas of up to 50 kilometers in open field. Among the LPWAN technologies and protocols, the open standard LoRaWAN stands out. LoRaWAN operates on the proprietary wireless technology LoRa (Long Range) and occupies an unlicensed frequency band. Among the important features of the LoRaWAN standard are not charging any fee for network usage and encouraging the customer to install their own network infrastructure. This short course covers the fundamentals of LoRaWAN networks, presenting theoretical aspects illustrated with practical labs. The practical part includes how to perform communication between LoRa devices and communication with a network server in order to obtain the monitored data.

RESUMO

As redes de longo alcance e baixa potência, conhecidas como LPWAN, são uma categoria de rede sem fio largamente utilizada para fornecer conectividade para o paradigma de Internet das Coisas e vem sendo amplamente utilizada para apoiar a construção de Cidades Inteligentes. Geralmente, uma rede LPWAN permite enviar periodicamente apenas poucos quilobits por segundo (Kbps), mas proporciona maior vida útil para as baterias dos dispositivos finais (sensores, atuadores) ao mesmo tempo em que pode cobrir áreas de até 50 quilômetros em campo aberto. Entre as tecnologias e protocolos de LPWAN, destaca-se o padrão aberto LoRaWAN, que opera sobre a tecnologia sem fio proprietária LoRa (Long Range) e ocupa uma faixa de frequência não licenciada. Entre as características importantes do padrão LoRaWAN estão não cobrar qualquer taxa pelo uso da rede e estimular o próprio cliente a instalar sua infraestrutura de rede. Este

minicurso aborda os fundamentos das redes LoRaWAN, apresentando aspectos teóricos ilustrados com laboratórios práticos. A parte prática inclui como realizar a comunicação entre dispositivos LoRa e a comunicação com um servidor de rede a fim de obter os dados monitorados.

1.1. INTRODUÇÃO

Desde a sua origem, as redes de computadores sofrem constantes evoluções. A inserção de capacidades computacionais e de comunicação com objetos comuns do cotidiano, possibilita à dispositivos interagirem entre si, fornecerem informações e serem controlados remotamente através de serviços hospedados em servidores na Internet. Assim, a “Internet das Coisas”, ou *Internet of Things* (IoT), é a integração de pessoas, processos e tecnologia com dispositivos e sensores conectados para permitir o monitoramento remoto, a manipulação e verificação do seu estado e avaliar as tendências de tais dispositivos [10]. Tal demanda exige das redes de computadores requisitos diferentes daqueles empregados para usuários que navegam na Internet, por exemplo, e faz com que o sucesso da sua evolução seja crucial para o avanço do mercado de IoT.

O mercado global de soluções para usuários finais da Internet das Coisas ultrapassou 100 bilhões de dólares em receita de mercado pela primeira vez em 2017 [4]. Há previsão que esse valor crescerá para cerca de 1,6 trilhão de dólares em 2025 [37], chegando a com 41 bilhões de dispositivos em 2027 [42]. Este número deve crescer a medida que a conectividade com a Internet se torna uma característica padrão para uma grande variedade de dispositivos eletrônicos e se deve ao fato de que IoT pode ser aplicado em diversos contextos:

- Cuidados em saúde (*Health Care*): IoT no campo da medicina é destinado a manter as pessoas seguras e saudáveis através de monitoramento em tempo real das funções vitais [20, 34];
- Casa Inteligente (*Smart Home*): em um ambiente de casa inteligente, iluminação, eletrodomésticos, computadores, câmeras de segurança e outros estão conectadas à Internet para permitir que o usuário as monitore e as controle independentemente da restrição de tempo e local [21];
- Rastreamento (*Tracking*): a localização tem muita importância no ambiente de vida conectado, como segurança, vigilância e movimentação de produtos e frotas. A localização e o rastreamento baseados em IoT são considerados mais abrangentes e precisos do que as técnicas utilizadas anteriormente [32, 5];
- Indústria Inteligente ou Indústria 4.0: com a introdução da IoT no ambiente de produção e manufatura, itens físicos como sensores, dispositivos e ativos corporativos são conectados à Internet. As informações geradas servem para a tomada de decisões relevantes no processo industrial e na qualidade do produto final [41];
- Agronegócio: IoT colabora no combate ao desperdício de insumos e aumentando a produção. A utilização das informações geradas por sensores colabora na utilização de recursos importantes para o aumento da produção. Utilizar recursos fundamentais

como a água, por exemplo, é importante independente da cultura cultivada como agricultura, suinocultura, pecuária e outras [9] e;

- Cidades inteligentes (*Smart Cities*): A melhoria da qualidade de vida nas cidades modernas é um problema diário com que os cidadãos e a administração pública precisam lidar. A utilização de dispositivos inteligentes baseados em IoT gera informações para monitorar a qualidade da água e do ar com alerta para aumento de poluentes, coleta de informações do consumo de água e energia elétrica das residências e órgãos públicos, controle de tráfego urbano, mobilidade urbana e outros

[33] [5].

Dependendo da aplicação o uso de energia da rede elétrica e a conexão com a Internet através de cabo *ethernet* são inviáveis. Assim a utilização de comunicação sem fio de baixo consumo energético nos objetos inteligentes é uma necessidade e um grande desafio, pois o fator energia é um dos principais limitantes em IoT e na maioria dos casos, o rádio utilizado para comunicação é um dos principais consumidores. Estimativas indicam que o custo energético da comunicação chega a 60% do gasto energético do dispositivo [26].

A Figura 1.1 apresenta algumas tecnologias de rede de comunicação baseadas em ondas de rádio (sem fio) e as suas coberturas alcançadas. No contexto IoT destacam-se as redes sem fio de baixo consumo de energia, baixa largura de banda e alcance variando de alguns centímetros até quilômetros. Considerando o alcance, as redes *Wireless Personal Area Network* (WPAN) foram projetadas para interconectar dispositivos eletrônicos dentro do espaço de uma pessoa, sendo exemplos BLE e ZigBee. *Bluetooth Low Energy* (BLE) transmite na frequência de 2,4 GHz, com 1 Mbps de taxa máxima de transmissão e alcance máximo de transmissão de 50 metros. ZigBee por sua vez opera nas frequências de 2,4 GHz, 868 MHz e 915 MHz com uma taxa máxima de 250 kbps e com um alcance máximo de 100 metros [8]. Redes WPAN podem ter cobertura próxima a proporcionada por redes de área local sem fio, ou *Wireless Local Area Network* (WLAN). Contudo uma WLAN oferece larguras de banda significativamente maiores e maior consumo energético. Já redes projetadas para conectar dispositivos a longas distâncias são chamadas *Wireless Wide Area Network* (WWAN) e quando enviam uma pequena quantidade de dados por longas distâncias, mantendo baixo consumo energético, são conhecidas como LPWAN (*Low Power Wide Area Network*) [6].

Dentre as tecnologias LPWAN mais utilizadas atualmente destacam-se: NB-IoT [11], Sigfox [35] e LoraWAN [15]. NB-IoT (*Narrowband Internet of Things*) é uma tecnologia de rádio celular especificada pela 3GPP para IoT, atua na faixa de 700 MHz com uma taxa de dados de 170 kbps (*downlink*) e 250 kbps (*uplink*), alcançando aproximadamente 35 km. Os serviços de infraestrutura de rede são fornecidos por operadoras com a cobrança de taxa pela utilização. Sigfox é uma tecnologia proprietária da empresa francesa de mesmo nome. Utiliza a faixa de 900 MHz com a largura de canal de 1 kHz, com um alcance de 3 km a 50 km e uma taxa de dados de 100 bps transportando até 12 bytes de *uplink*.

