SBrT 20

XXXVIII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS

LIVRO DE MINICURSOS

Florianópolis – SC

Evento Virtual

REALIZAÇÃO

SBrT

Fortaleza de São José da Ponta Grossa - Florianópolis - SC

editora IFPB

Livro de Minicursos
SBRT 2020

Editora

Instituto Federal de Ensino, Ciência e Tecnologia da Paraíba – IFPB

Organização

Edmar Candeia Gurjão (UFCG)

Realização

Sociedade Brasileira de Telecomunicações – SBrT

Florianópolis - SC
22 a 25 de novembro de 2020

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Nilo Peçanha – IFPB, campus João Pessoa

S612 Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (38. : 2020 : Florianópolis, SC).
Livro de minicursos SBRT 2020, 22 a 25 de novembro de 2020, organizador Edmar Candeia Gurjão. – João Pessoa : IFPB, 2021.
142p. : il.
PDF
Tamanho do arquivo 6,02 MB
ISBN 978-65-87572-23-9
Evento realizado pela Sociedade Brasileira de Telecomunicações (SBrT), 2020.
1.Telecomunicações. 2. Processamento de sinais.3. Redes IEEE.4. Sistema 5G. I. Gurjão, Edmar Candeia. II. Título.

CDU 621.39

Elaboração: Lucrecia Camilo de Lima - CRB 15/132

editora IFPB

CONTATO

Av. João da Mata, 256 - Jaguaribe. CEP: 58015-020, João Pessoa - PB.
Fone: (83) 3612-9722 | E-mail: editora@ifpb.edu.br

Organização do SBrT 2020

Coordenação Geral
Bartolomeu F. Uchôa Filho (UFSC)
Eduardo L. O. Batista (UFSC)

Presidentes de Honra
José C. M. Bermudez (UFSC)
Rui Seara (UFSC)

Coordenação Técnica
Cecilio Pimentel (UFPE)
Danilo Silva (UFSC)
Eduardo L.O. Batista (UFSC)

Coordenação Local
Márcio H. Costa (UFSC)
Richard D. Souza (UFSC)

Coordenação Financeira
Adão Boava (UFSC)
Ciro A. Pitz (UFSC)
Renato Machado (ITA)
Ugo S. Dias (UnB)

Coordenação de Divulgação
Cesar A. Azurdia Meza (Universidad de Chile, Chile)
Francisco Monteiro (Instituto de Telecomunicações, Lisboa)
Didier Le Ruyet (CNAM, França)

Diretoria da SBrT

Presidente
Cristiano Magalhães Panazio

Vice-Presidente de Atividades Técnicas
Cecilio José Lins Pimentel

Vice-Presidente de Finanças
Leonardo Tomazeli Duarte

Vice-Presidente de Desenvolvimento e Difusão
Eduardo Luiz Ortiz Batista

Vice-Presidente de Relações Externas
Ugo Silva Dias

Membros do Conselho
Lisandro Lovisolo (UERJ)
Edmar Candeia Gurjão (UFCG)
Charles Casimiro Cavalcante (UFC)
Renato da Rocha Lopes (Unicamp)
Eduardo Antônio Barros da Silva (UFRJ)

Prefácio

O XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT'2020) foi o primeiro na modalidade virtual da série de eventos realizados pela Sociedade Brasileira de Telecomunicaç oes desde a sua edição inaugural, em 1983. Sob o tema "Perspectivas do Sistema 5G para a Academia, o Governo, as Empresas e a População", o SBrT'2020 reuniu profissionais dos setores acadêmico, industrial, serviços e àreas afins de telecomunicações para debater sobre o tema. A Programação Técnica contou com 253 trabalhos (sendo 214 Artigos Completos e 39 Artigos de Iniciação Científica), 5 Palestras Plenárias, 1 Painel (que discutiu o "Futuro do 5G no Brasil e Suas Implicações para a Sociedade Moderna"), 7 Minicursos e 39 Sessões Técnicas. O sucesso do SBrT'2020 representou uma radiografia clara da atividade da comunidade científica da área de telecomunicações e processamento de sinais no Brasil que, mesmo em tempos difíceis como este por que passamos, da pandemia do COVID-19, permaneceu produtiva e atuante.

Este livro é composto de 5 capítulos, em que os autores detalham e aprofundam o material apresentado em cada minicurso. O material apresentado é de grande relevância na área de telecomunicações, envolvendo plataformas de simulação de redes 5G e internet das coisas, protocolos de segurança em redes 5G, fundamentos de compatibilidade eletromagnética, e aspectos teóricos e práticos de redes de sensores sem fio. A organização do Simpósio agradece aos autores a preparação deste conteúdo que beneficiará estudantes, pesquisadores e profissionais da área.

Bartolomeu F. Uchôa Filho (UFSC)
Eduardo L. O. Batista (UFSC)

Sumário

CAPÍTULO

1 Entendendo o núcleo 5G na prática, através de uma implementação de código aberto

João Paulo Lobianco Silva (Instituto de Informática (INF) - Universidade Federal de Goiás (UFG)), Samuel Wanberg Lourenço Nery (INF/UFG), Rogério S. e Silva (INF/UFG), Antonio Oliveira-Jr (INF/UFG & Fraunhofer Portugal AICOS), Kleber Cardoso (INF/UFG) e Cristiano Bonato Both (Universidade do Vale do Rio dos Sinos - UNISINOS)

Introdução

A forma de prestar serviços está passando por uma grande transformação em toda a sociedade [1]. Por exemplo, a adoção de serviços como telemedicina, veículos autônomos e Internet das Coisas (*Internet of Things* - IoT) promoverão uma grande revolução em nossas vidas [2, 3, 4, 5, 6]. Esses serviços possuem diferentes e restritos requisitos que demandam novas arquiteturas de telecomunicação para garantir a qualidade dos serviços prestados. Entretanto, o projeto, desenvolvimento e implantação dessas arquiteturas precisam evoluir para atender de forma rápida e flexível esses requisitos rigorosos de comunicação. Atualmente, existe uma grande expectativa que as redes móveis de quinta geração (5G) atendam a esses requisitos [7].

O projeto para redes de quinta geração (5G) é especificado pela 3GPP[1] (*3rd Generation Partnership Project*) e apresenta mudanças significativas em toda a arquitetura de redes móveis celulares [8]. Por exemplo, o Lançamento 15 [9], finalizado em junho de 2019, apresentou uma nova arquitetura para redes celulares, com base em serviços e microsserviços (*Service-Based Architecture - SBA*). O Lançamento 16 [10], finalizado em julho de 2020, está focado na expansão do sistema 5G (*5G System* - 5GS), por exemplo, introduzindo a definição de serviço prioritário de multimídia, serviços da camada de aplicativos em *Vehicle-to-everything*

[1] 3GPP-A Global Initiative: `https://www.3gpp.org/`

(V2X), acesso via satélite 5G, acesso á rede local em 5G, convergência sem fio e cabeada para 5G, posicionamento e localização de terminais, comunicações em domínios verticais e automação de rede [8].

Sistemas 5G (*5G System* - 5GS) estão sendo desenvolvidos e aprimorados para (pretensiosamente) fornecer conectividade em qualquer lugar. O Lançamento 15 do Sistema 5G com as especificações 3GPP, compreendendo o 5G Core (5GC) e 5G *New Radio* (NR) com equipamentos de usuário (*User Equipment* - UE) 5G, está sendo implantado comercialmente em vários países, usando frequências abaixo de 6 GHz e ondas milimétricas. Embora o foco principal do Lançamento 15 tenha sido nos serviços aprimorados de banda larga móvel, o foco do Lançamento 16 está nos novos recursos para *Ultra-Reliable Low Latency Communication* (URLLC) e IoT Industrial, incluindo comunicação sensível ao tempo (*Time Sensitive Communication* - TSC), serviços de localização aprimorados e suporte para redes não públicas (*Non-Public Networks* - NPNs). Além disso, alguns novos recursos cruciais, como NR em bandas não licenciadas (NR-*Unlicensed*), *Integrated Access & Backhaul* (IAB) e NR V2X, também estão sendo introduzidos como parte do Lançamento 16, bem como aprimoramentos para *Multiple-Input Multiple-Out* (MIMO) massivo, convergência sem fio e cabeada, SBA e fatiamento de recursos.

SBA projetada para sistemas 5G apresenta, entre outras características, a separação estrutural do plano de controle e plano de dados (adotamos plano de usuário como sinônimo nesse documento), alta inserção de funções virtuais e um novo conceito de fatiamento de rede (*Network Slicing*). A presença de fatiamento de rede nessa arquitetura visa apoiar a criação de múltiplos serviços *End-to-End* (E2E) virtuais e isolados. Para entender a evolução do 5GS é importante situá-lo no contexto da evolução das redes móveis. 5GS baseia-se nos conceitos da computação em nuvem, cujo núcleo é orientado à serviços (SBA), com suporte nativo a fatiamento de recursos de rede, virtualização e computação móvel de borda [11]. Duas opções arquiteturais são propostas para o 5GS [7]:

- **Arquitetura Não-Autônoma (*Non-Standalone* - NSA)**: a rede de acesso e sua nova interface de rádio (NR) devem ser usadas em conjunto com a infraestrutura de núcleo existente (*Evolved Packet Core - Control and User Plane Separation – EPC-CUPS*).

- **Autônoma (*Standalone* - SA)**: NR está conectado ao novo núcleo 5G (5GC), com todos os serviços projetados para a quinta geração de redes móveis. Nessa arquitetura, a característica fundamental é que os elementos são definidos como funções de rede, as quais fornecem serviços a outras funções, ou qualquer outro “consumidor” autorizado, mediante interfaces.

Os aspectos das arquiteturas NSA [9] e SA [10], que permitem a implantação virtualizada das redes 5G, podem ser totalmente distribuídos, redundantes e escaláveis. Além disso, as várias instâncias de funções de rede podem estar presentes individualmente ou em conjunto com serviços e microsserviços, representando a plenitude dessa nova geração de redes móveis.

A rede de acesso 5G usando NR SA e o núcleo 5G estão padronizados pela 3GPP, ou seja, um UE conecta, de forma confiável, diretamente à rede de núcleo 5G. Entretanto, existem redes de acesso não-3GPP que a operadora móvel não confia no ponto de acesso ao qual o equipamento de usuário, ou seja, o dispositivo, está conectado. Pode ser uma rede local sem fio (*Wireless Local Area Network* - WLAN) (por exemplo, de uma casa) ou uma rede IoT (por exemplo, sensores e dispositivos IoT de uma indústria ou propriedade rural). O UE canaliza todo o tráfego para um *gateway* na rede, o qual é confiável para a operadora móvel. Esse é o acesso não-3GPP padronizado no Lançamento 15 para 5G. Para um melhor entendimento, a Figura 1.1 apresenta um cenário para sistemas 5G com uma rede de acesso 3GPP e seus UEs, conectados em uma *Radio Access Network* (RAN), e uma rede de acesso *não-3GPP* com seus UE-IoT. Essas redes estão conectadas ao 5G-Core (SBA), utilizando os planos de controle e de usuário, para interconectar a rede de dados (*Data Network* - DN) - principalmente a Internet.

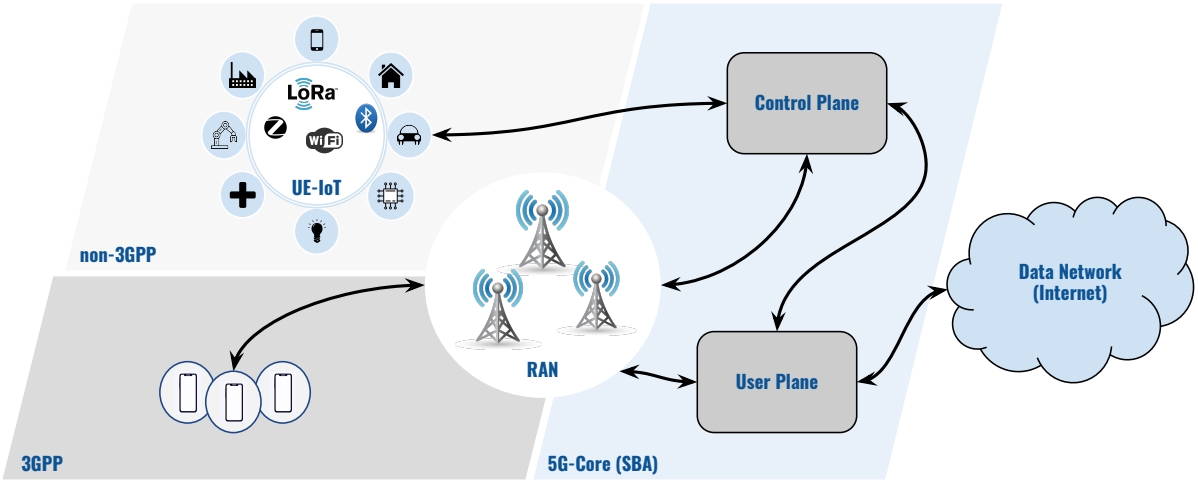


Figura 1.1 – Cenário de um sistema 5G.

A integração de redes de acesso não licenciadas (por exemplo, WLAN e DOCSIS (*Data Over Cable Service Interface Specification*)) com redes celulares licenciadas tem sido discutida a mais de uma década e está especificada nos Lançamentos 15 e 16. Entretanto, a integração de redes de acesso IoT (por exemplo, redes *Low Power Wide Area Networks* - LPWAN) não está especificada e deve usar a Função de Interconexão não-3GPP (*Non-3GPP Interworking function* - N3IWF) para acesso ao núcleo 5G. Com o crescimento massivo das redes IoT e uso em grande escala pelas indústrias e cidades inteligentes, cresce o interesse na integração de redes IoT não-3GPP à sistemas 5G. Dessa forma, neste minicurso, apresentamos um caso de uso dessa integração utilizando o my5G/my5GCore.

my5G[2] é uma iniciativa de software de código aberto que pretende tornar o sistema 5G acessível para todos. my5GCore[3] é um *fork* do projeto free5GC[4]. Esse projeto implementa o núcleo da rede 5G (5GC), seguindo as especificações técnicas do Lançamento 15 3GPP

[2] `https://github.com/my5G`
[3] `https://github.com/my5G/my5GCore`
[4] free5GC, National Chiao Tung University,https://www.free5gc.org/

e posteriores. Nesse contexto, o objetivo deste minicurso é apresentar, na prática, o 5GC (my5GCore) e um caso de uso IoT não-3GPP, através de uma implementação de código aberto da iniciativa my5G. Todo o processo de instalação, configuração, preparação do ambiente e o desenvolvimento de software no contexto do 5GC é apresentado, descrito e disponibilizado para os participantes do minicurso.

Este documento, para além desta **Introdução**, está organizado da seguinte forma: a Seção 1.1 (**Fundamentos de Softwarização**) apresenta os conceitos fundamentais para compreensão do tema; a Seção 1.2 (**O núcleo 5G**) apresenta a arquitetura SBA e as funções (adotamos componentes como sinônimo nesse documento) que formam o núcleo da rede; a Seção 1.3 (**5G em ação: Procedimentos de um sistema 5G**) apresenta as operações que compõem o sistema; a Seção 1.4 (**Desenvolvimento para o my5GCore: Caso de uso IoT não-3GPP**) apresenta o desenvolvimento do caso de uso 5GC (utilizando o my5GCore) e os resultados do estudo de caso contemplando o desenvolvimento de um UE IoT genérico com acesso não-3GPP ao núcleo da rede 5G; e a Seção 1.5 (**Material de apoio em repositório público**) descreve brevemente o conteúdo suplementar deste minicurso (códigos, vídeos) disponível em um repositório público e o seu endereço para acesso.

1.1 Fundamentos de softwarização

Sistemas 5G estão sendo desenvolvidos com o suporte das mais recentes tecnologias disruptivas, tais como, Redes definidas por software (*Software Defined Networks* - SDN), Virtualização das Funções de Redes (*Network Function Virtualization* - NFV), Computação em nuvem (*Cloud Computing*), Arquitetura baseada em serviços (*Service-Based Architecture* - SBA) e Conteinerização. Desta forma, é possível observar que a softwarização (i.e., a adoção de funcionalidades desenvolvidas em software, em vez de, integradas ao hardware) orienta o desenvolvimento das redes de comunicação, e é o caminho natural para a sua evolução. Alguns importantes conceitos habilitadores para desenvolvimento de sistemas 5G são apresentados em seguida.

Redes definidas por software (SDN) – é uma abordagem de rede em que o plano de controle é desacoplado do hardware específico de comunicação e fornecido um aplicativo de software [12]. SDN é um paradigma de redes programáveis que foi proposto como uma abordagem para facilitar a evolução das redes. O objetivo das SDNs é fornecer uma interface de programação aberta, permitindo o desenvolvimento de aplicativos que possam controlar e gerenciar dinamicamente a conectividade entre os elementos da rede [13].

Virtualização das funções de rede (NFV) – implementa funções de rede por meio de técnicas de virtualização de software e as executa em hardware comum (*commercial off-the-shelf* - COTS). Os dispositivos virtuais podem ser instanciados sob demanda sem a instalação de novos equipamentos [14]. A virtualização das funções de redes impacta em uma importante mudança no fornecimento de serviços de telecomunicações. Implementar as funções de rede em software é uma abordagem viável para tornar os equipamentos de rede

mais abertos e, portanto, permitir que os provedores de serviços de telecomunicações sejam mais flexíveis, mais ágeis na implantação de novos serviços, o que pode levar a reduções significativas de custos [15, 16].