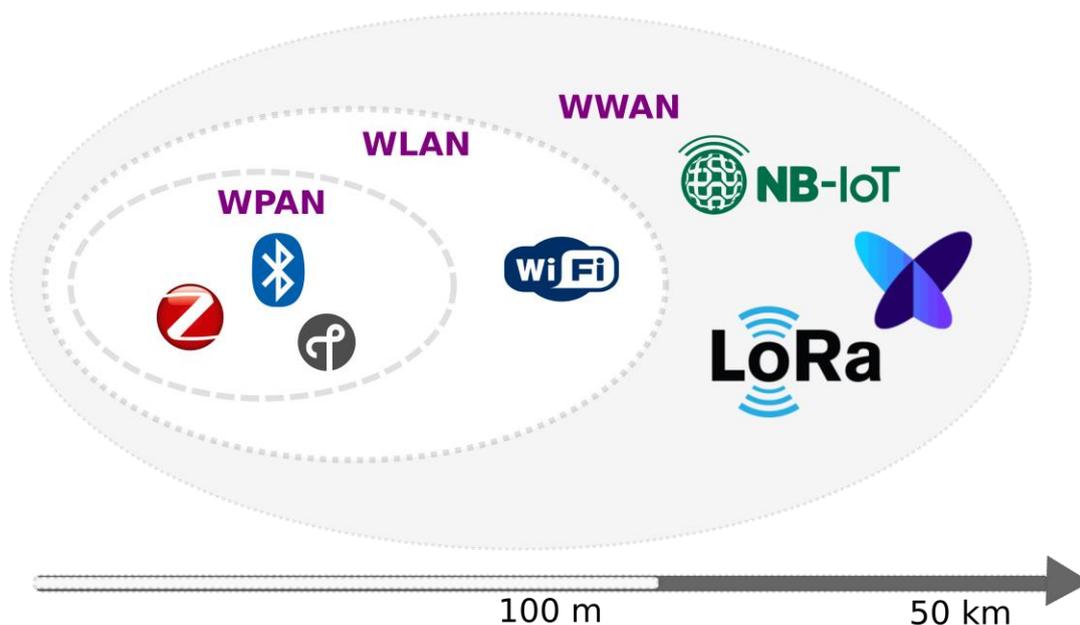


Figura 1.1: Cobertura das tecnologias de rede sem fio.

dados enviados dos dispositivos finais dentro da sua área de cobertura. Finalmente, LoRaWAN é um protocolo aberto de comunicação para a rede que funciona sobre a camada física *Long Range* (LoRa). Alcança uma distância entre 2 km até 40 km na faixa de 915 MHz (faixa destinada para o Brasil e homologada pela Anatel) e uma taxa de dados de 0,3 até 50 kbps. Uma das maiores vantagens dessa tecnologia é não ser necessário contratar uma operadora, possibilitando desenvolver diversas aplicações em cenários IoT de forma independente.

Tabela 1.1: Comparação entre as redes sem fio.

Protocolo	Alcance	Frequência	Taxa	Topologia
Wi-Fi	50 m	2,4/5 Ghz	1.300 Mbps	Estrela
BLE	80 m	2,4 Ghz	1 Mbps	Estrela/Malha
ZigBee	100 m	868/915 MHz/2,4 GHz	250 kbps	Estrela/Malha
NB-IoT	35 km	700 MHz/1,8/2,1/2,5 GHz	170-250 kbps	Estrela
SigFox	3-50 km	868/902 MHz	10-100 bps	Estrela
LoRaWAN	2-40 km	915 MHz (Brasil)	0,3-50 kbps	Estrela

Conforme apresentado em [25], a conectividade LoRa/LoRaWAN vem servindo de base para aplicações no contexto de cidades inteligentes, *smart grids*, agronegócio, saúde, localização, indústrias, entre outros. O mesmo trabalho apresenta que é crescente a quantidade de trabalhos científicos e práticos publicados envolvendo a tecnologia ao redor do globo. Como observado pelos autores os resultados de alcance da tecnologia LoRaWAN são influenciados pela topologia, densidade urbana, tipo de hardware, antena, modo de gerenciamento de pacotes, entre outros.

1.2. LoRa

LoRa é uma tecnologia de transmissão sem fio desenvolvida e patenteada pela empresa Semtech. LoRa realiza a modulação de radiofrequência (RF) baseada no espalhamento de espectral de chirp (*Chirp Spread Spectrum - CSS*). Essa técnica é utilizada na comunicação militar e espacial permitindo cobrir longas distâncias de comunicação e alcançar robustez à interferência [14]. Uma mensagem LoRa pode ser de *uplink* (enviada pelo dispositivo final) ou *downlink* (recebida pelo dispositivo final). A estrutura da mensagem é semelhante em ambos os casos, no entanto apenas a mensagem de *uplink* adiciona um código de verificação (CRC) que garante a integridade da carga útil (*PHYPayload*) [15].

No Brasil, a ANATEL publicou em agosto de 2018 o Ato nº 6.506 que aprova os procedimentos para avaliação da conformidade de equipamentos de radiocomunicação de radiação restrita, permitindo a operação de dispositivos LoRa no território nacional. O padrão adotado foi o australiano, ele utiliza a faixa de 915 MHz e compreende a faixa de 902 MHz a 907,5 MHz e 915 MHz a 928 MHz [33]. O padrão australiano possui 72 canais para *uplink* e 8 para *downlink*. Os canais de *uplink* vão de 0 a 63 utilizam uma largura de banda de 125 kHz com uma taxa de codificação 4/5, com início em 915,2 MHz sendo incrementado linearmente em 200 kHz até 927,8 MHz, já os canais de 64 a 71 possuem uma largura de banda de 500 kHz a partir de 915,9 MHz e incrementando linearmente de 1,6 MHz a 927,1 MHz. Para *downlink* os canais de 0 a 7 com uma largura de banda de 500 kHz, começando em 923,3 MHz e incrementando linear em 600 kHz até 927,5 MHz.

Para utilizar a modulação LoRa em um dispositivo final é preciso configurar três parâmetros: (i) Largura de Banda (*Bandwidth, BW*): com um dos tem três valores determinados, 125 kHz, 250 kHz ou 500 kHz, sofrendo um deslocamento de até 20% que não influenciará na decodificação [15]. (ii) Fator de Espalhamento (*Spreading Factor, SF*): determina o número de chirps necessários para representar um símbolo (um ou mais bits de dados), apresentando 2^{SF} valores possíveis de 7–12, quanto maior o SF mais energia será usada por bit e maior será o alcance entre transmissor e receptor [15]. (iii) Taxa de Codificação (*Coding Rate, CR*): define o número de bits destinados para dados de redundância na mensagem, a fim de realizar a recuperação de erros, definidos com os valores do CR: 4/5, 4/6, 4/7 e 4/8, incrementando o CR aumenta a proteção, mas também o tempo do bit no ar [15]. A Tabela 1.2 apresenta um exemplo de configuração para maior alcance e outro com o tamanho da mensagem maior [15]. É possível utilizar uma calculadora de *airtime* LoRaWAN para realizar as configurações para os dispositivos finais¹.

1.3. LoRaWAN

LoRaWAN é um protocolo aberto de comunicação que funciona sobre a camada física LoRa. Ele é mantido pela associação de empresas LoRa Alliance. A Figura 1.2 mostra a estrutura típica de uma rede LoRaWAN [15]. Os dispositivos finais (*End Nodes*) são objetos que possuem sensores ou atuadores. O *gateway* conecta os dispositivos na rede LoRaWAN. O servidor de rede gerencia a comunicação dos dispositivos com o servidor de aplicação através da Internet. O servidor de aplicação (*Application Ser-*

Tabela 1.2: Exemplo para configuração na frequência de 915 MHz.

Parâmetros	Maior Alcance	Maior Mensagem
<i>Data Rate (DR)</i>	0	4
<i>Spreading Factor (SF)</i>	SF10	SF8
<i>Bandwidth (BW)</i>	125 kHz	500 kHz
<i>Coding Rate (CR)</i>	4/5	4/5
<i>Bit Rate (BR)</i>	976 bps	12.500 bps
<i>Payload máximo</i>	11 bytes	242 bytes
<i>Air Time</i>	371 ms	175 ms

ver) exibe as informações vindas dos dispositivos para o usuário final [14]. Os elementos da arquitetura LoRaWAN são descritos a seguir.

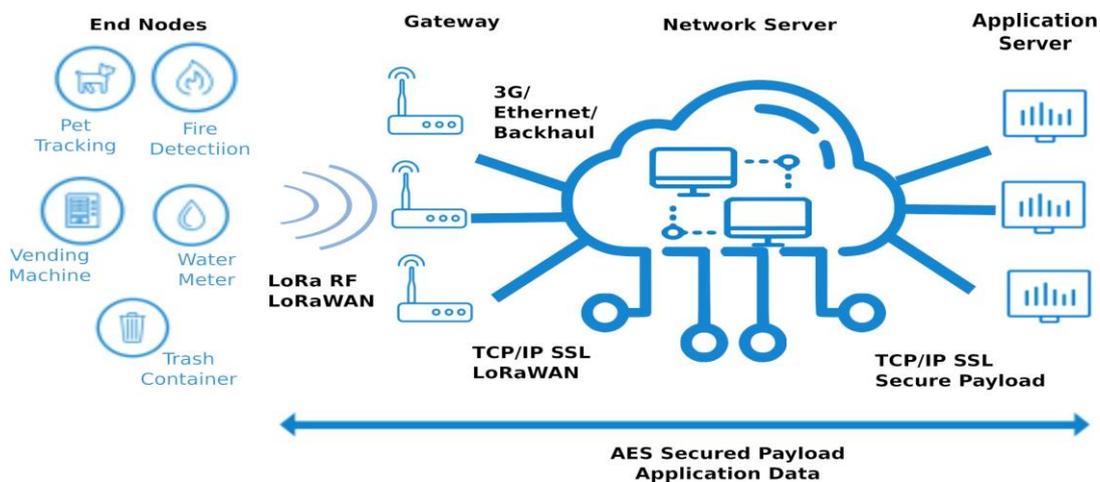


Figura 1.2: Visão geral da arquitetura LoRaWAN. Baseada em [14].

1.3.1. DISPOSITIVOS FINAIS

Os dispositivos finais coletam informações do ambiente, como temperatura, umidade, posição geográfica, batimentos cardíaco e outros, tanto em ambientes abertos como fechados. As informações são enviadas diretamente para um ou mais *gateways* da rede. Os dispositivos podem transmitir em um canal a qualquer momento, já para a próxima transmissão o canal deverá ser alterado respeitando a duração máxima de transmissão conforme as regulamentações locais.

Os *end nodes* podem assumir três configurações [15]. (i) Classe A: configuração obrigatória nas demais classes, nela o dispositivo final inicia comunicação com o *gateway* a qualquer momento, realizando uma transmissão e abrindo duas janelas de recepção de dados do *gateway*. (ii) Classe B: a comunicação é iniciada pelo *gateway*, que determina o momento de transmissão do dispositivo final e na sequência abre duas janelas de recepção. (iii) Classe C: o dispositivo final realiza uma transmissão de dados, abre duas janelas de recepção e mantém uma aberta até a próxima transmissão, fazendo com

que o dispositivo final necessite de uma fonte constante de energia. LoRaWAN oferece criptografia através do padrão AES em dois níveis: o primeiro está localizado na camada de rede garantindo a autenticidade do dispositivo final e o segundo está na camada de aplicação e garante que a informação gerada no dispositivo somente será conhecida pelo servidor de aplicação na comunicação entre dispositivos finais e a aplicação. Há dois tipos de autenticação: *Over The Air Activation (OTAA)* e *Activation By Personalization (ABP)*. A primeira é mais segura porque é o próprio dispositivo que gerencia sua ativação realizando um procedimento de conexão para ser ativado em uma rede selecionada. Já na ativação ABP o processo é mais simplificado pois os dispositivos finais estão diretamente vinculados a uma rede específica, ignorando o procedimento de entrada existente na OTAA [16]. A Figura 3 mostra os dois tipos de autenticação dos dispositivos finais.

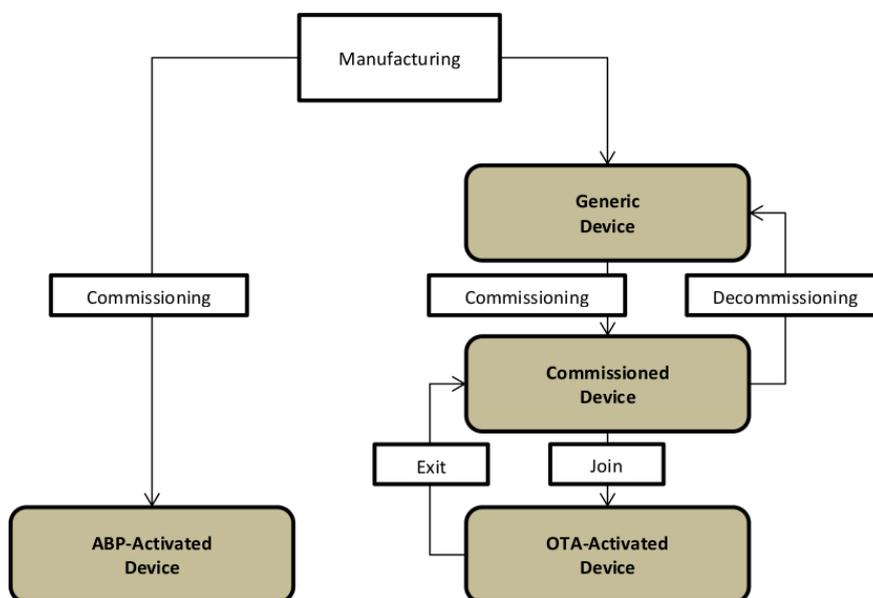


Figura 1.3: Autenticação por ABP e OTAA [16].

O desenvolvimento de dispositivos finais pode ser realizado através de placas de desenvolvimento que já incluem LoRaWAN [18] ou placas de prototipagem rápida como ESP32, Arduino, Raspberry Pi e outras, juntamente com módulos que possuem um ou vários sensores (temperatura, umidade, distância, GPS, etc.) e módulos transmissores [1]. Também é possível adquirir dispositivos finais prontos com sensores e atuadores de acordo com a finalidade do projeto [17]. No Brasil encontramos módulos LoRaWAN homologados pela Anatel fabricados por empresas nacionais ou importados. [31].

1.3.2. GATEWAYS

O *gateway*, concentrador ou ponto de acesso, é responsável por estabelecer a conexão com vários dispositivos finais via RF usando a tecnologia de transmissão LoRa e o protocolo LoRaWAN, assim pode receber as informações dos dispositivos finais, organizá-la e

repassá-la via TCP/IP (rede *ethernet*, Wi-Fi, 3G/4G) para o servidor de rede LoRaWAN. Toda a comunicação na rede LoRaWAN é bidirecional, mas o *uplink* é o tráfego predominante e o *downlink* é utilizado quando é necessário enviar alguma informação para os dispositivos finais como parâmetros de configuração por exemplo [14].

Podemos encontrar *gateways* comerciais completos com um ou vários canais [19], ou apenas módulos concentradores para adicionar em placas de prototipagem como a Raspberry Pi e criar um *gateway* com maior liberdade para configurar e monitorar o tráfego [30].

1.3.3. SERVIDORES DE REDE

O servidor de rede é responsável pelo gerenciamento da rede LoRaWAN: recebe as informações dos *gateways*, elimina os pacotes duplicados e então encaminha-os para os servidores de aplicação. Também é responsável por descriptografar os pacotes de *uplink* validando ou descartando em caso de inconsistência. Além disso, envia para os dispositivos finais as configurações do *data rate* e a potência de transmissão. Para soluções de servidores de rede há várias opções, dentre elas duas destacam-se:

- ChirpStak: fornece componentes de código aberto para redes LoRaWAN. O servidor apresenta uma solução pronta para uso, incluindo uma interface Web para gerenciamento de dispositivos e APIs para integração. É necessário instalar a solução em um servidor local ou em nuvem [7];
- The Things Network (TTN) ou The Things Stack (TTS): fornece um conjunto de ferramentas abertas e uma infraestrutura de rede LoRaWAN pronta para uso. Realiza o cadastro de *gateways*, dispositivos finais e faz a integração com servidores de aplicação [40]

1.3.4. SERVIDORES DE APLICAÇÃO

O servidor de aplicação é responsável por receber os pacotes vindos do servidor de rede e apresentar os dados monitorados dos dispositivos finais. Nesse segmento os *payloads* são armazenados, descriptografados e os dados brutos são transformados em informação útil para o usuário que pode visualizar através de *softwares* específicos. Essa configuração permite que os servidores de rede e de aplicação sejam instalados na mesma máquina ou em equipamentos separados no mesmo local ou em pontos geograficamente distantes. Existem diversos servidores de aplicação com as mais diversas funcionalidades, destacando-se:

- TagoIO: plataforma Web totalmente *cloud* com ferramentas para gerenciar dispositivos finais, armazenar dados, executar análises e integrar serviços [13];
- AllThingsTalk: além de exibir as informações dos dispositivos finais, possui vários *Software Development Kit* (SDK) para plataformas de prototipagem como Arduino e para as linguagens de programação Java, Python e Go [2];
- Cayenne: servidor de aplicação desenvolvido pela empresa myDevice. Tem como destaque a integração com as placas de prototipagem Arduino e Raspberry Pi [23].

Todos esses servidores de aplicação possuem uma *Application Programming Interface* (API) para auxiliar no desenvolvimento de soluções para diversos cenários. É importante salientar que os servidores de aplicação oferecem também planos de uso gratuitos.

As seções seguintes apresentam três atividades práticas relacionadas a comunicação LoRa e LoRaWAN.

1.4. ATIVIDADES PRÁTICAS

Esta seção apresenta os experimentos práticos para ilustrar o processo de desenvolvimento de uma aplicação LoRaWAN. Inicialmente são apresentados os dispositivos necessários, assim como os requisitos de *software*. Na sequência são descritas as práticas utilizando a tecnologia LoRa. Os primeiros experimentos têm foco na camada física realizando a comunicação somente com LoRa de forma ponto a ponto (P2P). Por fim é realizado um experimento utilizando um *gateway* mais robusto conectando-se ao servidor de rede TTSe a um servidor de aplicação.