Cloud Native – é um modelo para permitir acesso onipresente, conveniente e de rede sob demanda a um conjunto compartilhado de recursos de computação configuráveis (por exemplo: redes, servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e liberados com o mínimo esforço de gerenciamento ou interação do provedor de serviços [17]. O conceito *Cloud Native* trata da abordagem de como os aplicativos podem ser criados e operados na computação em nuvem. Esse conceito pode ser aplicado para as tecnologias, a arquitetura e as ferramentas e envolve tecnologias relacionadas a contêineres, microsserviços e Interfaces de Programação de Aplicações (*Application Programming Interfaces* - API), entre outras. A Cloud Native Computing Foundation (CNCF)[5] define a estrutura e as restrições para que aplicações sejam consideradas *Cloud Native*. Entre as restrições estão a utilização de metodologias de microsserviços e a implantação através de contêineres. Outro pilar crítico é a orquestração dinâmica de contêineres (por exemplo, Kubernetes[6], Docker Swarm[7]).

Microsserviços – são pequenos serviços autônomos que trabalham colaborativamente [18]. Os microsserviços são componentes de software que podem ser executados como processos e capazes de manter comunicação com outros microsserviços por meio de APIs bem definidas. A abordagem de microsserviços é considerada uma variante da Arquitetura Orientada a Serviços (*Service-Oriented Architecture* - SOA). O objetivo geral do desenvolvimento com microsserviços é tornar os aplicativos mais fáceis de escalar e mais rápidos para desenvolver, permitindo a inovação e acelerando o tempo de entrega de novos recursos. No entanto, alguma complexidade está associada, o que inclui gerenciamento, orquestração e criação de novos métodos de gerenciamento de dados [7]. Por exemplo, no contexto dos sistemas 5G, o conceito de microsserviços pode ser utilizado para a implantação e expansão do sistema, em linha com a rápida evolução da tecnologia da informação. Um dos objetivos da utilização de microsserviços nos sistemas 5G é a possibilidade de decompor os componentes em funções virtualizadas, com baixa granularidade, para tornar o serviço leve e com alta capacidade de compartilhamento. Esse objetivo se encaixa perfeitamente na necessidade de definir cenários de comunicação para banda larga móvel aprimorada (*enhanced Mobile Broadband* - eMBB), suporte a densidades extremamente altas de dispositivos conectados (*massive Machine Type Communication* - mMTC) e comunicações URLLC, uma vez que oferece modularidade, reutilização de funções e interoperabilidade com redes heterogêneas. Além disso, a introdução de microsserviços e interfaces para acessá-los simplifica os processos de atualização e manutenção dos sistemas 5G, reduzindo o custo de operação e acelerando a introdução de novos serviços [19].

[5] CNCF: `https://github.com/cncf/toc/blob/master/DEFINITION.md`

[6] Kubernetes: `https://kubernetes.io/`

[7] Docker Swarm: `https://docs.docker.com/get-started/swarm-deploy/`

Conteinerização – A virtualização revolucionou a infraestrutura de TI e permitiu aos fornecedores de tecnologia oferecer diversos serviços baseados em TI aos consumidores. Basicamente, a virtualização no nível do sistema permite que as instâncias de diferentes Sistemas Operacionais (SOs) sejam executadas simultaneamente em um único servidor através de um hipervisor, que cria e executa máquinas virtuais. Os contêineres, por outro lado, são isolados uns dos outros dentro do SO, mas compartilham o núcleo do SO. Os contêineres são amplamente usados em cenários onde é necessário otimizar recursos de hardware para executar vários aplicativos e melhorar a flexibilidade e a produtividade. Além disso, os ecossistemas e ferramentas para ambiente baseado em contêineres (por exemplo, Kubernetes) estão se expandindo rapidamente [7].

Muitas dessas tecnologias já estão sendo largamente utilizadas na área de Tecnologias de Informação. Além disso, a adoção dessas tecnologias de forma integrada na área de telecomunicação também está se tornando uma realidade. Por exemplo, a próxima seção apresentamos o novo núcleo 5G que leva em consideração a utilização dessas tecnologias, i.e., SBA.

1.2 O núcleo 5G

No contexto das redes de telecomunicações móveis, o núcleo 5G (5G Core - 5GC) é responsável pela execução das rotinas necessárias ao processo de autenticação, autorização, provisionamento de recursos de rede e conectividade. Especificamente em sistemas 5G, o núcleo deve ser capaz de fornecer interoperabilidade entre tecnologias de acesso fixas e móveis, suportando interação entre funções do plano de controle (reusabilidade, conexões flexíveis e descoberta de serviços) e o plano de usuário [20].

De acordo com Hedman et al. [7], 5GC está sendo projetado para usufruir dos benefícios da SBA, por exemplo, (i) desenvolvimento e manutenção mais eficiente, (ii) microsserviços associados a recursos individuais, (iii) instâncias de microsserviços sob demanda para melhor escalabilidade e (iv) ciclos de vida independentes para os microsserviços na atualização do software. Isso possibilita um desenvolvimento mais flexível de novos serviços, pois torna-se possível conectar outros componentes sem introduzir novas interfaces específicas. Nesse contexto, o núcleo é composto de várias funções de rede (*Network Functions* - NFs) que também são chamadas de componentes 5GC, e executa suas tarefas específicas através das interações entre as NFs com a rede de acesso, UEs e com o plano do usuário (*User Plane Function* - UPF). Além disso, 5GC fornece uma estrutura modular na qual as funcionalidades podem ser implantadas por meio de interfaces que possibilitam o fornecimento e consumo de serviços [21]. Uma NF pode operar de duas maneiras: *(i)* uma NF consumidora pode solicitar uma resposta de uma NF produtora, por exemplo, para solicitar informações sobre políticas do usuário; *(ii)* ou pode se inscrever em uma produtora e ser notificada se necessário, por exemplo, se o estado do UE mudar para inativo [22].

Os serviços são acessados através de interfaces baseadas em serviços (SBI) usando APIs e

uma combinação dos protocolos HTTP (*Hypertext Transfer Protocol*), REST (*Representational State Transfer*) e JSON (*JavaScript Object Notation*). Construir uma rede de acesso 5G desagregada, virtualizada e definida por software é a direção que a indústria já está tomando (por boas razões técnicas e comerciais) [23, 20], e dividir a rede 5G em seus componentes elementares também é a melhor maneira de explicar como o 5G funciona.

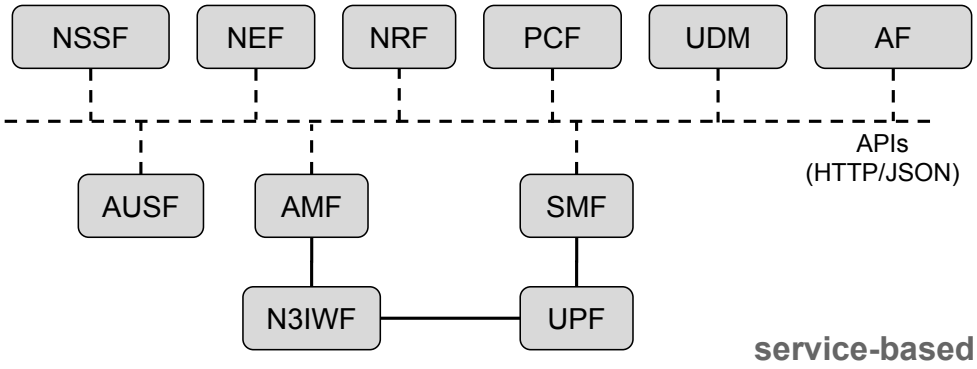


Figura 1.2 – Núcleo 5G.

A Figura 1.2 apresenta o 5GC e as interações entre as NFs no núcleo, das quais se destacam: NSSF - *Network Slice Selection Function*; NEF - *Network Exposure Function*; NRF - *NF Repository function*; PCF - *Policy Control Function*; UDM - *Unified Data Management*; AF - *Application Function*; AUSF - *Authentication Server Function*; AMF - *Access and Mobility Management function*; N3IWF - *Non-3GPP Interworking Function*; UPF - *User plane function*; SMF - *Session Management function*. As principais NFs são descritas a seguir.

1.2.1 Funções de Rede do 5GC

A **Função de Gerenciamento de Acesso e Mobilidade (AMF)** atua no estabelecimento de conexão entre o UE e 5GC. Essa ação aciona um conjunto de procedimentos para identificar o UE, fornecendo uma estrutura de segurança para oferecer um canal de transporte de mensagens. O principal objetivo do componente AMF é garantir que o processo de comunicação ocorra de forma coesa e transparente, considerando a mobilidade do usuário como um fator crítico. A partir das funções implementadas na AMF, a rede pode chegar a um determinado usuário para avisar sobre eventuais mensagens ou ligações recebidas, por exemplo. Além disso, o componente AMF pode permitir que um determinado UE inicie um processo de comunicação com outros UEs conectados à RAN ou com acesso à Internet. Outra funcionalidade fundamental da AMF é garantir a conectividade que mantém as sessões ativas quando os UEs se movem entre diferentes pontos de acesso.

A **Função de Gerenciamento de Sessão (SMF)** tem a responsabilidade de configurar a conectividade do UE para a DN, bem como gerenciar o Plano de Usuário para essa conectividade. A Figura 1.3 apresenta as principais interações da SMF com outras NFs no 5GC, com destaque para as associações com AMF, UPF, UDM e PCF, descritas a seguir.

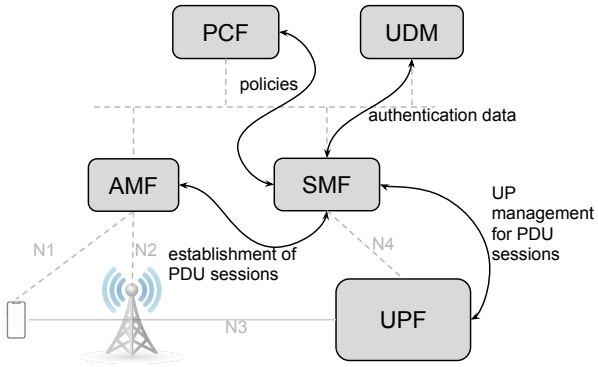


Figura 1.3 – Principais interações da SMF.

- Gerencia as sessões do usuário, que representam os dispositivos dos usuários conectados. A SMF se comunica indiretamente com o UE por meio do AMF, que retransmite mensagens relacionadas à sessão entre os dispositivos e SMF.
- Controla a UPF por meio da interface de rede N4. Esse controle inclui a capacidade da SMF de configurar a direção do tráfego de dados associado a uma UPF para uma determinada sessão de UE e executar ações de monitoramento e controle.
- Interage com a função UDM para a geração de dados de autenticação de dispositivos e autorização de acesso de usuários de acordo com dados de seu plano. Caso haja mais de uma instância SMF na rede, a UDM mantém registro de qual instância está atendendo a um dispositivo específico.
- Interage com as funções relacionadas à PCF, visando executar a política de controle de sessão dos UEs conectados. Essa ação de execução pode ser descrita como uma das principais tarefas associadas a um sistema 5G. Por exemplo, essa ação determina as diretrizes para a conectividade de dados entre um UE e a DN.

A **Função servidor de autenticação (AUSF)** fornece três serviços, (i) autenticação de UEs, via AMF, por meio das credenciais de acessos fornecidas pela UDM, (ii) criptografia para o tráfego seguro de informações nos procedimentos de atualização do UE, e (iii) fornecimento de parâmetros de segurança para proteção na execução dos processos de atualização de *roaming*. A Figura 1.4 apresenta as interações entre a AUSF e a AMF e UDM. Na interação com a AMF, a AUSF realiza as autenticações dos UEs (3GPP ou não-3GPP), por meio das credenciais criadas pela UDM.

A **Função de Repositório de Rede (NRF)** reúne informações de perfil das NFs que estão disponíveis na rede. A NRF provê às NFs consumidoras de serviços a descoberta e seleção das funções produtoras disponíveis [7]. Quando uma instância de uma NF é implantada ou alterada, por exemplo, devido a necessidade de escala, a NRF é atualizada com novas informações de perfil da NF. Os perfis podem ser atualizados pela própria NF ou por outra, em seu nome [24]. A NRF é uma função centralizadora responsável por guardar as informações sobre quais os serviços são oferecidos e como podem ser consumidos.

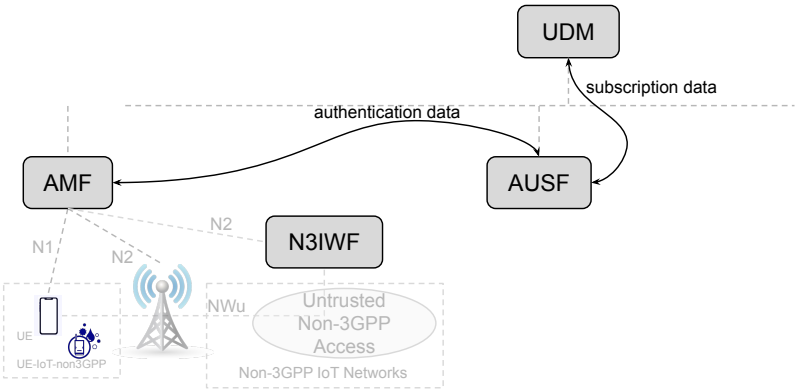


Figura 1.4 – Interações da AUSF.

A **Função de Exposição de Rede (NEF)** oferece suporte à exposição de eventos e recursos do sistema 5G para aplicativos e funções de rede dentro e fora da rede da operadora [7]. A NEF oferece suporte ao monitoramento de eventos e os disponibiliza para os aplicativos autorizados e outras NFs, por exemplo, localização de UE, acessibilidade, estado do *roaming* e perda de conectividade. A NEF interage com a AMF para sintonia do sistema 5G e para gerenciar o comportamento do UE. Além disso, a NEF pode suportar diferentes conjuntos de funcionalidades e interoperar com outras NEFs no 5GC [25].

A **Função de gerenciamento de dados unificados (UDM)** é um *front-end* para os dados de assinatura do usuário armazenados na UDR e gerencia os dados na rede de forma centralizada. Por meio da UDM, várias outras NFs executam diversas ações, por exemplo, (i) registro e autenticação de UEs, (ii) manuseio da identificação do usuário, (iii) acesso para usuários específicos com base nos dados de assinatura e a aplicação de regras de acesso, (iv) autorizações diferentes para assinantes de *roaming* e assinantes domésticos, e (v) fornecimento de informações à NEF para recuperação de informações/restrições de acesso não-3GPP do assinante. Caso haja mais de uma instância de AMF e SMF na rede, a UDM mantém registro de qual instância está atendendo a um dispositivo específico. A UDM também dá suporte a recuperação dos dados de assinatura individual do UE para seleção de *slice* (dados de autenticação, de sessão, de seleção de SMF, e de contexto).

Dentro do 5GC, a UDM fornece e consome serviços das funções AMF, AUSF, SMF, NEF, e UDR (entre outras NFs) através da SBI. A Figura 1.5 apresenta alguns exemplos dessas interações, por exemplo, (i) com a AMF, a UDM troca informações acerca do acesso e da mobilidade dos usuários, (ii) com a NEF, a UDM interage para obter informações de acesso de UEs não-3GPP ao 5GC, (iii) com a AUSF para autenticação de dados, (iv) com a SMF para a recuperação de dados de assinante e autenticação de dados, e (v) com a UDR, são realizadas as operações de armazenamento e recuperação de dados.

A **Função de Plano de Usuário (UPF)** tem como principal tarefa processar e encaminhar os dados de usuário. A funcionalidade da UPF é controlada a partir da SMF, i.e., indicando qual UPF deve se conectar com a rede externa e atuar como um ponto de conexão IP para os dispositivos dessa rede externa. Dessa forma, a mobilidade dos

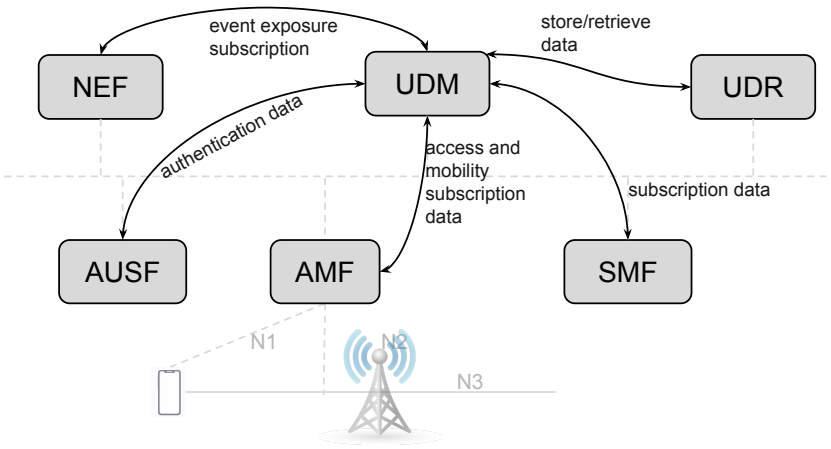


Figura 1.5 – Interações da UDM.

dispositivos é ocultada para a rede externa. Isso significa que os pacotes IP com um endereço de destino, pertencente a um dispositivo específico, são sempre roteáveis da Internet para a UPF específica que está servindo a esse dispositivo, mesmo quando o dispositivo está se movendo na rede. Além disso, a UPF pode aplicar marcações de *Quality of Service* (QoS) em pacotes para a rede de rádio ou para redes externas, possibilitando que a rede de transporte possa lidar com cada pacote com a prioridade correta em caso de congestionamento na rede.