1.4.1. DISPOSITIVOS

O dispositivo final responsável por realizar a comunicação LoRa durante as práticas é um Heltec WiFi LoRa 32 V2, visto na Figura 1.4. Integra um microprocessador ESP32, contando assim com todas as funcionalidades típicas como WiFi, Bluetooth, ADC (*Analog to Converter*), DAC (*Digital to Analog Converter*) e etc., além do módulo integrado LoRa e um *display* OLED.



Figura 1.4: Heltec WiFi LoRa 32 V2 [12]

As especificações do Heltec são comparadas com outros dispositivos semelhantes e são apresentadas na Tabela 1.3. Devido ao ESP32, o Heltec é superior ao Arduino em processamento, mas ainda limitado em relação ao Raspberry Pi.

Outro dispositivo a ser utilizado é um *gateway* composto por um Raspberry Pi 3

Tabela 1.3: Comparação entre dispositivos.

	Arduino Uno R3	ESP32	Raspberry Pi 3 B+
CPU	ATmega328	Xtensa LX6 DualCore	BCM2837B0 QuadCore
Clock	20 MHz	240 Mhz	1.4 Ghz
Arquitetura	8 bit	32 bit	64 bit
RAM	2 KB SRAM	520 KB SRAM	1 GB SDRAM

B+ e um módulo HAT que é um *gateway* LoRaWAN da Radioenge, visto na Figura 1.5.0 módulo possui GPS integrado, oferecendo geolocalização e sincronia de tempo com uma precisão de microsegundos.

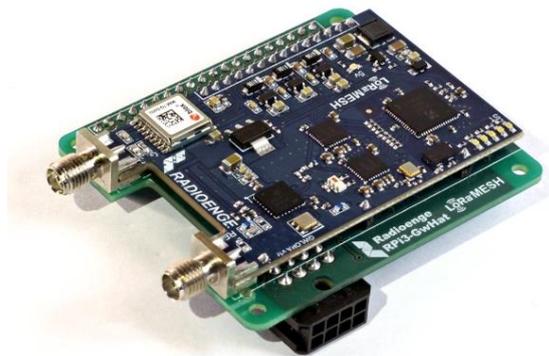


Figura 1.5: *Gateway* LoRaWAN Radioenge RD43HAT [30].

1.4.2. THE THINGS STACK

O *gateway* conecta-se a rede através da *The Things Stack* (TTS). TTS é uma rede colaborativa que conecta *gateways* LoRa [39]. É uma plataforma gratuita e aberta que conta com várias funcionalidades e integrações a serem utilizadas por aplicações desenvolvidas.

1.4.3. PROGRAMAÇÃO

A programação do dispositivo final pode ser em C, C++ e Python. Pode-se utilizar os *frameworks* Arduino ou Espressif ESP-IDF. A comunicação LoRa disponível para o dispositivo é por *firmware* e distribuído em forma de bibliotecas, LMIC² e arduino-LoRa. LMIC foi criada pela IBM, que descontinuou o projeto. Sendo assumido pela comunidade disponibilizando o código de forma aberta no GitHub. A arquitetura de um *firmware* que utiliza LMIC pode ser vista na Figura 1.6. Sendo a camada do microcontrolador a camada de *hardware*, contendo o módulo de radiofrequência LoRa, sensores e atuadores. A camada de *software* utiliza o *framework* do Arduino, sendo composta pela lógica da aplicação, os *drivers* dos componentes e da biblioteca LMIC. A biblioteca LMIC é responsável por seguir os protocolos LoRaWAN e *Medium Access Control* (MAC), manter o ambiente de execução e fazer a abstração de *hardware*

com relação aos transmissores. Já a biblioteca arduino-LoRa implementa a camada física da LoRaWAN, podendo ser utilizada de forma independente sendo a camada de controle MAC de responsabilidade do desenvolvedor.

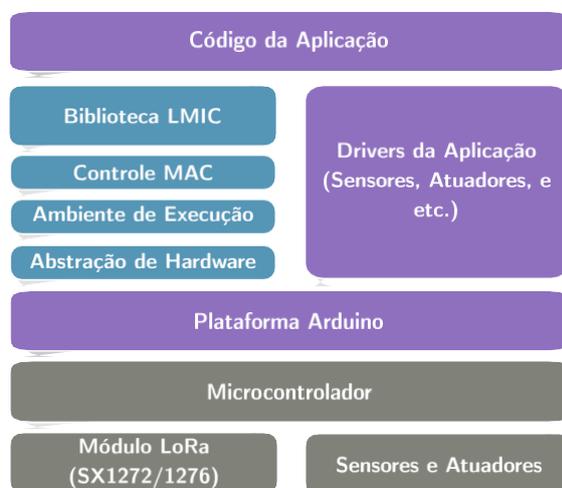


Figura 1.6: Arquitetura em camadas de uma aplicação LMIC. Baseada em [22].

1.4.3.1. IDE

A IDE (*Integrated Development Environment*), ambiente de desenvolvimento, a ser utilizado durante as práticas é o VSCode³ junto da extensão PlatformIO. É um ambiente simples e que permite uma rápida adaptação e fornece uma estrutura de projeto organizada, possuindo funcionalidades de depuração e testes unitários.

A instalação também é bem simples. Uma vez tendo o VSCode instalado, basta acessar a aba de extensões, procurar por *PlatformIO* e clicar em *Install*. A Figura 1.7 apresenta o processo de instalação da extensão indicando com números os passos a serem realizados.

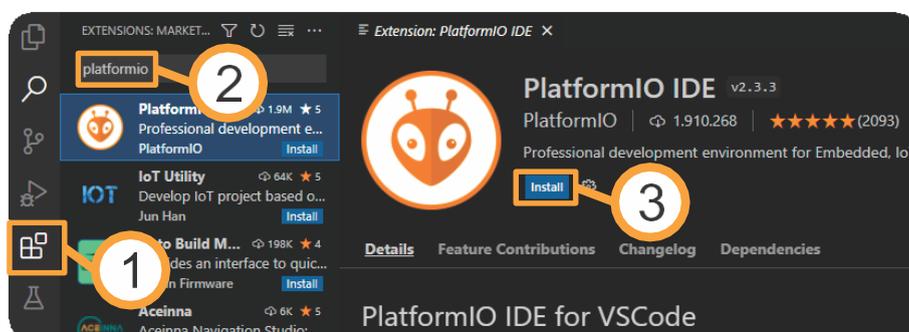


Figura 1.7: Instalação da extensão PlatformIO IDE.

Após instalação do PlatformIO será necessário reiniciar o VSCode. Na próxima inicialização a extensão estará habilitada como visto na Figura 1.8.

³Disponível em <https://code.visualstudio.com/>

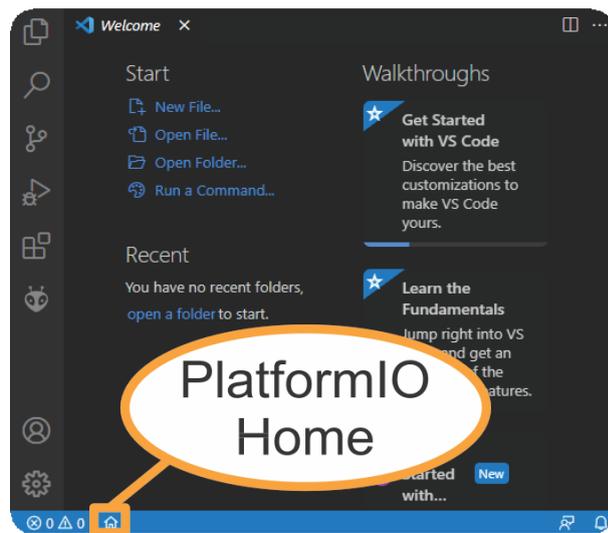


Figura 1.8: Inicialização do PlatformIO IDE.

Para criar um novo projeto basta clicar no botão *Home* e então *New Project*, surgindo uma tela semelhante a Figura 1.9. Onde pode-se definir o nome do projeto, a placa de desenvolvimento e o *framework* a serem utilizados.

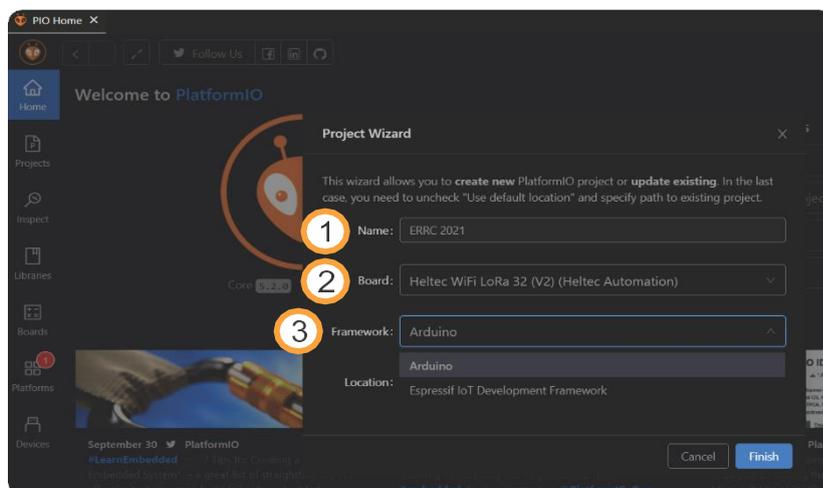


Figura 1.9: Painel de novo projeto do PlatformIO IDE.

Com o projeto criado podemos ver a estrutura do projeto e os botões importantes como na Figura 1.10. Onde em *lib* ficam localizadas as bibliotecas a serem utilizadas. Em *include* os arquivos de cabeçalho. Os arquivos principais se encontram no diretório *src*. Com relação aos botões, *Build* compila o *firmware* e monta o arquivo a ser gravado na placa. *Upload* faz a gravação de fato. E *Serial Monitor* disponibiliza uma interface serial com o microcontrolador.

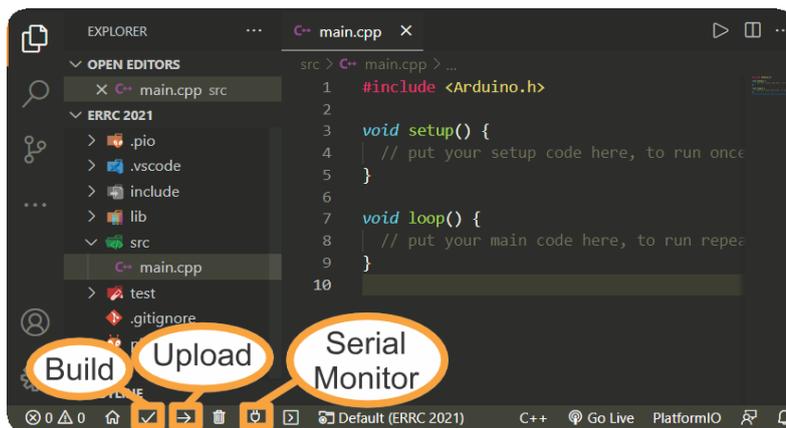


Figura 1.10: Botões importantes do PlatformIO IDE.

1.4.4. PRÁTICA 1: COMUNICAÇÃO LORA PONTO A PONTO

Esta prática tem o objetivo de demonstrar o funcionamento da comunicação LoRa utilizando somente a camada física. Será utilizada a biblioteca `arduino-lora` que está contida na biblioteca do dispositivo, `Heltec_ESP32`⁴ disponibilizada pela Heltec Automation. Para instalar a biblioteca basta acessar a aba de *Libraries* como na Figura 1.1 e pesquisar o termo “Heltec”, ou instalar manualmente baixando a biblioteca e colocando-a na pasta `lib`.

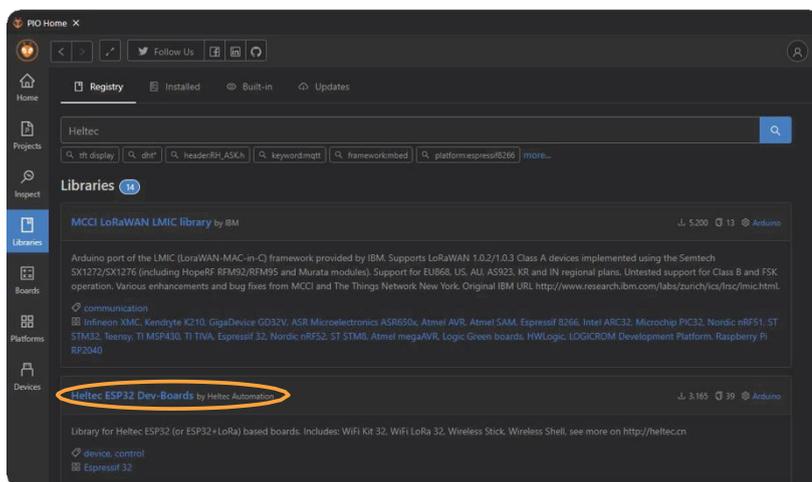


Figura 1.11: Biblioteca Heltec utilizada.

Agora basta executar o mesmo procedimento para instalar as bibliotecas *DHT sensor library* e *Unified Sensor* da Adafruit, bem como a biblioteca *SDS011 Sensor Library* de R. Zschiegner.

Esta prática, durante a apresentação, será dividida em três etapas. A primeira con-

⁴Disponível em: https://github.com/HelTecAutomation/Heltec_ESP32

siste de uma comunicação simples entre os dois dispositivos enviando um pacote LoRa com um contador e alguns dados relativos a intensidade do sinal. Na segunda etapa incluiremos o sensor de umidade e temperatura DHT11 e enviaremos os dados para o receptor. Por fim incluiremos o sensor SDS011, fabricado pela Nova Fitness. Os códigos completos estão disponibilizados em um repositório no GitHub⁵.

1.4.4.1. LISTAGENS DE CÓDIGO

Listagem 1 *Includes & Defines*

```
1  #include <Arduino.h>
2  #include "heltec.h"
3
4  #define BAND    915E6
   String rssi = "RSSI --";
   String packSize = "--";
   String packet;
```

Após a importação das bibliotecas necessárias e a definição das variáveis globais, a função `setup` inicializa o dispositivo Heltec e faz as configurações iniciais do *display* e do rádio LoRa.

Listagem 2 *Setup do receptor*

```
1  void setup() {
2
3      Heltec.begin(true, true, true, true, BAND);
4      Heltec.display->init(); //Inicia o display
5      Heltec.display->flipScreenVertically();
6      Heltec.display->setFont(ArialMT_Plain_10);
7      delay(1500);
8
9      Heltec.display->clear();
10
11     Heltec.display->drawString(0, 0, "LoRa iniciado com sucesso!");
12     Heltec.display->drawString(0, 10, "Esperando dados...");
13     Heltec.display->display();
```

Já a função `loop` aguarda a chegada do pacote LoRa. Uma vez recebido o pacote chama as funções responsáveis por formatar e apresentar os dados no *display*.

Com todas estas funções definidas, o *upload* para o dispositivo receptor pode ser feito, Caso o dispositivo seja ligado sem que exista um *sender* configurado e enviando pacotes o receptor ficará retido no *loop*, sem entrar na função `cbk`. Apenas o *display* do receptor ficará ligado, uma vez que o SDS011 utiliza a mesma entrada responsável pelo *reset* do OLED no *sender*

As bibliotecas inclusas no dispositivo responsável por enviar os dados diferem um pouco do *receiver*, será necessário utilizar as bibliotecas do DHT11 e do SDS011, assim como utilizar algumas variáveis globais e instanciar as devidas classes.

Listagem 3 Outras funções do receptor

```
1 void loop() {
2     int packetSize = LoRa.parsePacket();
3     if (packetSize)
4         cbk(packetSize);
5     delay(10);
6 }
7
8 void LoRaData() {
9     Heltec.display->clear();
10
11     Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
12     Heltec.display->setFont(ArialMT_Plain_10);
13
14     Heltec.display->drawString(0, 15, "Received " + packetSize + " bytes"); //apenas em 1.4.4.1
15     Heltec.display->drawString(0, 0, rssi);
16
17     Heltec.display->drawString(0, 15, packet);
18     Heltec.display->display();
19 }
20
21
22
23
24
25
```

Listagem 4 Includes & Defines do sender

```
1 #include <Arduino.h>
2 #include "heltec.h"
3
4 #include <Adafruit_Sensor.h>
5 #include <DHT.h>
6
7 #include <DHT_U.h>
8 #include <SDS011.h>
9
10 #define BAND 915E6
11 #define DHTPIN 3
12 #define DHTTYPE DHT11
13
14 DHT dht(DHTPIN, DHTTYPE);
15
16 SDS011 my_sds;
```

A função `setup` é responsável por iniciar todos os objetos. Desta vez na função `Heltec.begin()` iremos utilizar o valor de `false`, já que o `display` não será iniciado. A função `loop` é responsável por montar e enviar o pacote LoRa com os dados da temperatura, umidade e de qualidade do ar (PM10 e PM2.5).