A **Função de Políticas de Controle (PCF)** provê as políticas de controle para as funcionalidades relacionadas com o gerenciamento de sessões para o acesso e mobilidade, para a seleção de acesso dos UEs e a seleção de sessões *Protocol Data Unit* (PDU). Além disso, PCF provê suporte para as transferências em *background*.

A **Função de Interconexão não-3GPP (N3IWF)** é responsável pela interação entre redes não-3GPP (por exemplo, IoT) não confiáveis e o 5GC. A N3IWF suporta conectividade baseada nas interfaces N2 e N3 para o núcleo, enquanto suporta conectividade segura IPSec (*IP Security*) com o UE. A Figura 1.6 apresenta as interações da N3IWF no 5GC. No procedimento de registro, o UE se conecta por túneis IPSec ao N3IWF via interface *NWu*; após o registro o UE suporta sinalização NAS com o 5GC pela interface N1 [26]. A N3IWF interage com o plano de controle (AMF) por meio da interface N2 e com a UPF por meio da interface N3 [27].

A interação entre as várias funções de rede da SBA 5G é discutida em detalhes a seguir respectivamente nas subseções 1.2.2 (Registro, Descoberta e Requisição de Serviços), 1.2.3 (Gerenciamento de Conexão, Registro e Mobilidade) e 1.2.4 (Gerenciamento de Sessão (5GSM)).

1.2.2 Registro, Descoberta e Requisição de Serviços

SBA define a arquitetura através da qual as NFs no núcleo podem produzir e consumir serviços, sem a necessidade de uma interface ponto-a-ponto entre elas. Ao inicializar, cada NF deve informar à NRF quais serviços ela oferece, juntamente com as versões suportadas,

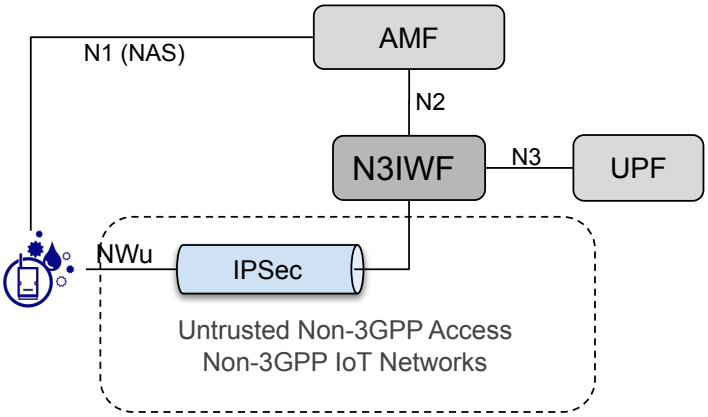


Figura 1.6 – Interações da N3IWF.

além do IP a partir do qual as outras NFs podem consumir cada um desses serviços. Esse procedimento é chamado de Registro de Serviço (*Service Registration*). Quando alguma NF precisa de algum serviço fornecido por outra NF, ela deve consultar a NRF sobre quais as NFs disponíveis que oferecem esse serviço. A NRF retorna uma lista com todas as insâncias de NFs disponíveis. Ao receber a lista, cabe à NF que fez a consulta, decidir a qual instância solicitará o serviço. Esse procedimento de consulta é chamado de Descoberta de Serviço (*Service Discovery*). As NFs que oferecem serviços são chamadas de produtoras, enquanto aquelas que consomem serviços são chamadas de consumidoras. Cada NF age como consumidora em alguns momentos e como produtora em outros. Além disso, é possível uma consumidora se inscrever (*subscribe*) junto a uma produtora, para ser notificada quando determinado evento acontecer. A comunicação de sinalização no 5GC usa o paradigma HTTP e REST, o que facilita a interoperabilidade com outros sistemas.

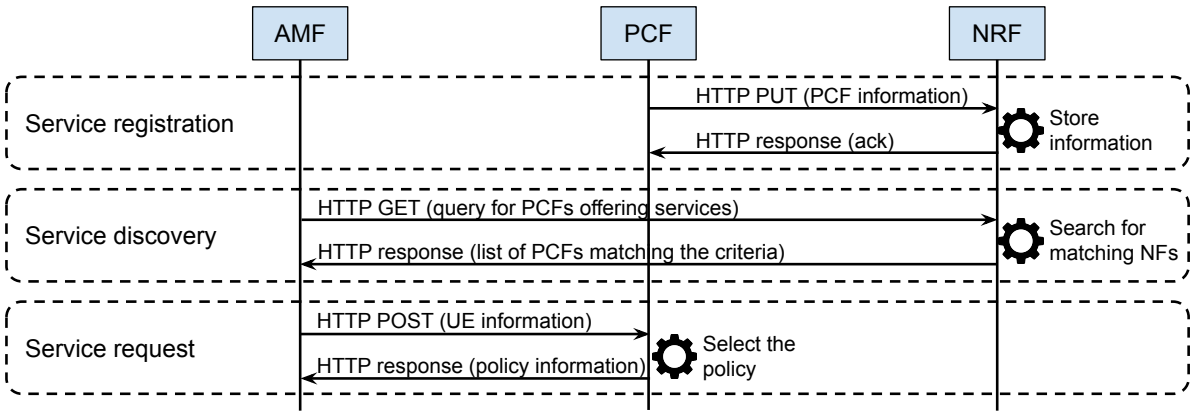


Figura 1.7 – Fluxo de registro, descoberta e requisição de serviços na SBA 5G.

A Figura 1.7 apresenta um modelo simplificado para exemplificar o procedimento de registro, descoberta e requisição de serviços no âmbito da recuperação das políticas de acesso de um usuário ao sistema 5G. Conforme ilustrado na figura, inicialmente uma NF, neste caso a PCF, registra os serviços que fornece na NRF – ***Service registration***, em seguida, a AMF consulta a NRF em busca de uma PCF adequada e, em resposta, recebe uma lista de PCFs

(segundo critérios) – ***Service discovery***. Finalmente, a AMF requisita as políticas de acesso (para um dado UE) fornecidas pela PCF e as recebe como resposta – ***Service request***.

1.2.3 Gerenciamento de Conexão, Registro e Mobilidade

As duas principais funções das redes móveis são prover conectividade de dados e mobilidade para seus usuários. No 5G, os conceitos relacionados à conectividade de dados são tratados dentro do gerenciamento de sessão (*5G Session Management* - 5GSM), enquanto os conceitos de mobilidade são tratadas dentro do gerenciamento de mobilidade (*5G Mobility Management* - 5GMM), que inclui o gerenciamento de conexão (*5G Connection Management* - 5GCM) e gerenciamento de registro (*5G Registration Management* - 5GRM). Para facilitar o entendimento, os conceitos relacionados a 5GCM, 5GRM e 5GMM são abordados nessa subseção e os conceitos relacionados a 5GSM são tratados na subseção 1.2.4 (Gerenciamento de Sessão (5GSM)).

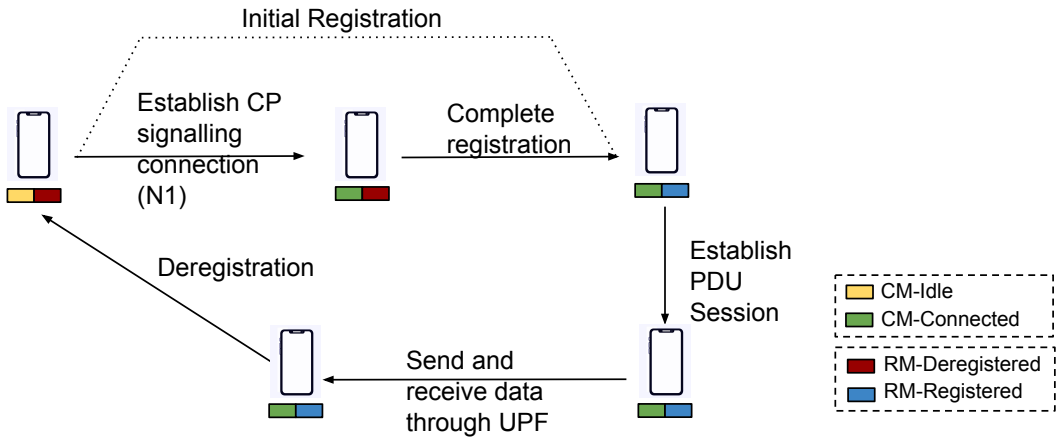


Figura 1.8 – Fluxo básico executado por um UE com as mudanças de estado CM e RM.

O fluxo básico para que o usuário consiga acessar serviços em uma rede de dados (por exemplo, a Internet), é iniciando com a criação da conexão de sinalização com o plano de controle (N1), registro e estabelecimento de uma sessão PDU. é importante destacar, que a todo momento é preciso manter a continuidade de serviços e sessão, mesmo quando o UE sai da área de registro. Cada uma dessas etapas está relacionada com um ou mais desses quatro gerenciamentos. A Figura 1.8 mostra como os estados de gerenciamento de conexão (*Connection Management* - CM) e gerenciamento de registro (*Registration Management* - RM), representados respectivamente pelos retângulos esquerdo e direito abaixo do UE, são alterados conforme cada etapa é executada. O UE inicia nos estados *CM-IDLE* e *RM-DEREGISTERED*, mudando para os estados *CM-CONNECTED* e *RM-REGISTERED* ao final do procedimento de registro inicial, retornando para os estados *CM-IDLE* e *RM-DEREGISTERED* ao final do procedimento de cancelamento de registro.

O **gerenciamento de conexão (5GCM)** é responsável pelo estabelecimento e término da conexão NAS de sinalização de plano de controle entre UE e AMF, através da interface de

referência N1. N1 é uma conexão lógica que compreende as conexões entre UE e a rede de acesso (3GPP ou não-3GPP) e entre a rede de acesso e AMF (N2), como pode ser observado na Figura 1.9. Após o estabelecimento dessa conexão de sinalização, o UE pode interagir com o núcleo para realizar os procedimentos de controle, como registro, autenticação, recebimento de políticas e estabelecimento de sessão PDU.

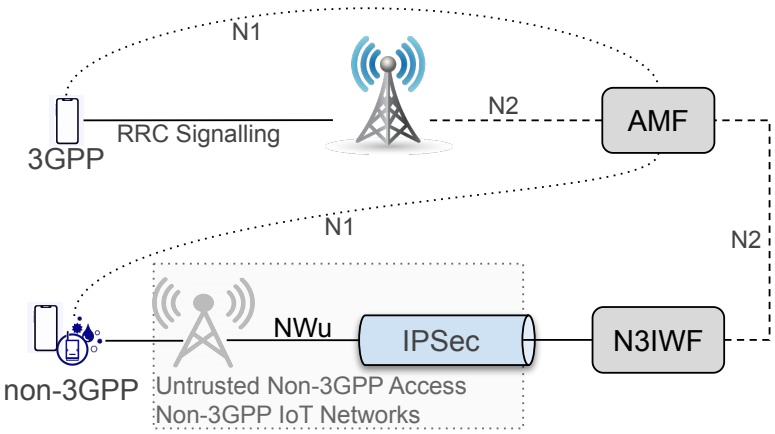


Figura 1.9 – Conexão de sinalização do plano de controle para acessos 3GPP e não-3GPP.

Os procedimentos de Registro e Requisição de Serviço podem alterar o estado da conexão de *CM-IDLE* para *CM-CONNECTED*, enquanto o procedimento de Cancelamento de registro pode alterar o estado de *CM-CONNECTED* para *CM-IDLE*. Além disso, para acessos 3GPP, o UE também pode decidir liberar a conexão e entrar no estado *CM-IDLE*, para tentar economizar energia, restabelecendo a conexão e entrando em (*CM-CONNECTED*), quando precisar enviar dados ou em resposta a um *Paging*, isto é, quando a rede notifica o UE que possui dados para entregá-lo.

A conexão pode estar em dois estados, *CM-CONNECTED* ou *CM-IDLE* e são mantidos tanto no UE quanto na AMF. A Figura 1.10 mostra as transições de estado CM no UE e AMF. Conforme ilustrado na Figura 1.10(a), a AMF muda de estado para *CM-CONNECTED* quando o contexto N2 é estabelecido com a rede de acesso, caso contrário, altera para o estado *CM-IDLE*. Além disso, conforme ilustrado na Figura 1.10(b), o UE muda de estado para *CM-CONNECTED* quando a conexão de sinalização com a rede de acesso é estabelecida, e quando é liberada altera para o estado *CM-IDLE*.

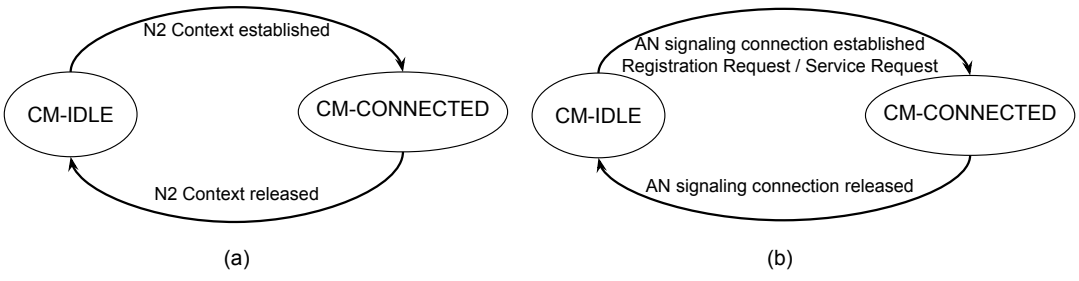


Figura 1.10 – Transições de estado de gerenciamento de conexão no (a) AMF e (b) UE [27].

O **gerenciamento de registro (5GRM)** é usado para registrar e cancelar o registro do UE junto à rede, além de estabelecer o contexto de usuário. O usuário precisa se registrar junto à rede para receber serviços, por exemplo, acesso à Internet. Existem dois estados de registro: *RM-REGISTERED* e *RM-DEREGISTERED*. Basicamente, quando o UE é iniciado, ele tenta se registrar junto ao núcleo, através do procedimento de registro inicial, o qual inclui etapas para criar conexão de sinalização (N1), identificação, autenticação e autorização, criação do contexto de usuário e aplicação de políticas. Caso o registro seja bem sucedido, o UE transita do estado *RM-DEREGISTERED* para *RM-REGISTERED*, além do contexto do usuário ser criado junto à AMF também com o estado *RM-REGISTERED*. Semelhante ao estado de CM, o estado de RM é mantido tanto no UE quanto na AMF. Uma vez que o UE transita para *CM-REGISTERED*, a 3GPP define várias interações entre UE e AMF para gerenciar os estados de CM e RM. Nos acessos 3GPP, o UE deve atualizar seu registro periodicamente ou quando ocorre mobilidade. Nos acessos não-3GPP, somente é necessário refazer o registro quando houver necessidade de atualizar parâmetros de protocolos que são negociados durante o procedimento de registro inicial, uma vez que a 3GPP não prevê mobilidade para esse tipo de acesso. Além disso, UE e AMF possuem temporizadores que são iniciados quando a conexão entra em *CM-IDLE*, sendo suspenso quando retornam para *CM-CONNECTED* ou executando o procedimento de cancelamento de registro implícito, quando expiram.

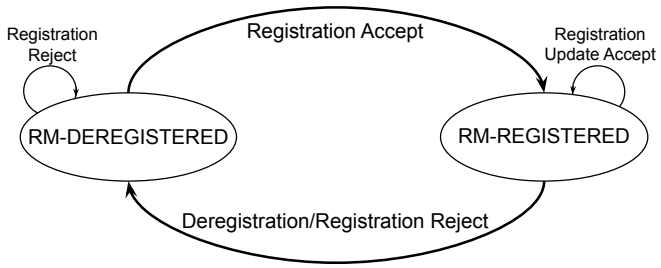


Figura 1.11 – Transições de estado de registro [27].

A Figura 1.11 mostra as transições de estado de RM no UE e AMF. Quando uma requisição de registro ou atualização de registro é bem sucedida, com recebimento da mensagem *Registration Accept* ou *Registration Update Accept*, o UE e AMF mudam de estado para *RM-REGISTERED*. Quando a requisição é rejeitada ou é realizado o procedimento de cancelamento de registro, UE e AMF altera de estado para *RM-DEREGISTERED*.

O **gerenciamento de mobilidade (5GMM)** é responsável por garantir que a rede consiga alcançar o usuário para notificar sobre chamadas ou mensagens recebidas, que o usuário consiga iniciar comunicação com outros usuários ou acessar serviços (por exemplo, a Internet) e pela continuidade dos serviços e sessões mesmo quando o usuário está se movimentando dentro ou entre tecnologias de acesso [7]. Para isso, o sistema 5G oferece um conjunto de procedimentos para gerenciamento de mobilidade que se dividem em 3 categorias:

- **Procedimentos comuns**: podem ser iniciados sempre que o UE está no estado *CM-CONNECTED*. Inclui os procedimentos para autenticação mútua entre UE e 5GC, identificação, transporte entre UE e AMF, atualização de configuração de UE e reporte de erros.

- **Procedimentos específicos**: só um procedimento pode estar sendo executado por um UE, por tipo de acesso (3GPP ou não-3GPP). Inclui os procedimentos para registro e cancelamento de registro.