Após a definição de todas estas funções podemos fazer o *upload* do código para o dispositivo responsável por enviar os pacotes. O SDS011 demora cerca de um minuto para fazer a leitura dos dados de PM10 e PM2.5. Portanto, caso o receptor esteja ligado, dentro de aproximadamente um minuto o `display` será atualizado com as informações

Listagem 5 Setup & Loop do sender

```
1 void setup() {
2   Serial.begin(9600);
3
4   Heltec.begin(false, true, true, true, BAND);
5   dht.begin();
6
7   my_sds.begin(&port);
8   pinMode(25, OUTPUT);
9 }
10
11 void loop() {
12   float h = dht.readHumidity();
13   float t = dht.readTemperature();
14   err = my_sds.read(&p25, &p10); if
15   (!err) {
16     Serial.println("Ok");
17     LoRa.beginPacket();
18     LoRa.print("Umi: ");
19     LoRa.print(h);
20     LoRa.println("%");
21
22     LoRa.print("Temp: ");
23     LoRa.print(t);
24     LoRa.println("Graus celsius");
25     LoRa.print("PM25: ");
26 }
```

temperatura, umidade, PM10, PM2.5 e RSSI.

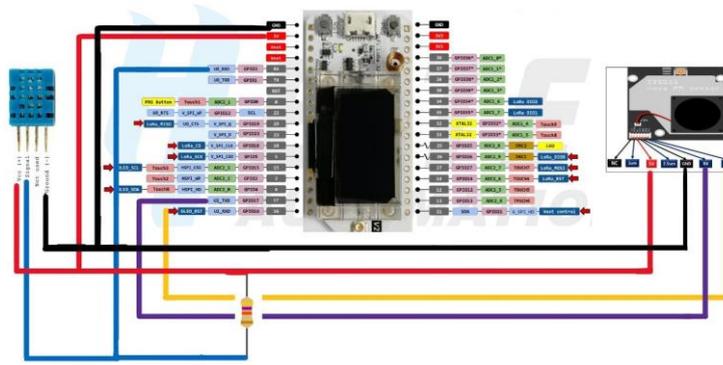


Figura 1.12: Diagrama esquemático da prática 1.

1.4.5. PRÁTICA 2: COMUNICAÇÃO LORAWAN COM SERVIDOR DE REDE

A comunicação com um servidor de rede TTS é possível utilizando um *gateway* de ao menos 4 canais. Uma alternativa de baixo custo era a programação de um *gateway* “caseiro” de um canal com algum dispositivo final atuando como *gateway*. Mas, recentemente a plataforma TTN recebeu uma atualização, passando para TTS. Esta nova plataforma é escala e soluciona alguns problemas encontrados na sua versão anterior. Porém,

os *gateways* simples passaram a não funcionar mais.

Portanto, nesta prática será utilizado o *gateway* comercial homologado pela Ana-tel para demonstrar a comunicação com o servidor de rede TTS. O próprio fabricante disponibiliza um tutorial demonstrando como realizar a instalação do *firmware* do *gateway* e como registrá-lo na rede. Como a plataforma TTS sofreu mudanças recentemente o seu registro será abordado.

Acessando a TTS⁶ é possível realizar o *login*, ou cadastro, na plataforma. Uma vez feita a autenticação o console é exibido apresentando as seções de aplicações e *gateways*. Para registrar um *gateway* entramos na área dos *gateways* e clicamos em *Add Gateway*. Neste momento três parâmetros devem ser configurados, *Gateway-ID*, *Gateway EUI* e *Frequency Plan*. *Gateway-ID* se refere a um identificador único que vai dar nome ao *gateway*. *Gateway EUI* também é um identificador único do dispositivo, mas este é fornecido pelo fabricante. Já *Frequency Plan* está relacionado ao plano de frequência a ser usado, no caso Australia 915-928 MHz, FSB2. O processo também pode ser visualizado no vídeo no YouTube⁷.

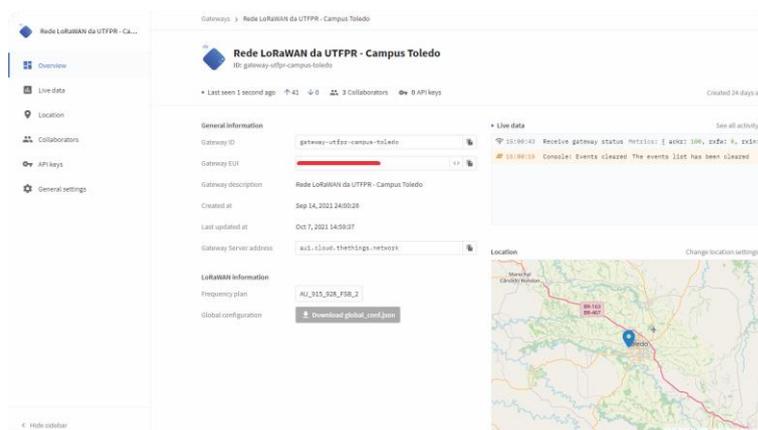


Figura 1.13: Console *gateway*.

Com o *gateway* configurado é momento de adicionar uma aplicação. Na seção *Applications* clique em *Add Application*. De forma bem simples preencha os campos como informado, aqui somente o *Application ID* é obrigatório.

No painel da aplicação precisamos adicionar um dispositivo final. Basta clicar em *Add End Device*. Mudamos para configuração manual e selecionamos a versão LoRaWAN e o plano de frequência. As chaves podem ser geradas automaticamente, destaque para AppEUI que deve ter pelo menos um número diferente de zero. Esse processo pode ser visto na Figura 1.14.

Para a programação do dispositivo será utilizada a última versão da LMIC. Podendo ser instalada tanto pelo gerenciador de bibliotecas quanto manualmente. O código

Register end device

From The LoRaWAN Device Repository [Manually](#)

LoRaWAN version ⓘ *

MAC V1.0.3 | v

Regional Parameters version ⓘ *

PHY V1.0.3 REV A | v

Frequency plan ⓘ *

Australia 915-928 MHz, FSB 2 (used by TTN) | v

[Show advanced activation, LoRaWAN class and cluster settings](#) v

DevEUI ⓘ *

70 B3 D5 7E D0 04 65 BE Generate 9/50 used

AppEUI ⓘ *

00 00 00 00 00 00 00 01 Fill with zeros

AppKey ⓘ *

B3 E9 8E 17 29 29 D5 1A B6 B0 21 A8 11 06 41 3F Generate

End device ID ⓘ *

errc-2021

This value is automatically prefilled using the DevEUI

After registration

View registered end device

Register another end device of this type

Register end device

Figura 1.14: Registrando dispositivo final.

completo pode ser encontrado no repositório do minicurso no GitHub, sendo necessário somente modificar as chaves para corresponder com o dispositivo criado como na Listagem 6 e configurar a frequência. Para configurar a frequência, dentro da pasta da biblioteca LMIC existe um arquivo chamado `lmic_project_config`, este arquivo deve ser alterado de forma a selecionar a frequência a ser usada pelo dispositivo, no caso, `CFG_au915`.

Outro ponto importante a ser considerado é a variável que armazena a mensagem a função que faz o envio da mensagem. Estamos enviando uma mensagem contendo a frase *"Hello World!"* atribuída à variável `payload`. A rotina responsável pelo envio da mensagem é `do_send`. Ela é configurada para um envio periódico, que pode ser alterado através da variável `TX_INTERVAL`.

Uma vez que o dispositivo foi programado basta ligá-lo e o tráfego de mensa-

Listagem 6 Chaves de autenticação dos dispositivos

```
1 // AppEUI em formato LSB
2 static const u1_t PROGMEM APPEUI[8] = {0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
3
4 // DevEUI em formato LSB
5 static const u1_t PROGMEM DEVEUI[8] = {0xBE, 0x65, 0x04, 0xD0, 0x7E, 0xD5, 0xB3, 0x70};
```

gens começará a aparecer na console TTS. Inicialmente os dados serão mostrados em forma de *bytes*, portanto faz-se necessário a decodificação da mensagem. Acessando a *abaPayload Formatters - Uplink* podemos incluir um código para formatar a mensagem. O código pode ser visualizado na Listagem 8, onde aplicamos a transformação para Strings somente aos *bytes* do *payload* completo, *input*.

Listagem 7 Payload Formatter bytes para string

```
1 function decodeUplink(input) {
2   return {
3     data: {
4       msg: String.fromCharCode.apply(null, input.bytes)
5     },
6     warnings: [],
7     errors: []
8   };
9 }
```



Figura 1.15: Tráfego de mensagens no servidor de rede TTS.