- **Procedimentos para gerenciamento de conexão**: utilizado para estabelecimento da conexão de sinalização entre UE e núcleo ou para requerer a ativação de recursos do plano de usuário para envio de dados. Inclui os procedimentos de Requisição de serviço, *Paging* e Notificação.

A rede de acesso é composta por células que representam as antenas de rádio através das quais um UE 3GPP consegue acessar a rede 5G. Essas células estão agrupadas em áreas de cobertura (*Tracking Area* - TA). As células dentro de uma mesma TA fazem *broadcast* (difusão) do identificador dessa TA e é através desse identificador que um UE consegue saber em qual TA ele está atualmente. Durante o procedimento de registro, o UE recebe uma área de registro (*Registration Area* - RA), contendo a lista das TA suportadas por esse registro. Quando o UE se movimenta entre duas células dentro da mesma área de registro, é realizado o procedimento de *Handover* para que a RAN de destino passe a servir o UE. Caso o UE se mova para fora da sua área de registro, ele executa um procedimento de atualização de registro de mobilidade (*Mobility Registration Update*), selecionando uma nova célula, que pode escolher uma nova AMF para servir esse UE, juntamente com uma nova área de registro.

1.2.4 Gerenciamento de Sessão (5GSM)

Nos sistemas 5G, o gerenciamento de sessão é responsável por prover conectividade de dados entre o UE e a rede de dados da operadora, assim como gerenciar os componentes chaves do núcleo da rede para estabelecimento e manutenção do plano de dados [7]. Os recursos de gerenciamento de sessão em rede 5G foram projetados para obter maior flexilidade de operação, sendo capazes de suportar diferentes casos de uso. Essa característica corresponde ao uso de sessões PDU para o transporte de dados como um meio estruturado e flexível para operar com diferentes tipos de protocolos de comunicação na rede de acesso.

Uma sessão PDU representa uma conexão lógica entre o UE e a rede de dados (DN). PDU é o protocolo referência para transporte de dados no plano do usuário em redes 5G. No 5GC, as funcionalidades de gerenciamento de sessão estão diretamente associadas à camada de sessão PDU. Além disso, o protocolo é responsável por, gerenciar a distribuição de endereços IP, aplicar regras de QoS, gerenciar a mobilidade, garantir segurança e executar rotinas de cobrança e políticas de controle de acesso.

Para conectar com a rede de dados, o UE deve primeiro submeter uma requisição de estabelecimento de sessão PDU para o 5GC. No conteúdo da requisição, é informado o nome da rede de dados (*Data Name Network* - DNN) à qual se conectará. Cada sessão PDU representa uma única conexão entre o UE e a DN (por exemplo, Internet, *IP Multimedia System* (IMS) ou *Mobile Edge Computing* (MEC), entre outras) e também existem os casos de uso específicos de conexão com rede de dados para aplicações industriais ou comerciais privadas.

Ao se associar à rede 5G, o UE através da interface RAN estabelece uma nova conexão com a UPF, e os dados provenientes das aplicações do UE são encapsulados em pacotes PDUs. O tráfego da aplicação é encaminhado para a DN através da UPF no núcleo da rede, por uma rede subjacente, específica para transporte de pacotes PDUs. O tráfego da aplicação do usuário é relacionado com o tipo de serviço que é consumido, exemplos de serviços habituais são: web (HTTP/HTTPS), e-mail (SMTP), e transferência de arquivos (FTP).

Para exemplificar o processo de estabelecimento da conexão PDU em redes 5G, a Figura 1.12 apresenta o plano de usuário, com o esquema da pilha de protocolos da rede de transporte subjacente relacionada à camada de transporte para sessões PDU. Essa camada deve ser separada da rede de transporte responsável por conectar os nós existentes na topologia da rede ao sistema 5G. O plano de usuário representa a conexão da sessão PDU com a DN, sendo construído sob tunelamento dos dados pela rede de transporte com o intuito de garantir criptografia fim-a-fim entre o UE e a DN. Além disso, tem como objetivo desacoplar a camada de sessão PDU da rede de transporte subjacente, possibilitando ao operador da rede utilizar qualquer tecnologia de comunicação na construção da rede de acesso, de forma transparente para a camada da sessão PDU.

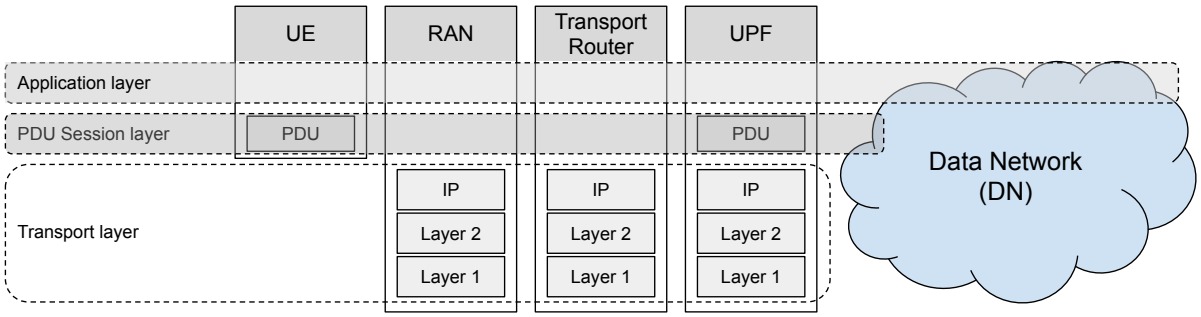


Figura 1.12 – Pilha de protocolos para construção do plano de usuário com estabelecimento da sessão PDU.

Na implantação da rede de transporte subjacente, é utilizado o protocolo IP com diferentes tecnologias de acesso ao meio físico, por exemplo, o protocolo MPLS (*Multiprotocol Label Switching)*), redes convencionais *Ethernet*, enlaces de conexão ponto-a-ponto, e enlaces sem fio. As sessões de PDU são transparente para a infraestrutura relacionada à camada de transporte na rede *backbone* da operadora, como no caso de roteadores IP intermediários ou comutadores (*switches*) da camada de enlace (Figura 1.12).

O UE pode estabelecer múltiplas sessões PDUs para diferentes casos de uso, em que

o acesso simultâneo em mais de uma rede de dados se faz necessário. Por exemplo, em cenários que o UE deseja obter conectividade com a Internet e ao mesmo tempo com uma rede IMS ou uma interface de computação móvel em borda MEC,ou estabelecimento de múltiplas sessões PDU, para um mesma DN simultaneamente por múltiplos caminhos de dados UPF para fins de redundância, persistência e escalabilidade. A Figura 1.13 ilustra diferentes esses cenários.

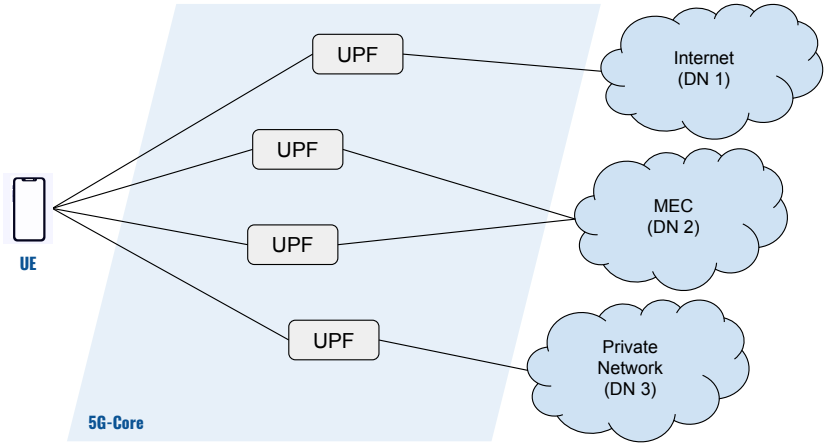


Figura 1.13 – Estabelecimento de múltiplas sessões PDU.

A sessão PDU é o ponto central para o estabelecer conectividade entre o UE e a rede de dados DN. Para isso, alguns parâmetros são fundamentais para descrever as propriedades PDU, sendo que essas propriedades são definidos durante o processo de estabelecimento da sessão e não podendo ser alterados durante o tempo de vida da sessão PDU [7]. Algumas das principais propriedades para sessão PDU são:

- ***PDU Session property***: Trata-se do identificador único da sessão PDU, aplicado tanto para identificação de sessão no UE, quanto para a rede de dados.
- ***Slice identifier (S-NSSAI)***: Refere-se à fatia de rede (*network slice*) em que a sessão PDU irá estabelecer conexão para prosseguir com a conectividade de dados.
- ***Data Network Name* (DNN)**: é o nome da rede de dados específica, na qual a sessão PDU estabelecerá a conectividade de dados.
- ***PDU Session Type***: Trata-se do protocolo base, em nível de usuário, para comunicação fim a fim utilizado na rede de transporte subjacente, onde são transportados os pacotes da sessão de PDU. Os protocolos mais habituais são o IPv4 e IPv6, *Ethernet* ou pacotes não estruturados projetados para suportar aplicações IoT (por exemplo, 6LoWPAN, MQTT e CoAP). Por exemplo, para requisição de sessões PDU que utilizam o protocolo IP, o UE deve primeiro configurar a requisição de sessão PDU para o tipo especifico do protocolo baseado nas capacidades da pilha IP. Além disso, o UE que possui suporte para IPv6 ou IPv4 deve definir em seu escopo o tipo de sessão PDU de acordo com as

capacidades possíveis do UE ou pela política recebida do operador do sistema (por exemplo, IPv4, IPv6 ou IPv4v6 quando ambos são usados). A função de rede SMF é encarregada por definir qual endereço será atribuído ao UE. Quando a SMF recebe uma requisição de estabelecimento de sessão PDU, a SMF seleciona o tipo da sessão PDU endereçada ao UE baseada no tipo de protocolo suportado pela rede de dados.

- ***Service and Session Continuity (SSC) mode***: Refere-se à longevidade do ponto de ancoragem do plano do usuário da sessão de PDU. A SSC é uma ferramenta para condução da escolha do plano de dados com a seleção e ancoragem da UPF em melhor disposição na topologia de rede durante a fase de estabelecimento da sessão PDU. A arquitetura 5GC inclui suporte mais flexível para lidar com diferentes níveis de mobilidade de dados, a SCC implementa 3 modos de operação [27], (i) SCC1 – o endereço IP é mantido independente da mobilidade, ou seja, a mesma UPF se mantém acessível para ser usada por toda a rede; (ii) SSC2 – a rede irá liberar e acionar o dispositivo para reestabelecer novas sessões conforme sua mobilidade na rede, ao contrário do SSC1, o SSC2 implica em interrupção do serviço, o que pode, ou não, ser aceitável; e (iii) SSC3 – tenta combinar os benefícios de SSC1 e SSC2, permitindo baixos atrasos por meio da liberação e reestabelecimento das sessões, mas também permitindo disponibilidade de serviço contínuo, o que é obtido mantendo a conexão estabelecida com a UPF de origem até que a sessão seja confirmada na nova UPF de destino. O SSC3 impõe requisitos adicionais ao dispositivo que precisará manter duas sessões para o mesmo serviço por um período de tempo [27, 7].

- ***User Plane Security Enforcement information***: Campo destinado para indicar se algum material criptográfico e de integridade de dados está ativo para sessão PDU específica.

Dentro da perspectiva dos sistemas 5G, o gerenciamento de sessão tem como funcionalidade principal o gerenciamento do plano de usuário para as sessões PDU. O plano de usuário trata-se da comunicação Fim-A-Fim entre o UE e a rede de dados para transporte dos dados proveniente das aplicações do usuário. Os procedimentos necessários para implantar o plano de usuário na estrutura de telecomunicação da rede 5G, pode ser dividido em três etapas: (i) o UE, antes de se conectar à rede da dados, deve se conectar com a tecnologia de acesso da rede 5G, podendo ser via RAN em caso de acesso 3GPP ou através da N3IWF para não-3GPP; (ii) deve ser estabelecida uma conexão com o plano de dados entre a rede de acesso e um serviço de dados UPF, via interface de referência N3; (iii) se necessário, devem ser estabelecidos outros saltos no núcleo da rede 5G por várias UPFs onde a conexão será estabelecida por um processo de ancoragem de sessão PDU (*PDU Session Anchor* - PSA) por meio de interface de referência N9, continuando até o acesso final na rede de dados DN por meio da interface de referência N6.

A Figura 1.14 apresenta a topologia com a pilha de protocolos do plano de usuário devidamente estabelecida para as sessões PDU. Para acessos 3GPP, a conexão do UE passa

primeiro pela RAN representada pela comunicação do UE com a nova interface de rádio em sistemas 5G (gNB) via interface de referência uU. Para acessos não-3GPP, a conexão do UE é estabelecida diretamente com o 5GC, via função de interconexão de rede N3IWF via interface de referência NwU. O acesso não-3GPP permite uma flexibilização maior das redes 5G para conectar redes heterogêneas fora do escopo da 3GPP, com uma maior quantidade de casos de uso, por exemplo, aplicações IoT [7].

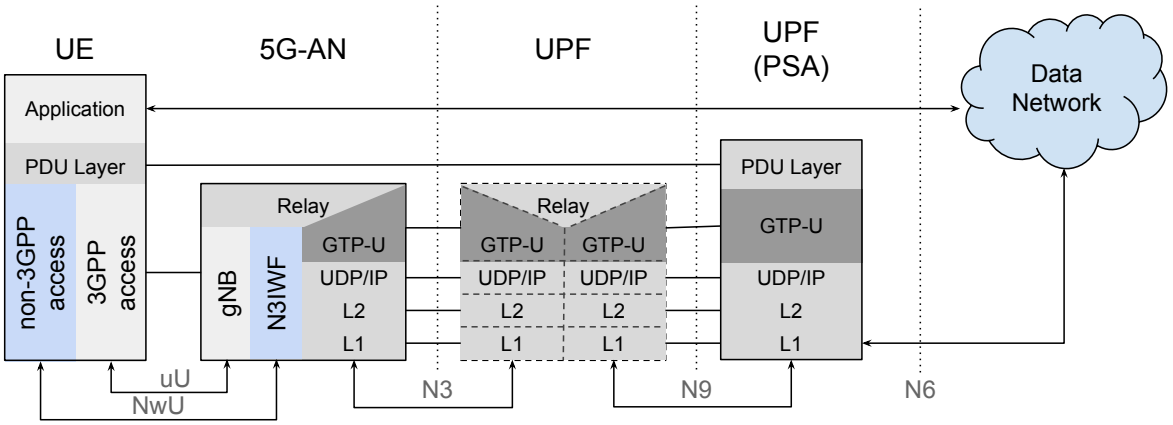


Figura 1.14 – Pilha de protocolos do plano de usuário para sessões PDU.

Ainda sobre o plano de usuário relacionado a Figura 1.14, entre as interfaces de referência N3 e N9, os dados trafegam por meio de tunelamento do protocolo GTP-U (*GPRS Tunnelling Protocol User Plane*) via camada de transporte UDP (vide Figura 1.14). O GTP-U é utilizado unicamente no contexto das redes 5G para transportar dados relativos ao plano de usuário. Os túneis GTP-U são utilizados entre dois nós de rede GTP-U para separar o tráfego em diferentes fluxos de conexão. Dessa forma, um caminho entre a rede de acesso (RAN ou N3IWF) e a UPF é estabelecido com um endereço IP e uma porta UDP para estabelecimento da conexão, e o código de identificador local TEID (*Tunnel Endpoint Identifier*). O TEID está presente no cabeçalho do GTP-U para indicar qual túnel particular dentro da topologia do plano do usuário que os *payloads* vão trafegar. No caminho GTP-U, podem existir múltiplos nós UPF via interface de referência N9 para uma melhor disposição de roteamento e de recursos de rede PSA até a saída para rede de dados via interface N6. Neste contexto, a próxima subseção descreve os principais protocolos e interfaces pelo qual as funções de rede do núcleo 5G e com o UE interagem entre si.

1.2.5 Principais Protocolos e Interfaces de Referência

O **Estrato sem acesso (*Non-Access Stratum* - NAS)** é um conjunto de protocolos para a transmissão de sinalização entre o UE e a AMF via interface N1. O acesso pode ser 3GPP (por exemplo, gNB), via interface de referência *Uu*, ou não-3GPP (por exemplo, Wi-Fi, LoRaWAN), via interface de referência *NWu*, conforme ilustrado na Figura 1.15. Entre as funções do protocolo NAS destacam-se, (i) o tratamento de registro de UE e mobilidade, o qual inclui controle de acesso (gerenciamento de conexão, autenticação, tratamento de

segurança NAS, identificação e configuração de UE), (ii) o suporte de procedimentos de gerenciamento de sessão (estabelecimento e manutenção da conectividade da sessão PDU, e QoS para o plano do usuário entre o UE e a DN) e (iii) o transporte de outros tipos de mensagens entre UE e a AMF (por exemplo, mensagens SMS, protocolos de serviços de localização, dados a UDM, mensagens de orientação de *roaming*, e políticas de UE) [26, 7].

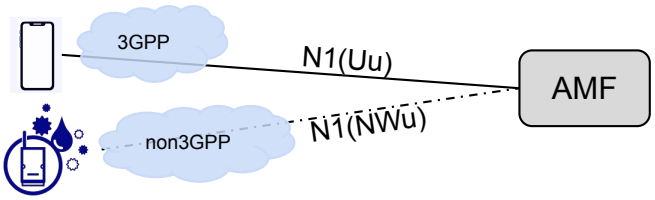


Figura 1.15 – Protocolo NAS.