Exemplificando como enviar as mensagens para um servidor de aplicação utilizaremos a plataforma Tagolo ⁸. Para isso, acessamos o *site* e geramos uma nova chave de autorização, em *Devices* clicando em *Authorization*. Nesta seção pode-se dar um nome a chave e clicar em *Generate*. Esta chave é necessária para realizar a integração entre TTS e Tagolo. Ainda em *Devices*, adicionamos um novo dispositivo clicando em *Add Device*

Selecione a opção de dispositivo customizado presente em *LoRaWAN TTI/TTN v3*, atribuímos um nome ao dispositivo e colocamos o seu respectivo *DevEUI*. O painel *Devices* pode ser visto na Figura 1.16, onde estão destacados os botões principais do processo.

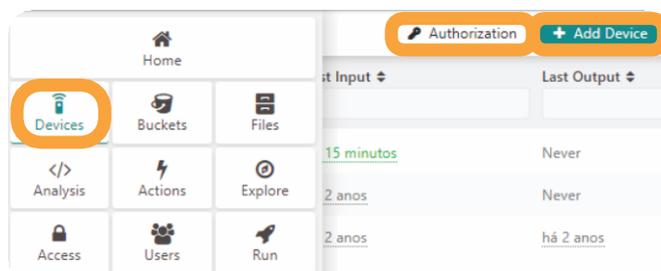


Figura 1.16: Seção de dispositivos da plataforma TagoIO.

De volta a TTS, adicionamos um *WebHook* para a TagoIO em *Integrations*, selecionando a opção TagoIO, como demonstrado na Figura 1.17. Neste momento, inserimos um nome para a integração e a chave gerada anteriormente pela TagoIO. Finalizando o processo de criação da integração ao clicar em *Create tagoio webhook*.

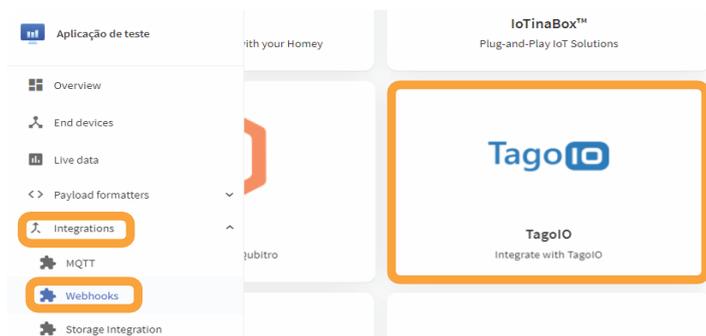


Figura 1.17: Seção de integrações *Webhooks* da TTS.

Alterando o código para enviar um valor de um contador conforme pressionamos um botão e também *Payload Formatter* para decodificar os *bytes* em um número inteiro, como na Listagem 8.

Nesta etapa a plataforma TagoIO fornece a criação de painéis. Estes painéis são formados por diferentes *widgets*, que acabam por exibir as informações recebidas do dispositivo. Para criar um painel clicamos em *Add Dashboard*, destacado na Figura 1.18, assim como o botão *Add Widget* que deve ser selecionado em seguida. Após clicar em *Add Widget* as opções de *widget* serão mostradas, como na Figura 1.19.

Listagem 8 Payload Formatter bytes para inteiro

```
1 function decodeUplink(input) {  
2   return {  
3     data: {  
4       counter: (input.bytes[0] << 8) + input.bytes[1]  
5     },  
6     warnings: [],  
7     errors: []  
8   };  
9 }
```

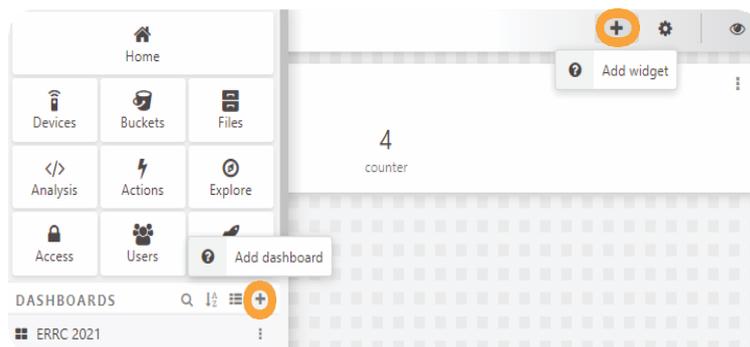


Figura 1.18: Adicionar painel e adicionar *widget* da TagIO.

Como a variável a ser exibida é do tipo numérico, adicionamos um *Display*. Ao selecionar o *widget* a ser adicionado um outro painel é aberto para configurá-lo. Este painel pode ser visualizado na Figura 1.20. Aqui selecionamos o dispositivo criado e a variável a ser exibida, no caso, *counter* que foi definida no Payload Formatter.

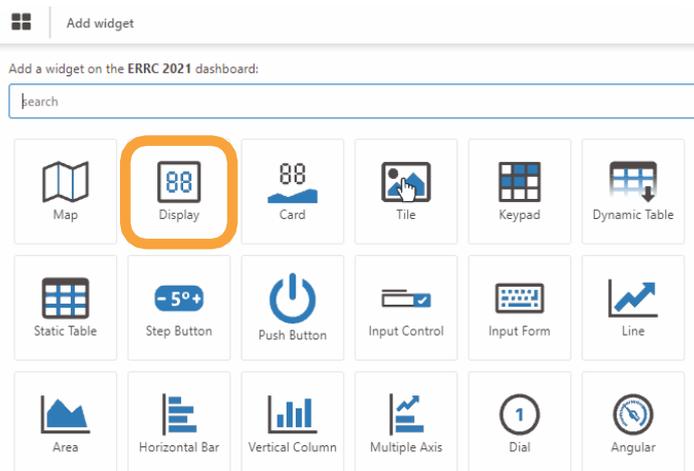


Figura 1.19: *Widgets* disponíveis na TagIO.

Desta forma, toda vez que o botão do ESP32 Heltec for pressionado, o valor de

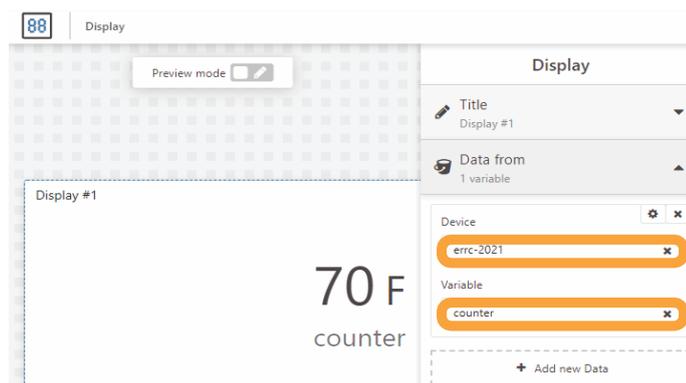


Figura 1.20: Configuração do *widget display*.

counter será incrementado e enviado para o servidor de rede TTS. O servidor decodifica e encaminha a mensagem para o servidor de aplicação, TagoIO, que por sua vez faz a exibição dos dados.

REFERÊNCIAS

- [1] Thiago Almeida et al. “Em Busca de um Algoritmo de Retransmissão em Redes LoRaWAN”. Em: Anais do V Workshop de Computação Urbana. Evento Online: SBC, 2021, pp. 70–83. URL: <https://sol.sbc.org.br/index.php/courb/article/view/17105>.
- [2] ALSO. AllThingsTalk. URL: <https://www.allthingstalk.com/>. Acesso: 4 set/2021.
- [3] ATC. REDE ATC LoRaWAN. URL: <https://americantower.com.br/pt/solu%C3%A7%C3%B5es/rede-neutra-loRaWAN.html>. Acesso: 7 out/2021.
- [4] Bain. Unlocking Opportunities in the Internet of Things. Online. Acessado em 23/06/2019. 2018. URL: <https://www.bain.com/insights/unlocking-opportunities-in-the-internet-of-things/>.
- [5] Edson Tavares de Camargo, Fabio Alexandre Spanhol e Álvaro Ricieri Castro e Souza. “Deployment of a LoRaWAN network and evaluation of tracking devices in the context of smart cities”. Em: Journal of Internet Services and Applications 12.8 (out. de 2021), pp. 1–24. ISSN: 1869-0238. DOI: 10.1186/s13174-021-00138-7.
- [6] M. Centenaro et al. “Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios”. Em: IEEE Wireless Communications 23.5 (out. de 2016), pp. 60–67. ISSN: 1558-0687. DOI: 10.1109/MWC.2016.7721743.
- [7] ChirpStack. The ChirpStack project. 2021. URL: <https://www.chirpstack.io/project/>. Acesso: 26 set/2021.
- [8] CSA. ZIGBEE - The Full-stack Solution Interlacing all your Smart Devices. URL: <https://zigbeealliance.org/solution/zigbee/>. Acesso: 22 out/2021.