O NAS consiste de dois protocolos básicos para suportar essas funcionalidades, o protocolo de gerenciamento de mobilidade do 5G (*5G Mobility Management* - 5GMM) e o protocolo de gerenciamento de sessões 5G (*5G Session Management* - 5GSM). O 5GMM é utilizado entre o UE e a AMF e é basicamente o protocolo NAS usado para atender as requisições de registro, mobilidade, segurança e transporte do protocolo 5GSM, transporte geral de outros tipos de mensagens e também é usado para transportar informações entre UE e a PCF, e entre UE e a SMSF (*Short Message Service Function*). O protocolo 5GSM é executado na interação entre o UE e a SMF (através da AMF), oferecendo suporte ao gerenciamento de conectividade de sessão PDU e é transportado sobre o protocolo 5GMM [7].

O **protocolo de aplicação NG (*NG Application Protocol* - NGAP)** é projetado para uso na interface N2 entre a RAN e a AMF. O grupo RAN 3GPP[8] também atribui o nome de NG à interface entre RAN e a AMF que na arquitetura geral do sistema é chamada de N2. O nome do protocolo NGAP é, portanto, derivado do nome da interface NG com a adição de *Application Protocol* (AP), o qual é um termo que foi usado muitas vezes pela 3GPP para denotar um protocolo de sinalização entre duas funções de rede.

O NGAP suporta os mecanismos para lidar com os procedimentos entre a AMF e RAN e também suporta transporte transparente para procedimentos que são executados entre o UE e a AMF ou outras funções da 5GC. O NGAP é aplicável a acessos 3GPP e não-3GPP integrados com 5GC. Isso inclui operações como atualizações de configuração, transferência de contexto de UE, gerenciamento de recursos de sessão PDU e também suporte para procedimentos de mobilidade. NGAP também é usado para transmitir mensagens NAS de *downlink* e *uplink*, bem como suportar operações *CM-IDLE* e *CM-CONNECTED*, por exemplo, na liberação de contexto do UE [28].

A rede de acesso 5G (*5G Access Network* - 5G-AN) e a AMF são conectadas por meio de uma camada de rede de transporte que é usada para transportar a sinalização das mensagens NGAP entre elas. O NGAP depende de um mecanismo de transporte confiável e é projetado para ser executado em cima do *Stream Control Transmission Protocol* (SCTP). O

[8] The TSG Radio Access Network (TSG-RAN) - `ttps://www.3gpp.org/specifications-groups/ran-plenary`

SCTP compartilha características básicas com UDP e TCP. De maneira semelhante ao UDP, o SCTP fornece transferência de dados orientada a mensagens, mas também fornece (de maneira semelhante ao TCP) transporte confiável, garantindo que os dados cheguem ao destino sem erros. Além disso, o SCTP é um protocolo orientado à conexão, o que significa que todos os dados entre dois terminais SCTP são transferidos como parte de uma sessão (ou associação, como é chamada pelo SCTP). Os *endpoints* SCTP na 5G-AN e na AMF configuram associações SCTP entre elas que são identificadas pelos endereços de transporte usados. Uma associação SCTP é genericamente denominada *Transport Network Layer Association* (TNLA) [29, 30].

O **protocolo de controle de encaminhamento de pacotes (*Packet Forwarding Control Protocol* - PFCP)** é utilizado por meio da interface N4 entre o plano de controle (SMF) e o plano de usuário (UPF) no Sistema 5G. O PFCP utiliza UDP e é complementar ao *GPRS Tunneling Protocol* (GTP) que fornece os meios de controle para um componente de sinalização do plano de controle para gerenciar o processamento e encaminhamento de pacotes realizados por um componente do plano do usuário (*GTP for the User Plane* - GTP-U). [31, 27]. Existem dois tipos de procedimentos PFCP, relacionados ao nó e relacionados à sessão. Os procedimentos relacionados ao nó são usados para estabelecer uma associação de nó entre a SMF e a UPF e ainda para enviar informações no nível do nó entre a SMF e a UPF. Os procedimentos relacionados à sessão são usados para gerenciar as sessões PFCP correspondentes a sessões PDU individuais. Uma Associação PFCP de nível de nó é configurada entre a SMF e a UPF antes de se estabelecer quaisquer sessões PFCP na UPF. Ambas, UPF e SMF, podem tomar a iniciativa de uma associação PFCP, sendo obrigatória quando iniciada pela SMF e opcional quando iniciadas pela UPF [7].

Como já foi descrito o núcleo 5G, a próxima seção apresenta os procedimentos de um sistema 5G para que os serviços sejam ofertados e consumidos.

1.3 5G em ação: Procedimentos de um sistema 5G

Na especificação técnica (TS) 23.502 [32], a 3GPP define uma série de procedimentos que devem ser suportados por um sistema 5G. Os procedimentos são fluxos que definem sequências de mensagens a serem trocadas, bem como ações a serem executadas dependendo da chegada ou retorno de cada uma delas. As NFs no núcleo, RAN e UE, devem implementar as partes que lhes correspondem desses procedimentos, para que assim consigam interagir e serviços possam ser ofertados e consumidos. A seguir, são abordados os procedimentos de registro, cancelamento de registro, requisição de serviços e estabelecimento de sessão PDU.

1.3.1 Registro

O **procedimento de registro** é uma etapa obrigatória para que o UE consiga acessar a maioria dos serviços em um sistema 5G. Esse procedimento é executado em diferentes

momentos, dependendo do tipo de acesso (3GPP ou não-3GPP) ou estados 5GCM e 5GRM. A seguir, apresentamos uma lista dos principais tipos de registro:

- **Registro Inicial (3GPP e não-3GPP)**: executado quando o UE está em *RM-DEREGISTERED* e deseja acessar serviços que exigem registro, por exemplo, quando o dispositivo de usuário é ligado.
- **Atualização de Registro (3GPP e não-3GPP)**: executado quando o UE está em *RM-REGISTERED*, mas precisa atualizar algum dos parâmetros que são negociados durante o procedimento de registro inicial ou em reposta a alguma solicitação da rede, como atualização de configuração de UE, informação de assinante, política ou configuração de *slice*.
- **Registro Emergencial (3GPP e não-3GPP)**: executado quando o UE somente precisa acessar serviços emergenciais.
- **Atualização de Registro de Mobilidade (3GPP)**: executado quando um UE em *RM-REGISTERED* sai da sua área de registro (área de cobertura). Como a 3GPP não prevê mobilidade nos acessos não-3GPP, somente é executado em acessos 3GPP.
- **Atualização Periódica de Registro (3GPP)**: executado periodicamente pelo UE em *RM-REGISTERED*, para garantir que o UE continue alcançável a partir da rede 5G, ou seja, é uma forma de dizer que continua ativo. Os acessos não-3GPP não usam o registro periódico e, pelo contrário, definem um temporizador para cancelamento de registro caso a conexão fique em *CM-IDLE* por muito tempo.

A Figura 1.16 mostra uma versão simplificada do procedimento de registro inicial, focando nas partes mais importantes e ignorando detalhes da conexão entre UE e RAN, que variam para acessos 3GPP e não-3GPP. A seguir, encontra-se uma descrição das principais etapas de registro:

- **Requisição de Registro:** representa o início do procedimento de registro e deve conter os parâmetro da rede de acesso, algum identificador de usuário, parâmetros de segurança, o código do *slice* desejado, além de várias outras informações opcionais.
- **Estabelecimento da N1:** durante o registro, é feito o estabelecimento da conexão de sinalização entre UE e rede de acesso, usando *Radio Resource Control* (RRC) para acessos 3GPP e IPSec para acessos não-3GPP, além da conexão entre rede de acesso e AMF na interface de referência N2, através da associação NGAP de UE.
- **Criação do contexto de usuário:** é criado o contexto de usuário para guardar informações sobre o usuário, como identificadores além dos estados 5GCM, 5GRM e sessões PDU estabelecidas, os quais são independentes para os acessos 3GPP e não-3GPP. O contexto pode ser armazenado na UDSF ou na própria AMF caso a UDSF não esteja presente.

- **Autenticação:** durante o registro, é feita a autenticação do usuário. Para isso a AMF interage com outras NFs como AUSF e UDM, passando o identificador de usuário e de *slice* que vieram na requisição de registro NAS.

- **Aplicação de políticas de UE e de gerenciamento de acesso (*Access Management* - AM):** opcionalmente, a AMF pode interagir com a PCF para criar e aplicar políticas de usuário (*UE Policy*) e gerenciamento de acesso (*AM Policy*).

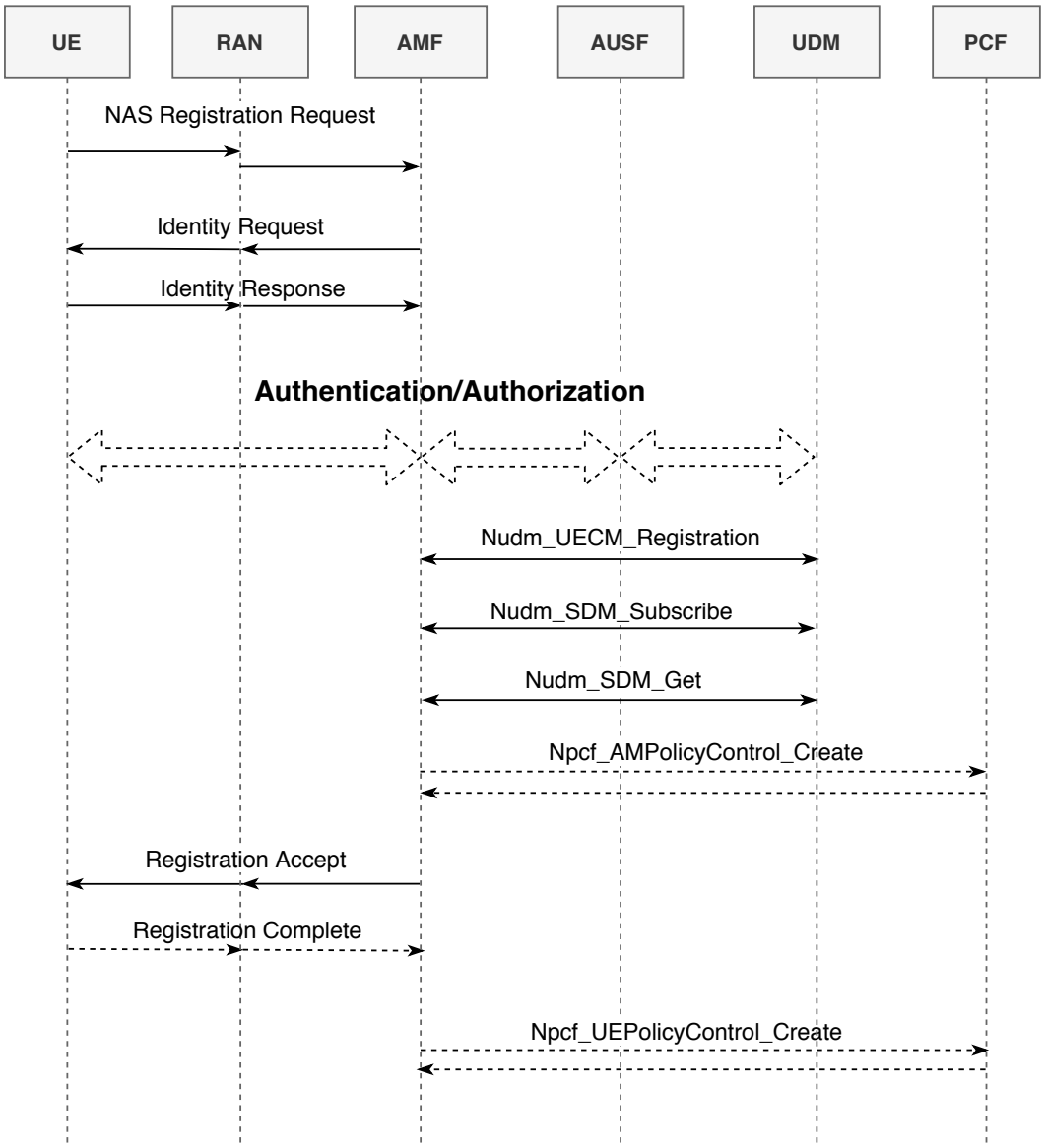


Figura 1.16 – Procedimento de Registro Inicial Simplificado.

Ao fim do procedimento de registro inicial, com o recebimento pelo UE da mensagem *Registration Accept* a partir da AMF, o UE entra no estado *CM-CONNECTED* e *RM-REGISTERED*. O UE também recebe uma lista de TA suportadas pela AMF que formam a área de registro para esse UE. A partir desse momento, os outros tipos de registro irão continuar ocorrendo para garantir que o UE continue ativo e alcançável a partir da rede ou quando houver

mudança de área de cobertura. Além disso, quando UE ou AMF ficarem em *CM-IDLE* por muito tempo, ambos podem iniciar o procedimento implícito de cancelamento de registro.

1.3.2 Cancelamento de Registro

O **procedimento de cancelamento de registro** é utilizado para retirar acesso do UE ao sistema 5G e pode ser iniciado tanto pelo UE quanto pela rede [7]. Os cenários mais comuns em que esse procedimento é executado são:

- **Cancelamento de Registro iniciado pelo UE**: quando o UE decide que não precisa mais estar registrado na rede. Esse procedimento é comumente executado quando o UE é desligado (*switch off*), porém pode ser executado pelo UE a qualquer momento.
- **Cancelamento de Registro iniciado pela Rede**: quando a rede decide que o UE não deve mais estar registrado. A rede pode decidir cancelar o registro do UE a qualquer tempo, por exemplo, por alguma atualização dos dados do assinante, mudança de política ou configuração de *slice*.
- **Temporizador para cancelamento de registro implícito**: quando o temporizador de cancelamento de registro expira na AMF, ela pode executar o procedimento de cancelamento de registro do UE. Esse temporizador é disparado toda vez que UE ou AMF entram em *CM-IDLE*, sendo cancelado quando retornam para *CM-CONNECTED*, ou executam o procedimento implícito de cancelamento de registro quando o temporizador expira. Esse tempo de espera do temporizador é enviado para o UE pela AMF durante o procedimento de registro e deve ser maior na AMF do que no UE, para que dê tempo do UE refazer a conexão de sinalização de controle antes que a AMF inicie o cancelamento de registro.

A Figura 1.17 apresenta o fluxo simplificado para procedimento de cancelamento de registro iniciado pelo UE. As trocas de mensagens estão agrupadas em 5 etapas que seguem descritas a seguir:

- **(A):** O procedimento de cancelamento de registro inicia com o envio da mensagem *Deregistration Request* do UE para a AMF. Essa mensagem deve conter a identificação temporária do usuário (5G-GUTI), o tipo de cancelamento de registro (*Normal ou Switch-off*) e o tipo de acesso (3GPP, não-3GPP ou ambos).
- **(B):** A AMF envia uma requisição à SMF para que libere os recursos de plano de usuário associados às sessões PDU ativas. Além disso, as políticas de gerenciamento de sessão também são removidas e a SMF cancela sua inscrição junto à UDM para não receber mais notificações sobre atualização de dados do assinante.
- **(C):** Caso haja associações de políticas de gerenciamento de acesso ou de UE estabelecidas, a AMF solicita à PCF que elas sejam removidas.

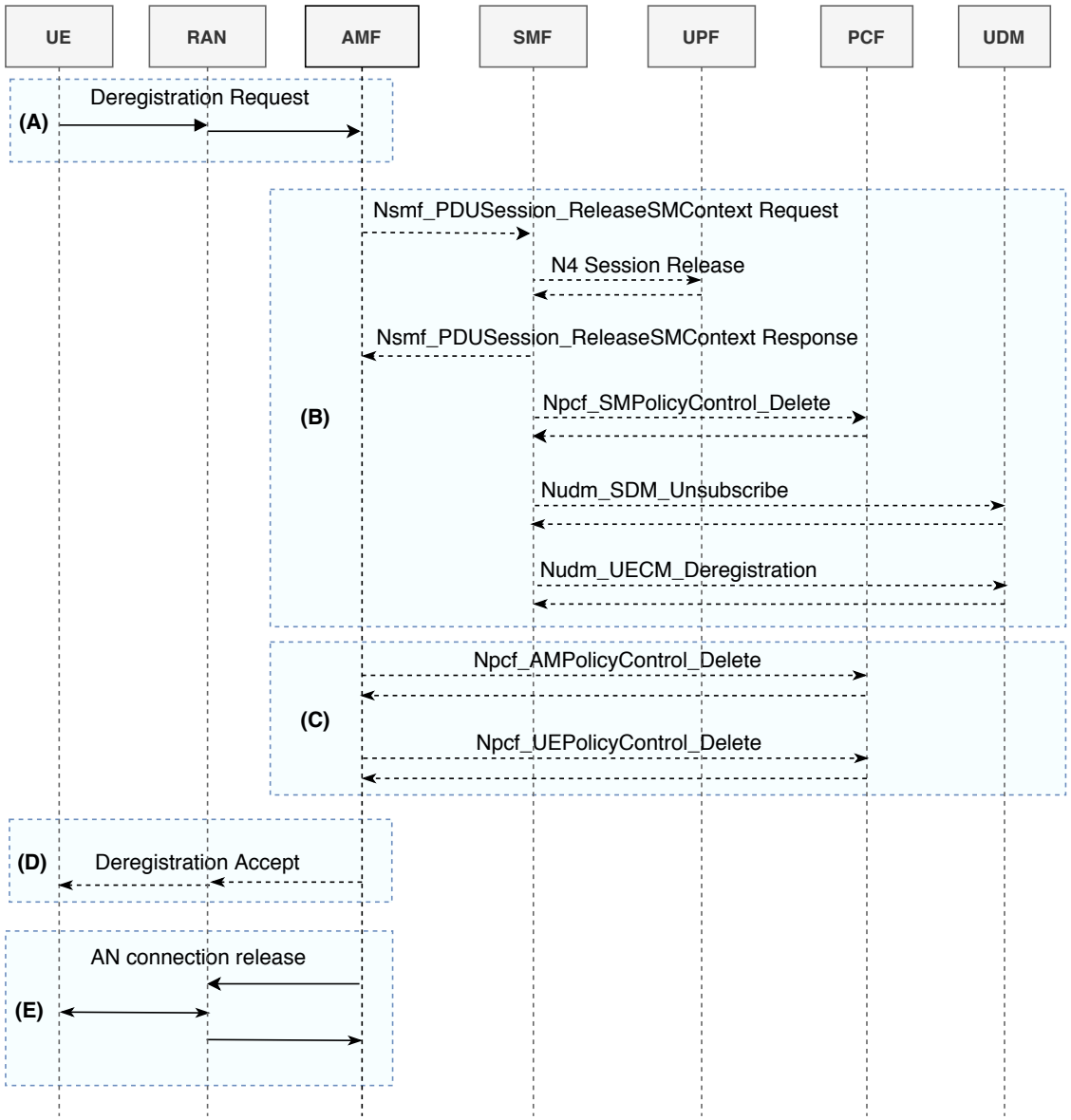


Figura 1.17 – Procedimento de cancelamento de registro iniciado pelo UE (Adaptado de [7]).

- **(D):** O recebimento da mensagem *Deregistration Accept* pelo UE indica que os recursos junto ao núcleo já foram devidamente liberados e removidos. Essa mensagem é opcional porque em caso alguma falha e seu não recebimento, o UE prosseguirá com a mudança para *RM-DEREGISTERED*. Isso ocorre porque os temporizadores na AMF farão com que ela inicie o procedimento de cancelamento de registro quando o temporizador expirar.

- **(E):** As conexões de sinalização do plano de controle (N1) são liberadas e o UE entra em *CM-IDLE*.

1.3.3 Requisição de Serviços

O **procedimento de requisição de serviço** é usado para estabelecer a conexão de sinalização entre UE e AMF e pode ser iniciado tanto pelo UE quanto pela rede. Se for executado pelo UE em *CM-IDLE*, traz o UE para *CM-CONNECTED* após o estabelecimento da N1. Caso seja executado pelo UE em *CM-CONNECTED*, faz com que as conexões de plano de usuário das sessões PDU estabelecidas sejam ativadas. A Figura 1.18 mostra o fluxo simplificado da requisição de serviço iniciada pelo UE. As funções de cada uma das 5 etapas seguem descritas a seguir:

- **(A)**: o procedimento inicia com o envio de uma mensagem NAS de Requisição de Serviço pelo UE para a AMF. A mensagem contém os parâmetros da rede de acesso, de segurança e a lista de sessões PDU caso o UE queira estabelecer as conexões de plano de usuário para elas.
- **(B)**: durante a requisição de serviço, a rede pode exigir que o UE se autentique novamente. Essa autenticação segue os mesmos passos do procedimento de registro.
- **(C)**: caso a mensagem de Requisição de Serviço contenha informações sobre sessões PDU com conexões de plano de usuário para serem ativadas, a AMF notifica a SMF responsável por cada uma das sessões. Além disso, a AMF inclui as informações das sessões PDU na mensagem Serviço Aceito (*Service Accept*) retornada.
- **(D)**: nessa etapa são estabelecidas a associação NGAP de UE entre rede de acesso e AMF, e a conexão de plano de usuário para esse UE.
- **(E)**: a AMF então notifica cada SMF novamente para informar o resultado do estabelecimento do plano de usuário e identificador do túnel com a rede de acesso. Caso a PCF tenha se inscrito para receber informações sobre a localização do UE, a SMF notifica a PCF sobre a nova localização do UE.

1.3.4 Estabelecimento de Sessão PDU

A sessão PDU representa uma conexão lógica entre o UE e a rede de dados (plano de usuário) em sistemas 5G. No contexto do gerenciamento de sessão, cada sessão PDU estabelece uma associação com somente uma única rede de dados que é definida pelo UE por meio do parâmetro DNN.

O processo de estabelecimento de sessão PDU incia-se com a condução do plano de controle para requisição de nova sessão PDU. A Figura 1.19 apresenta o fluxo de operação com componentes chave da estrutura da rede 5G que fazem parte da execução do plano de controle. Por exemplo, para o caso de uso do acesso via 3GPP, o UE sempre deve iniciar os procedimentos de estabelecimento de sessão, gerando uma nova mensagem de requisição para estabelecimento de sessão PDU. UE inicia o processo de requisição de estabelecimento

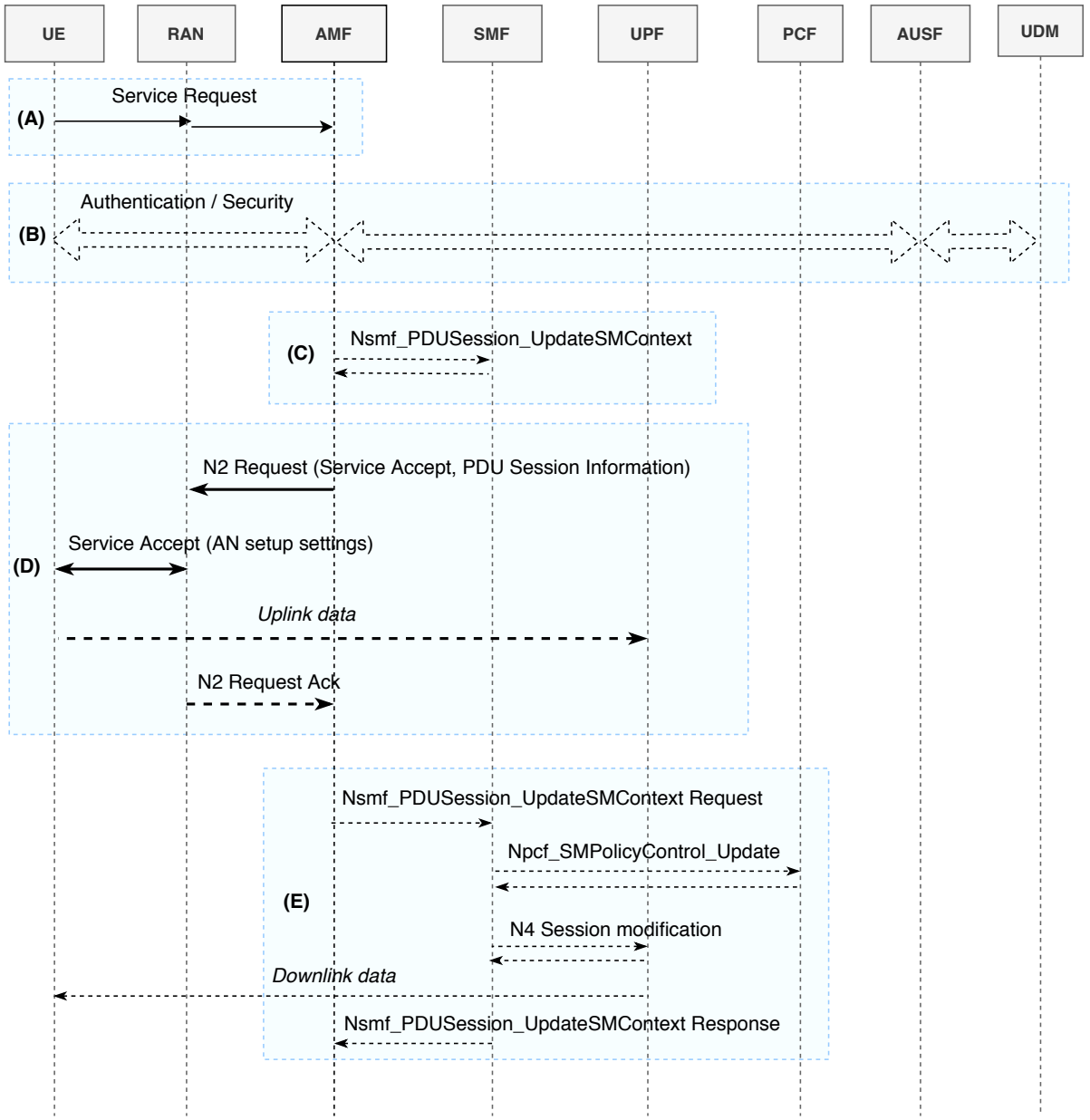


Figura 1.18 – Requisição de Serviço iniciado pelo UE [7].

de sessão PDU enviado uma mensagem de sinalização NAS (*PDU Session Establishment*) via interface de referência N1. No corpo da mensagem, estão os parâmetros de requisição PDU conforme dispostos na Seção 1.2.4. Essa requisição passa pela RAN até a AMF exposto no núcleo da rede. Para o processo de estabelecimento de sessão PDU é assumido que o UE já está registrado na AMF, conforme os procedimentos listados na Seção 1.3.1.

Na Figura 1.19 em (1), ao receber a mensagem de requisição para estabelecimento de sessão PDU, a AMF deve primeiro verificar se a mensagem é de uma nova requisição, analisando se o campo do tipo da requisição é de uma requisição inicial e se o ID da sessão PDU não está alocado por alguma outra sessão PDU no núcleo da rede. No caso de uma nova requisição, o próximo passo é prosseguir para o gerenciamento de sessão, para isso, a AMF deve selecionar para qual SMF deverá repassar a mensagem de estabelecimento de

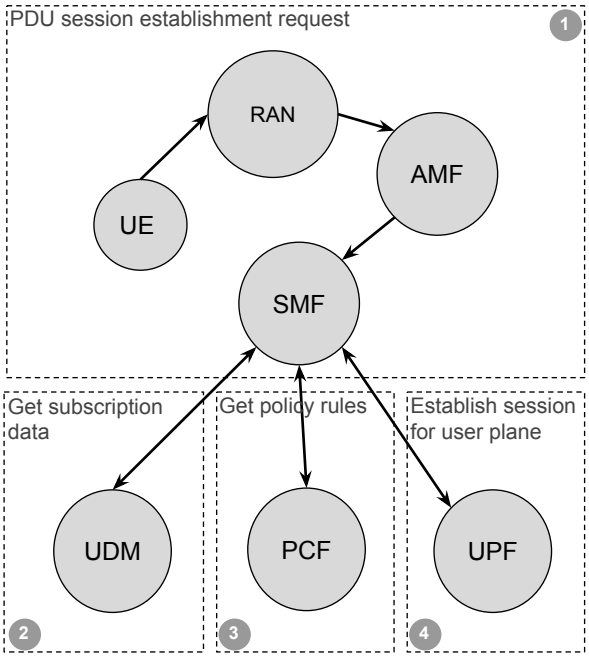


Figura 1.19 – Procedimentos para requisição de estabelecimento de sessão PDU.

sessão PDU, por meio da descoberta de serviços de rede da NRF.

A AMF então remonta os parâmetros da nova mensagem de sessão PDU e os associa ao ID da SMF, repassando para o controle do gerenciamento de sessão do núcleo 5G. Para acionar o gerenciamento de sessão, a AMF envia uma mensagem de requisição *NsmfPDUSessionCreateSMContext* para a SMF com os parâmetros relacionados ao pacote de estabelecimento de sessão PDU original, tais como o identificador único do UE (*Subscription Permanent Identifier* - SUPI), a DNN, o código de seleção do *slice* da rede S-NSSAI, o ID da sessão PDU, entre outros [32]. Ao receber a requisição *NsmfPDUSessionCreateSMContext* da AMF, a SMF deve proceder a verificação dos parâmetros de subscrição e autenticação do UE, ilustrado em (2) na Figura 1.19. A SMF então submete à UDM uma requisição para verificar se os paâmetros de sessão PDU repassados pela AMF já estão registrados. Caso contrário, a UDM procederá o armazenamento dos dados de subscrição da sessão PDU, para isso a SMF envia uma requisição *NudrDMUpdate* para a UDR com os parâmetros da sessão PDU e envia uma resposta à AMF com o código de identificação do novo contexto para o gerenciamento de sessão armazenado no núcleo da rede [32].

Após etapa de verificação das credenciais, a SMF inicia os procedimentos para estabelecimento da sessão PDU junto à PCF (políticas relacionadas a métricas para cobrança, aplicação de qualidade de serviço para o UE e regras de associação de operação do fluxo de dados e ancoragem para a UPF) [32], ilustrado em (3) na Figura 1.19. Em seguida, a SMF envia uma requisição de estabelecimento de sessão para condução do plano de usuário UPF, com a regras de conectividade da sessão PDU que deverão ser aplicadas na UPF selecionada, conforme ilustrado em (4) na Figura 1.19.

Seguindo para construção do plano de usuário, a SMF inicialmente envia para a rede de acesso por radio RAN, uma mensagem de requisição para configuração de recursos de rádio.

A lista completa de parâmetros contidos na mensagem de requisição pode ser consultada em [32]. Seguindo o fluxo de operação para o plano de de usuário, é iniciada uma troca de sinalização entre a RAN e do UE relacionada a comandos de protocolos de rádio para gerenciamento de alocação de recursos de rádio necessários para condução do plano de usuário, por exemplo, reconfiguração dos parâmetros de rádio para aplicação de regras de QoS relacionadas à requisição sessão PDU vinda da SMF para a qual se destina a conexão de acesso [32].

Para completar o plano do usuário por parte da rede de acesso, é realizado então o estabelecimento do túnel GTP-U por meio da interface de referência N3 correspondente à sessão PDU, ilustratdo em (3) na Figura 1.20. Ao finalizar os procedimentos para estabelecimento dos recursos de rádio e também construção do túnel de dados na interface N3, a RAN deve enviar uma mensagem de confirmação para a SMF e no corpo da mensagem deve conter o ID da sessão PDU e informação que túnel de dados foi devidamente estabelecido na rede acesso 5G.

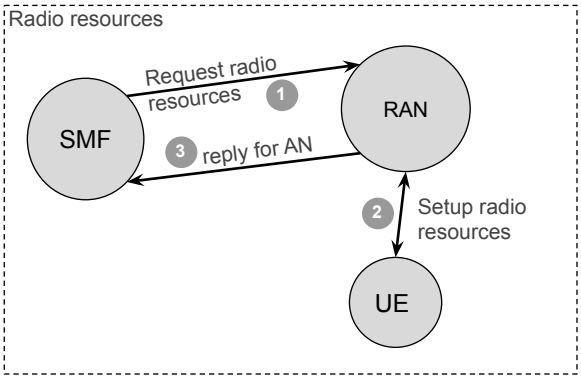


Figura 1.20 – Procedimentos para configuração de recursos de rádio.

Na Figura 1.21, é apresentada a condução final do plano de usuário. A SMF ao receber informações de configuração de túnel da rede de acesso RAN, começa um processo de negociação via interface de referência N4, iniciando os procedimentos de modificação com a UPF selecionada para servir ao plano de dados da sessão PDU. Basicamente, a SMF providência informação relativa a construção do túnel GTP-U com a rede de acesso e também regras do repasse e roteamento para sessão PDU, em especifico, até a rede de dados selecionada para servir no transporte dos dados provenientes do UE. Desta forma, a próxima seção discute um caso de uso, com vistas ao desenvolvimento completo, na prática

1.4 Desenvolvimento para o my5GCore: Caso de uso IoT não-3GPP

O caso de uso 5GC desse minicurso visa apresentar os processos de instalação, configuração, preparação do ambiente e o desenvolvimento de software no contexto do 5GC,

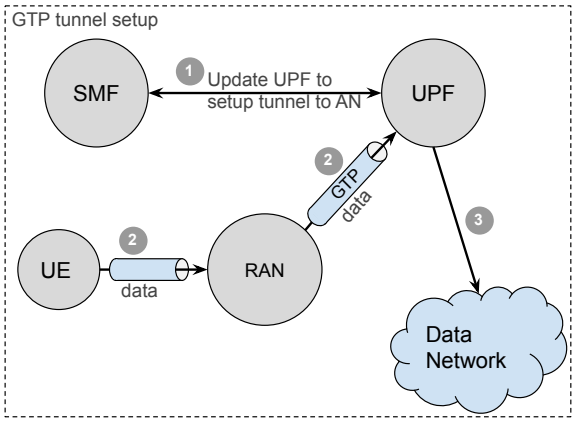


Figura 1.21 – Procedimentos para estabelecimento de conectividade de dados.

adotando como núcleo da rede o my5GCore. Além disso, o caso de uso IoT não-3GPP tem como objetivo apresentar os resultados parciais do desenvolvimento de um equipamento de usuário IoT não-3GPP, integrado ao 5GC. Nesse contexto, apresentamos os processos de instalação e configuração no my5GCore, a instalação e configuração do ambiente necessário ao desenvolvimento e testes, um *tour* pelo código my5GCore e, finalmente, o processo empregado no desenvolvimento do UE-IoT-não-3GPP.