- [9] Manishkumar Dholu e K.A. Ghodinde. "Internet of Things (IoT) for Precision Agriculture Application". Em: 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI). 2018, pp. 339–342. DOI: 10.1109/ICOEI.2018.8553720.
- [10] Ala Al-Fuqaha et al. "Internet of things: A survey on enabling technologies, protocols, and applications". Em: IEEE communications surveys&tutorials 17.4 (2015), pp. 2347–2376.
- [11] GSMA. NB-IoT Deployment Guide to Basic Feature Set Requirements. GSMA. 2019, p. 51. URL: <https://www.gsma.com/iot/wp-content/uploads/2019/07/201906-GSMA-NB-IoT-Deployment-Guide-v3.pdf>. Acesso: 07 out/2021.
- [12] Heltec Automation. WiFi LoRa 32 (V2). 2021. URL: <https://heltec.org/project/wifi-lora-32/>. Acesso: 22 set/2021.
- [13] Tago LLC. TagoIO. URL: <https://tago.io/>. Acesso: 4 set/2021.
- [14] LoRa Alliance. What is LoRaWAN? 2015. URL: <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>. Acesso: 28 oct/2021.
- [15] LoRa Alliance. LoRaWAN 1.1 Specification. LoRa Alliance. 2017, p. 101. URL: https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm_specification_v1.1.pdf. Acesso: 23 set/2021.
- [16] LoRa Alliance. LoRaWAN® Back-End Interfaces v1.0. 2017. URL: <https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm-backendinterfaces-v1.0.pdf>. Acesso: 17 out/2021.
- [17] LoRa Alliance. LoRaWAN Certified Device. 2021. URL: https://lora-alliance.org/showcase/search/?_sft_product_categories=end-node-sensors-orother-devices. Acesso: 26 set/2021.
- [18] LoRa Alliance. LoRaWAN Development Board. 2021. URL: https://lora-alliance.org/showcase/search/?_sft_product_categories=developmentboard,module. Acesso: 26 set/2021.
- [19] LoRa Alliance. LoRaWAN Gateway. 2021. URL: https://lora-alliance.org/showcase/search/?_sft_product_categories=gateway. Acesso: 26 set/2021.
- [20] Obaidulla Al-Mahmud et al. "Internet of Things (IoT) Based Smart Health Care Medical Box for Elderly People". Em: 2020 International Conference for Emerging Technology (INCET). 2020, pp. 1–6. DOI: 10.1109/INCET49848.2020.9153994.
- [21] Timothy Malche e Priti Maheshwary. "Internet of Things (IoT) for building smart home system". Em: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 2017, pp. 65–70. DOI: 10.1109/I-SMAC.2017.8058258.
- [22] MCCI. Arduino LoRaWAN MAC in C (LMIC). 2019. URL: <https://github.com/mcci-catena/arduino-lmic/blob/master/doc/LMIC-v3.0.99.pdf>. Acesso: 22 set/2021.
- [23] myDevices. Cayenne. URL: <https://developers.mydevices.com/cayenne/features/>. Acesso: 4 set/2021.
- [24] Hassan Noura et al. "LoRaWAN security survey: Issues, threats and possible mitigation techniques". Em: Internet of Things 12 (2020), p. 100303. ISSN: 2542-6605. DOI:

- <https://doi.org/10.1016/j.iot.2020.100303>. URL:
<https://www.sciencedirect.com/science/article/pii/S2542660520301359>.
- [25] Lucas de Oliveira, Arlindo da Conceição e Lauro S. Neto. “Revisão sistemática da literatura sobre aplicações das tecnologias LoRa e LoRaWAN”. Em: Anais Estendidos do VIII Simpósio Brasileiro de Engenharia de Sistemas Computacionais. Salvador: SBC, 2018. URL: https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/11002.
- [26] Javan de Oliveira Júnior e Marcio Oyamada. “Avaliando o impacto da compressão de dados no desempenho e energia em redes LoRa”. Em: Anais Estendidos do X Simpósio Brasileiro de Engenharia de Sistemas Computacionais. Evento Online: SBC, 2020, pp. 81–88. DOI: 10.5753/sbesc_estendido.2020.13094. URL: https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/13094.
- [27] André Pastório e Edson Camargo. “Técnicas de Geolocalização em Redes Lo-RaWAN como Abordagem de Tolerância a Falhas para Dispositivos IoT Baseados em GPS”. Em: Anais do XXII Workshop de Testes e Tolerância a Falhas. Evento Online: SBC, 2021, pp. 29–42. URL: <https://sol.sbc.org.br/index.php/wtf/article/view/17202>.
- [28] André Pastório, Luiz Rodrigues e Edson de Camargo. “Uma Revisão Sistemática da Literatura Sobre Tolerância a Falhas em Internet das Coisas”. Em: Anais Estendidos do X Simpósio Brasileiro de Engenharia de Sistemas Computacionais. Evento Online: SBC, 2020, pp. 57–64. DOI: 10.5753/sbesc_estendido.2020.13091. URL: https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/13091.
- [29] João Pastório, Edson Tavares de Camargo e Álvaro Ricieri Castro e Souza. “Simulação do monitoramento de veículos em uma rede LoRaWAN usando NS-3”. Em: ERRC 2021 - IC (). Out. de 2021. URL: <http://XXXXX/220180.pdf>.
- [30] Radioenge. Gateway LoRaWAN. 2021. URL: <https://www.radioenge.com.br/solucoes/iot/18-gateway-lorawan.html>. Acesso: 26 set/2021.
- [31] Radioenge. Módulo LoRaWAN. 2021. URL: <https://www.radioenge.com.br/solucoes/iot/17-modulo-lorawan.html>. Acesso: 26 set/2021.
- [32] Swathi Ramnath et al. “IoT based localization and tracking”. Em: 2017 International Conference on IoT and Application (ICIOT). 2017, pp. 1–4. DOI: 10.1109/ICIOTA.2017.8073629.
- [33] Jonas Rossato, Fabio Spanhol e Edson Camargo. “Implantação e Avaliação de uma Rede Sem-Fio de Longo Alcance e Baixa Potência para Cidades Inteligentes”. Em: Anais do IV Workshop de Computação Urbana. Rio de Janeiro: SBC, 2020, pp. 192–205. DOI: 10.5753/courb.2020.12363. URL: <https://sol.sbc.org.br/index.php/courb/article/view/12363>.
- [34] Sureshkumar Selvaraj e Suresh Sundaravaradhan. “Challenges and opportunities in IoT healthcare systems: a systematic review”. Em: SN Applied Sciences 2 (jan. de 2020). DOI: 10.1007/s42452-019-1925-y.
- [35] Sigfox. Sigfox Technology. 2021. URL: https://www.sigfox.com/en/whatsigfox/technology#id_technology. Acesso: 7 out/2021.
- [36] Marcos da Silva et al. “Avaliação de Dispositivos de Rastreamento em uma Rede LoRaWAN no Contexto de Cidades Inteligentes”. Em: Anais do IV Workshop de Computação Urbana. Rio de

Janeiro: SBC, 2020, pp. 1–14. DOI: 10.5753/courb.2020.12349. URL: <https://sol.sbc.org.br/index.php/courb/article/view/12349>.

[37] Statista. Forecast end-user spending on IoT solutions worldwide from 2017 to 2025. Acesso em 30/03/2020. 2020. URL: <https://www.statista.com/statistics/976313/global-iot-market-size/>.

[38] TCT. TCT Telecom. URL: <https://www.tctelecom.com/>. Acesso: 7 out/2021.

[39] The Things Industries. The Things Stack (TTS). 2021. URL: shorturl.at/klwTU. Acesso: 22 set/2021.

[40] TTN. The Things Network. URL: <https://www.thethingsnetwork.org/>. Acesso: 7 set/2021.

[41] Antonio Velez-Estevez, Lorena Gutiérrez-Madroñal e Inmaculada Medina-Bulo. “IoT-TEG 4.0: A New Approach 4.0 for Test Event Generation”. Em: IEEE Transactions on Reliability (2021), pp. 1–13. DOI: 10.1109/TR.2021.3087781.

[42] vXchnge. Comprehensive Guide to IoT Statistics You Need to Know in 2020. Acesso em 30/03/2020. 2020. URL: <https://www.vxchnge.com/blog/iot-statistics>.