O propósito do caso de uso em estabelecer o acesso IoT não-3GPP tem como benefício expandir as funcionalidades iniciais de comunicação do sistema 5G para que sistemas de comunicação heterogêneos IoT possam ser utilizados por meio da integração com o núcleo da rede 5G. Além disso, esse caso de uso permite a expansão da possibilidade de comunicação sob espectro de frequência de bandas não licenciadas. Dessa forma, as vantagens em utilizar diferentes tecnologias de comunicação sem fio IoT, mesmo com os avanços significativos trazidos pelo novos sistemas 5G, está relacionada a convergência de diferentes redes de telecomunicação sem fio, i.e., utilizando espectro de frequências licenciadas e não licenciadas.

Tratando especificamente de tecnologia de comunicação para Internet das coisas, o uso de redes LPWA não-3GPP para conectar centenas de milhares de dispositivos IoT, se mostra eficiente em relação a parâmetros de desempenho, tais como consumo de energia [33]. Além disso, a capacidade de integração de tecnologias heterogêneas ao núcleo da rede 5G mostra vantagens para operadores de telecomunicações públicos ao permitir que novas fontes de serviços sejam ofertadas.

Para realizar os procedimentos de acesso via não-3GPP com a estrutura do núcleo da rede 5G, o Lançamento 15 especifica o padrão para acesso não confiável. Seguindo as especificações técnicas da N3IWF, os passos para realizar o acesso não confiável são determinados por meio de protocolos para gerenciamento de contexto de associação segura para condução do plano de controle e plano de dados.

1.5 Material de apoio em repositório público

Como material complementar do minicurso, criamos demonstrações que incluem a execução passo-a-passo dos procedimentos de registro, cancelamento de registro e estabelecimento de sessão PDU. Além disso, são realizadas a captura e análise de tráfego de rede, enfatizando especialmente as pilhas de protocolos envolvidos na comunicação do UE com a rede de acesso e núcleo. Os túneis criados para o tráfego de dados via plano de usuário e para implantação de um canal seguro para o acesso não-3GPP também são mostrados e discutidos. Finalmente, a comunicação da SMF com a UPF, contendo as instruções para processamento e encaminhamento de pacotes para um UE é apresentada.

Todas as demonstrações apresentadas, códigos, procedimentos de instalação, configuração, implantação, vídeos, bem como o respectivo material de apoio para sua execução estão disponíveis no seguinte repositório público:
`https://github.com/LABORA-INF-UFG/SBrT2020-Minicurso6`.

Conclusões

Este minicurso apresentou o núcleo 5G (my5GCore) na prática através de uma implementação de código aberto da iniciativa my5G que pretende tornar o sistema 5G acessível para todos. Além disso, foi discutido os fundamentos de softwarização em 5G e a tendência de integração dos conceitos da Tecnologia de Informação com a área de telecomunicação. Posteriormente, apresentou-se detalhadamente o sistema 5G, enfatizando os procedimentos de registro, requisição de serviços e estabelecimento de uma sessão PDU, bem como as funções virtualizadas definidas no Lançamento 15 e complementadas no Lançamento 16 da 3GPP. Adicionalmente, apresentou-se um caso de uso IoT não-3GPP, descrevendo os processos de instalação, configuração, preparação do ambiente e o desenvolvimento de software no contexto do 5GC.

Uma grande expectativa surge em relação aos sistemas 5G, seus benefícios e as inúmeras oportunidades acadêmicas e profissionais que virão. O arcabouço ferramental do my5GCore e a base teórica e prática disponibilizada pela iniciativa my5G podem subsidiar o desenvolvimento de negócio, indústria, teses, dissertações e artigos. Por exemplo, o my5GCore é apenas um dos projetos sendo desenvolvidos no my5G. Realizamos esse minicurso com objetivo de disseminar e permitir que todos possam acessar as tecnologias discutidas nesse minicurso, dessa forma, esperamos que todos usem esse material e estejam sempre atentos as atualizações disponibilizadas no nosso repositório.

Referências Bibliográficas

[1] A. Osseiran, J.F. Monserrat, and P. Marsch. *5G Mobile and Wireless Communications Technology*, volume 1. Cambridge University Press, Cambridge, United Kingdom, 1 edition, 2016.

[2] A. Ahad, M. Tahir, and K. A. Yau. 5g-based smart healthcare network: Architecture, taxonomy, challenges and future research directions. *IEEE Access*, 7:100747–100762, 2019.

[3] D. Kombate and Wanglina. The internet of vehicles based on 5g communications. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 445–448, 2016.

[4] C. R. Storck and F. Duarte-Figueiredo. A survey of 5g technology evolution, standards, and infrastructure associated with vehicle-to-everything communications by internet of vehicles. *IEEE Access*, 8:117593–117614, 2020.

[5] L. Chettri and R. Bera. A comprehensive survey on internet of things (iot) toward 5g wireless systems. *IEEE Internet of Things Journal*, 7(1):16–32, 2020.

[6] D. Moongilan. 5g internet of things (iot) near and far-fields and regulatory compliance intricacies. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 894–898, 2019.

[7] P. Hedman, L. Frid, S. Rommer, M. Olsson, S. Sultana, and C. Mulligan. *5G Core Networks: Powering Digitization*, volume 1. Elsevier Science & Technology, United Kingdom, 1 edition, 2019.

[8] 3GPP. 3rd generation partnership project (3gpp). `https://www.3gpp.org/` Accessed: 2020-10-02.

[9] J.M. Meredith and M. Pope. 3rd generation partnership project technical specification group services and systems aspects release 15 description. Technical Report 3GPP TR21.915 V15.9.0, 3rd Generation Partnership Project-3GPP, Sophia Antipolis CEDEX, France, 12 2018. `https://www.3gpp.org/ftp/Specs/archive/21_series/21.915/`.

[10] J.M. Meredith. 3rd generation partnership project technical specification group services and systems aspects release 16 description. Technical Report 3GPP TR21.916 V16.4.0, 3rd Generation Partnership Project-3GPP, Sophia Antipolis CEDEX, France, 03 2020. `https://www.3gpp.org/ftp/Specs/archive/21_series/21.916/`.

[11] X. Foukas, G. Patounas, A. Elmokashfi, and M.K. Mahesh. Network slicing in 5g: Survey and challenges. *IEEE Communications Magazine*, 55(5):94–100, 2017.

[12] J. Costa-Requena, V. Guasch, and J. L. Santos. Software defined networks based 5g backhaul architecture. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, IMCOM '15, New York, NY, USA, 2015. Association for Computing Machinery.

[13] M. Anan, A. Al-Fuqaha, N. Nasser, T.Mu, and H.Bustam. Empowering networking research and experimentation through software-defined networking. *J. Netw. Comput. Appl.*, 70(C):140 – 155, 2016.

[14] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee. Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2):90–97, 2015.

[15] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati. Network function virtualization in 5g. *IEEE Communications Magazine*, 54(4):84–91, 2016.

[16] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys Tutorials*, 18(1):236–262, 2016.

[17] P. M. Mell and T. Grance. The nist definition of cloud computing. Technical Report SP 800-145, Gaithersburg, MD, USA, 2011. https://bigdatawg.nist.gov/_uploadfiles/M0006_v1_3333767255.pdf.

[18] S. Newman. *Building Microservices*, volume 1. O'Reilly Media, Inc., Sebastopol, CA, United States, 1 edition, 2015.

[19] K. V. Cardoso, C. B. Both, L. R. Prade, C. J. A. Macedo, and V. H. L. Lopes. A softwarized perspective of the 5g networks, 2020. https://arxiv.org/abs/2006.10409.

[20] Nokia. Build a cloud-native 5g core that connects everyone and everything, 2019. https://www.nokia.com/networks/portfolio/5g-core/#defining-a-new-5g-core Accessed: 2020-05-22.

[21] G. Mayer. Restful apis for the 5g service based architecture. *Journal of ICT Standardization*, 6(1):101–116, 2018.

[22] G. Brown. Service-based architecture for 5g core networks. *A Heavy Reading white paper produced for Huawei Technologies Co. Ltd. Online*, 1:2018, 2017. https://www.huawei.com/en/press-events/news/2017/11/HeavyReading-WhitePaper-5G-Core-Network.

[23] Ericsson. Ericsson mobility report, 06 2019. https://www.ericsson.com/en/mobility-report/reports/june-2019, Accessed: 2020-08-12.

[24] K. Kymalainen. Network function repository services. Technical Report 3GPP TS 29.510 V16.4.0, 3rd Generation Partnership Project-3GPP, Sophia Antipolis CEDEX, France, 07 2020. https://www.3gpp.org/ftp/Specs/archive/29_series/29.510/.

[25] S. Arora and K. Kymalainen. 5g system; network exposure function northbound apis. Technical Report 3GPP TS 29.522 V16.4.0, 3rd Generation Partnership Project-3GPP, Sophia Antipolis CEDEX, France, 07 2020. https://www.3gpp.org/ftp/Specs/archive/29_series/29.522/.

[26] F. Firmin and K. Kymalaine. Non-access-stratum (nas) protocol for 5g system (5gs). Technical Report 3GPP TS 24.501 V16.5.1, 3rd Generation Partnership Project-3GPP, 07 2020. https://www.3gpp.org/ftp/Specs/archive/24_series/24.501/.

[27] M. Pope. System architecture for the 5gs. Technical Report 3GPP TS23.501 V15.2.0, 3rd Generation Partnership Project-3GPP, Sophia Antipolis CEDEX, France, 06 2018. https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/.

[28] I. Toufik and J. Krause. Non-ng application protocol (ngap). Technical Report 3GPP TS 38.413 V16.2.0, 3rd Generation Partnership Project-3GPP, Sophia Antipolis CEDEX, France, 08 2020. https://www.3gpp.org/ftp/Specs/archive/38_series/38.413/.

[29] I. Toufik and J. Krause. 3gpp ts 38.410 v16.2.0, ng-ran-ng general aspects and principles. Technical report, 3rd Generation Partnership Project-3GPP, Sophia Antipolis CEDEX, France, 07 2020. https://www.3gpp.org/ftp/Specs/archive/38_series/38.410/.

[30] J. Korhonen and J. M. Meredith. Nr and ng-ran overall description. Technical Report 3GPP TS 38.300 V16.2.0, 3rd Generation Partnership Project-3GPP, Sophia Antipolis CEDEX, France, 2020. https://www.3gpp.org/ftp/Specs/archive/38_series/38.300/.

[31] K. Kymalaine. Interface between the control plane and the user plane nodes. Technical Report 3GPP TS29.244 V16.5.0, 3rd Generation Partnership Project-3GPP, Sophia Antipolis CEDEX, France, 09 2020. https://www.3gpp.org/ftp//Specs/archive/29_series/29.244/.

[32] M. Pope. Procedures for the 5g system (5gs). Technical Report 3GPP TS23.502 V16.6.0, 3rd Generation Partnership Project-3GPP, Sophia Antipolis CEDEX, France, 09 2020. https://www.3gpp.org/ftp//Specs/archive/23_series/23.502/.

[33] C. Bockelmann, N. K. Pratas, G. Wunder, S. Saur, M. Navarro, D. Gregoratti, G. Vivier, E. De Carvalho, Y. Ji, Ä. StefanoviÄ, P. Popovski, Q. Wang, M. Schellmann, E. Kosmatos, P. Demestichas, M. Raceala-Motoc, P. Jung, S. Stanczak, and A. Dekorsy. Towards massive connectivity support for scalable mmtc communications in 5g networks. *IEEE Access*, 6:28969–28992, 2018.

CAPÍTULO

2 Aspectos Teóicos e Práticos de Redes IEEE 802.15.4g SUN

Emerson Brasil Gomes (UFCG), Allan José Bispo do Nascimento (IFPB), Ruan Delgado Gomes (IFPB), Reinaldo Cézar de Morais Gomes (UFCG), Pere Tuset-Peiró (UOC)

Introdução

As tecnologias de redes de sensores sem fio são tipicamente baseadas na camada física (*physical layer*, PHY) e na camada de acesso ao meio (*media access control*, MAC) definidas pelo instituto de engenheiros eletricistas e eletrônicos (*institute of electrical and electronics engineers*, IEEE) [1] no padrão IEEE 802.15.4-2020 [2] para redes sem fio pessoais de baixas taxas de transmissão (*low-rate wireless personal area network*, LR-WPAN). Essas tecnologias permitem implantar redes de baixo custo e baixo consumo de energia para aplicaçóes de sensoriamento ou controle. Porém, o meio de comunicação sem fio apresenta um conjunto de dificuldades, principalmente quando os dispositivos são implantados em locais que possuem objetos altamente reflexivos (metálicos), que provocam problemas de propagação, como sombreamento e atenuação por multipercurso [3]. Problemas de interferência eletromagnética, bem como variaçóes temporais e espaciais na qualidade dos enlaces, também podem ocorrer, sendo necessário lidar com essas dificuldades para prover boa qualidade de serviço.

Para aumentar a confiabilidade na comunicação sem fio, algumas tecnologias empregam comunicações em múltiplos saltos, com rotas redundantes, combinado com salto em frequência (ex: WirelessHART [4], ISA100.11a [5] e 6TiSCH [6]). Nesse tipo de rede, todos os nós da rede precisam atuar também como roteadores e encaminhar pacotes até o nó sorvedouro (*gateway*). No entanto, essas tecnologias requerem alta densidade de nós para operarem adequadamente, o que pode não ser facilmente alcançável em alguns cenários.

Uma alternativa que surgiu recentemente é o uso de redes sem fio de longa distância e baixo consumo de energia (*low-power wide area networks*, LPWAN), como LoRa© e SigFox©. Essas tecnologias tipicamente operam em bandas Sub-GHz (i.e., 915 MHz no Brasil), o que permite a implantação de redes esparsas ou com topologia em estrela com grande área de abrangência. Entretanto, as taxas de dados máximas das redes supracitadas são baixas, sendo de até 50 kbps para as redes Lora© e 0,1 kbps para as redes Sigfox© [3], o que pode inviabilizar seu uso para aplicações mais exigentes com relação a taxas de transferência de dados. Além disso a latência acima de 1000 ms em alguns cenários, tanto para as redes Lora© como para as redes Sigfox© [7], não atende à s necessidades das aplicações industriais mais exigentes em termos de latência. Por fim, tratam-se de padrões proprietários que podem trazer limitações para os usuários finais, com suas patentes e *copyrights*.

Para a implementação de redes LPWAN existe a alternativa de utilização do padrão IEEE 802.15.4-2020, que inclui as novas definições de camada física definidas na emenda IEEE 802.15.4g [8], visando aplicações de redes inteligentes de utilidades (*smart utility networks*, SUN). Trata-se de um padrão aberto, que entrega uma latência a partir dos 20 ms e que também oferece grande flexibilidade, uma vez que dezenas de configurações diferentes de camada física são suportadas, com taxas de bit que variam de 6,25 kbps a 2.400 kbps [2][9], o que permite definir diferentes relações de compromisso entre taxa de bits, alcance e consumo de energia. Além disso, é permitida a operação tanto em bandas Sub-GHz como na banda de 2,4 GHz [2]. A emenda IEEE 802.15.4g também serve como base para uma especificação de redes sem fio inteligentes de utilidades (*wireless smart utility networks*, Wi-SUN), que é uma aliança para especificar, gerenciar e certificar dispositivos de diferentes fabricantes, de modo a atender aos requisitos de interoperabilidade [10].

Tendo em vista essa diversidade de configurações e tecnologias disponíveis, este capítulo tem como objetivo apresentar o padrão IEEE 802.15.4-2020 SUN, com foco nas definições incorporadas a partir da emenda IEEE 802.15.4g. Este capítulo foi elaborado a partir de um minicurso ministrado no XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, sobre aspectos teóicos e práticos de redes IEEE 802.15.4 SUN, para implementação de aplicações de sensoriamento distribuído. São apresentadas as novas definições incluídas na versão de 2015 [11] do padrão IEEE 802.15.4 e que continuam presentes na versão de 2020 do mesmo padrão. Os aspectos apresentados serão evidenciados a partir de resultados obtidos em estudos experimentais. Também é realizada uma discussão sobre os desafios em aberto para implantação deste tipo de rede, e quais são as oportunidades de pesquisa na área. Os objetivos específicos deste capítulo são listados a seguir:

- Apresentar uma visão geral das redes de sensores sem fio, com foco em redes de longo alcance e baixa potência;
- Apresentar a evolução do padrão IEEE 802.15.4;
- Apresentar as novas definições de camada física (IEEE 802.15.4g) e camada de acesso ao meio (IEEE 802.15.4e [12]) introduzidas na versão 2015 do padrão;

- Apresentar resultados experimentais obtidos a partir da implantação de redes IEEE 802.15.4g em diferentes ambientes e com diferentes configurações;
- Discutir os novos desafios para a pesquisa em redes IEEE 802.15.4-2020 SUN, incluindo o desenvolvimento de mecanismos adaptativos (ex: seleção dinâmica de modulação ou de canais) para melhorar a confiabilidade dessas redes;
- Demonstrar a implementação de uma rede IEEE 802.15.4 simples usando as modulações SUN e a plataforma OpenMote-B [13].

2.1 Visão geral das redes de longo alcance e baixa potência (LPWAN)

Com o surgimento da internet das coisas (*internet of things*, IoT) e comunicações máquina a máquina (*machine to machine*, M2M), um crescimento maciço na implantação de nós sensores é esperado, equipando carros, medidores, sensores, terminais de ponto de venda, produtos eletrônicos, vestimentas e as mais variadas máquinas. O crescimento exponencial da IoT está impactando virtualmente todas as áreas da indústria e quase todas as áreas de mercado. Há um imenso potencial para um crescimento também exponencial no desenvolvimento de aplicações baseadas em sensoriamento sem fio em quase todos os setores da sociedade, permitindo conectar sensores, veículos, robôs e máquinas á internet, para os mais variados objetivos [14].

Os dispositivos de IoT usualmente apresentam restrições, como a baixa taxa de transferência de dados disponível e a necessidade de apresentarem um baixo consumo de energia, uma vez que em muitos casos eles são alimentados por baterias. Diversas tecnologias de comunicação podem ser utilizadas para prover comunicação aos dispositivos e cada uma apresenta diferentes características, com relação a diferentes aspectos, como taxa de bits, custo, alcance e consumo de energia. A escolha da tecnologia mais apropriada deve considerar as características do ambiente, dos dispositivos e os requisitos das aplicações.

Na Figura 2.1 [15] é possível visualizar uma comparação entre as principais tecnologias de rádio utilizadas para soluções de IoT. As opções de rádio amplamente utilizadas para curto alcance (ex: ZigBee [16] e Bluetooth) não são adaptadas para cenários que requerem transmissão de longo alcance. Tecnologias mais conhecidas como Wi-Fi e Bluetooth consomem bastante energia. Soluções com base em comunicações celulares (ex: redes 3G, 4G e 5G) podem fornecer cobertura maior, mas seus dispositivos possuem um consumo alto de energia, além de um alto custo fixo com manutenção e assinaturas. Portanto, os requisitos de algumas aplicações de IoT impulsionaram o surgimento de um novo tipo de tecnologia de comunicação sem fio, as redes de longo alcance e baixa potência (LPWAN), que atendem os requisitos de baixo custo, baixo consumo de energia e longo alcance [17], embora em contrapartida ofereçam uma baixa taxa de bits, em comparação com outros padrões.

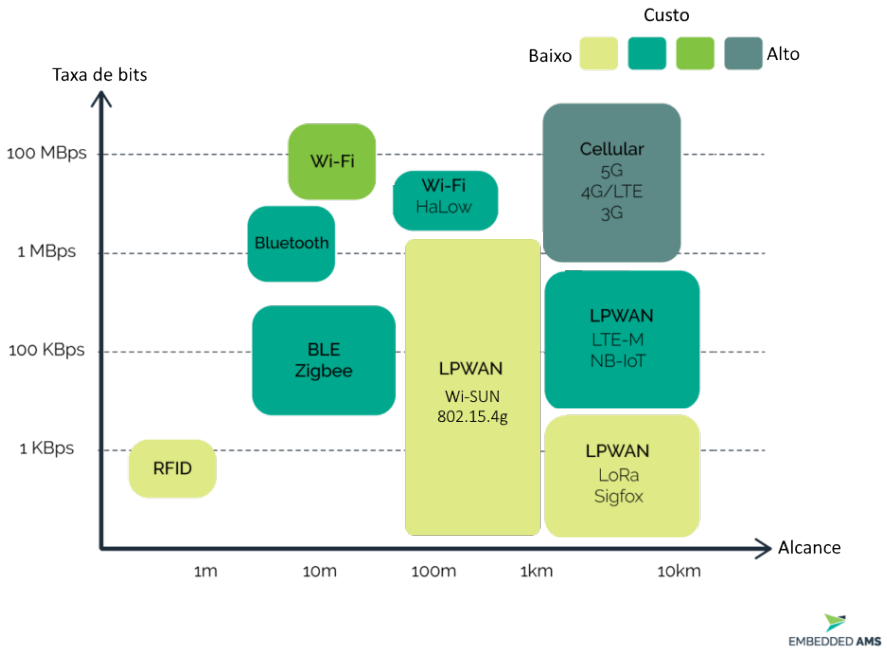


Figura 2.1 – Tecnologias sem fio: Taxa de bits e consumo de energia versus alcance (baseado em [15]).

Com foco específico em algumas tecnologias LPWAN, a Figura 2.2 [7] mostra uma comparação de taxa de bits e o alcance delas. Como pode-se verificar, a tecnologia Wi-SUN é a mais indicada quando se deseja utilizar aplicações que demandam taxas de bits maiores, chegando a 2,4 Mbps em algumas configurações definidas na versão de 2020 do padrão [2]. Por outro lado, ela não é a mais indicada quando os dispositivos estão a distâncias muito grandes, apesar de que essa inferioridade poder ser suplantada implantando dispositivos adicionais e formando uma rede em malha.

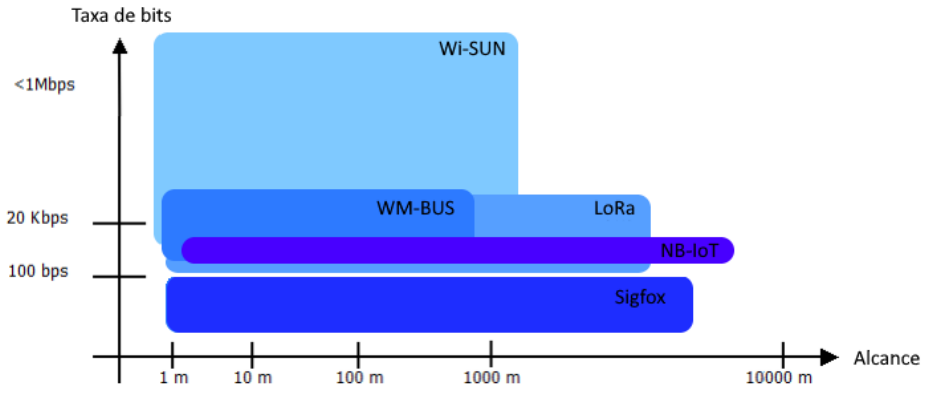


Figura 2.2 – Redes LPWAN: Taxa de bits versus alcance (baseado em [7]).

Na Figura 2.3 é mostrada uma comparação entre as tecnologias LPWAN com relação í latência da rede e ao alcance. Apesar de não atingir distâncias de vários quilômetros, as redes IEEE 802.15.4g/Wi-SUN conseguem oferecer latência reduzida, compatíveis com as exigidas por aplicações de uso crítico, em relação a outras tecnologias LPWAN, que devido a

mecanismos de camada MAC, mantêm os dispositivos em inatividade durante muito tempo, atrasando portanto transmissões e recepções de pacotes [3].

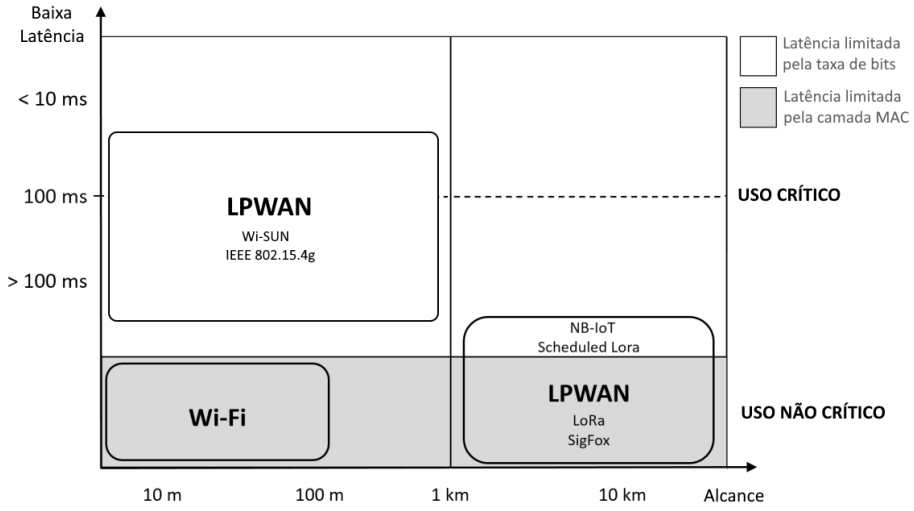


Figura 2.3 – Redes LPWAN: Latência versus alcance (baseado em [3]).

Atualmente existem vários esforços de padronização das redes LPWAN, que vêm sendo realizados por diversos ógãos padronizadores, tais como o IEEE, o instituto de padrões de telecomunicações europeu (*european telecommunications standards institute*, ETSI) [18], o projeto de parceria da terceira geração (*3rd generation partnership project*, 3GPP) [19], e também alguns consócios da indústria, tais como o grupo de interesse especial Weightless (*Weightless special interest group*, WSIG) [20] e as alianças LoRa© [21] e DASH7 [22]. No longo prazo, espera-se que a adoção desses padrões provavelmente reduza a fragmentação do mercado de LPWAN e permita a coexistência de várias tecnologias concorrentes [23]. A Figura 2.4 organiza os padrões propostos de acordo com as organizações que as desenvolve.

2.2 Históia e evolução do padrão IEEE 802.15.4

Motivado pelas crescentes demandas por redes que unissem os requisitos de baixo consumo de energia e de baixo custo, o comitê de novos padrões (*new standards committee*, NESCOM) do IEEE criou oficialmente, em dezembro de 2000, o grupo de trabalho 4 (*task group 4*, TG4), dentro de um grupo maior, o 802.15, que desenvolve padrões para as redes sem fio de alcance pessoal (*wireless personal area networks*, WPAN), ou redes sem fio de curtas distâncias. O TG4 tinha como objetivo principal desenvolver um novo padrão de WPAN, com baixa taxa de transmissão, ou seja, uma *low-rate* WPAN (LR-WPAN). Assim, foi dado o primeiro passo para o que culminaria na criação do padrão IEEE 802.15.4, que fornece conectividade sem fio, com baixa complexidade, baixo custo, baixo consumo de energia e baixa taxa de transferência de dados entre dispositivos portáteis, fixos ou móveis. Pode-se afirmar que o padrão é apropriado quando o consumo de energia é uma questão importante e trocas pouco frequentes de pequenos pacotes de dados é suficiente para atender aos requisitos de comunicação. É importante frisar que o TG4 teve como escopo de projeto

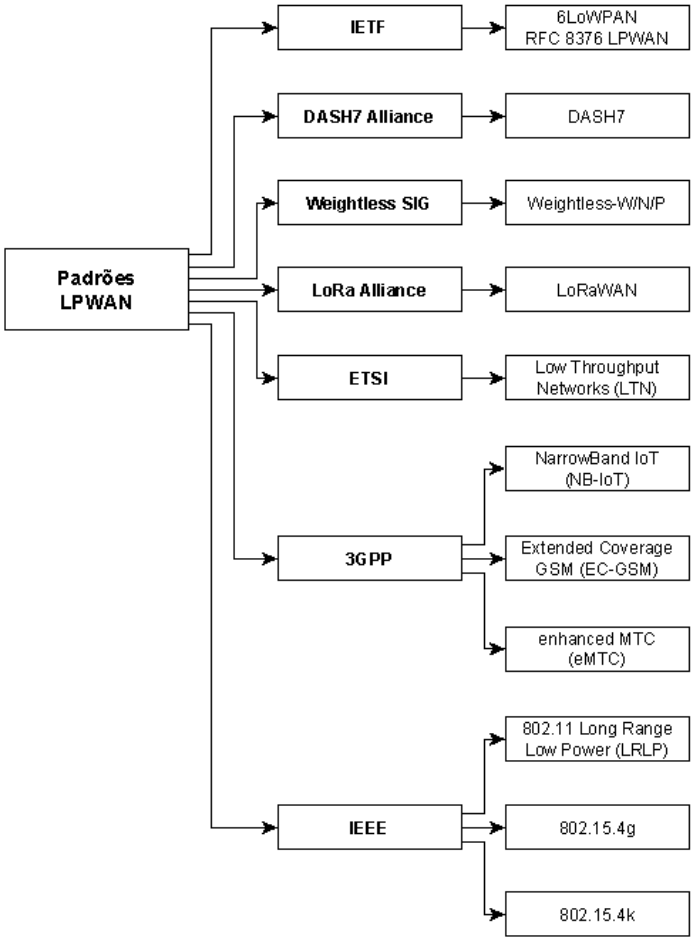


Figura 2.4 – Padrões de LPWAN. [23]

apenas a especificação das camadas físicas e de controle de acesso ao meio, assim, está fora da definição do padrão IEEE 802.15.4 quaisquer outras camadas da rede [24].

A Tabela 2.1 contempla todas as publicações relacionadas ao padrão IEEE 802.15.4. As publicações estão divididas por tipo (base, emenda ou correção) e por status (ativo, esboço ou inativo). Nos parágrafos a seguir a evolução do padrão é detalhada.

O padrão inicial, IEEE 802.15.4-2003 [25], especificou duas camadas físicas, ambas utilizando a técnica de espalhamento espectral *direct sequence spread spectrum* (DSSS). A primeira operando a 2.450 MHz, com taxa de bits de 250 kbps e a segunda utilizando as bandas de 915 MHz e 868 MHz, com taxas de bit de 40 kbps e 20 kbps, respectivamente. A camada MAC é única e lida com todos os acessos ao canal, empregando o mecanismo de acesso múltiplo com verificação de portadora e prevenção de colisão (*carrier sense multiple access with collision avoidance*, CSMA-CA), além de ainda ser responsável pelas seguintes tarefas: Gerar pacotes sinalizadores (*beacons*) para a rede se o dispositivo for um coordenador, sincronizar com os *beacons*, apoiar a associação e desassociação a uma *Personal Area Network* (PAN), suportar segurança do dispositivo, manipular e manter o mecanismo garantidor de *slots* de tempo (*guaranteed time slots*, GTS) e finalmente fornecer um canal confiável entre duas entidades MAC. Após a versão inicial, quatro novas versões base foram lançadas nos anos de 2006, 2011, 2015 e 2020.

Tabela 2.1 – Evolução do Padrão IEEE 802.15.4

Código do Padrão	Aprovação	Publicação	Status	Tipo
802.15.4-2003	12/05/2003	01/10/2003	Inativo	Base
802.15.4a-2007	22/03/2007	31/08/2007	Inativo	Emenda
802.15.4-2006	08/06/2006	07/09/2006	Inativo	Base
802.15.4c-2009	19/03/2009	17/04/2009	Inativo	Emenda
802.15.4d-2009	19/03/2009	17/04/2009	Inativo	Emenda
802.15.4-2011	16/06/2011	05/09/2011	Inativo	Base
802.15.4e-2012	06/02/2012	16/04/2012	Inativo	Emenda
802.15.4f-2012	06/02/2012	20/04/2012	Inativo	Emenda
802.15.4g-2012	29/03/2012	27/04/2012	Inativo	Emenda
802.15.4j-2013	06/02/2013	27/02/2013	Inativo	Emenda
802.15.4k-2013	14/06/2013	14/08/2013	Inativo	Emenda
802.15.4m-2014	27/03/2014	30/04/2014	Inativo	Emenda
802.15.4n-2016	29/01/2016	29/04/2016	Inativo	Emenda
802.15.4p-2014	27/03/2014	05/05/2014	Inativo	Emenda
802.15.4q-2016	29/01/2016	29/04/2016	Inativo	Emenda
802.15.4s-2018	15/02/2018	27/06/2018	Inativo	Emenda
802.15.4t-2017	14/02/2017	14/04/2017	Inativo	Emenda
802.15.4u-2016	22/09/2016	10/02/2017	Inativo	Emenda
802.15.4-2015	05/12/2015	22/04/2016	Inativo	Base
802.15.4-2015 Cor1	15/02/2018	09/07/2018	Inativo	Correção
802.15.4v-2017	18/05/2018	30/06/2017	Inativo	Emenda
802.15.4x-2019	21/03/2019	26/04/2019	Inativo	Emenda
802.15.4w-2020	04/06/2020	25/09/2020	Ativo	Emenda
P802.15.4y	14/05/2018	-	Esboço	Emenda
802.15.4z-2020	04/06/2020	25/08/2020	Ativo	Emenda
802.15.4-2020	06/05/2020	23/07/2020	Ativo	Base
802.15.4aa	-	-	Esboço	Emenda

Em 2006 [26] a nova versão base liberada após a versão de 2003, adicionou mais duas opções de camadas físicas, que produzem taxas de dados mais altas nas bandas de frequência mais baixas, assim as configurações PHY de 868/915 MHz que já suportavam taxas de dados de 20 kbps e 40 kbps, agora opcionalmente poderiam suportar, 100 kbps e 250 kbps, respectivamente. A configuração PHY de 2.450 MHz permanece suportando uma taxa de dados de 250 kbps. Assim, passaram a existir quatro camadas físicas:

- OQPSK[1]-DSSS na banda de 2.450 MHz (802.15.4-2003);
- BPSK[2]-DSSS na banda de 868/915 MHz (802.15.4-2003);
- OQPSK-DSSS na banda de 868/915 MHz (802.15.4-2006);
- BPSK-PSSS[3] e ASK na banda de 868/915 MHz (802.15.4-2006).

[1] *Offset Quadrature Phase-Shift Keying*
[2] *Binary Phase Shift Keying*
[3] *Parallel Sequence Spread Spectrum*

A camada de acesso ao meio permaneceu compatível com a versão inicial, mas adicionou novas opções e alguns aprimoramentos, incluindo: suporte a uma base de tempo compartilhada com um mecanismo de marcação de tempo; suporte a agendamento de pacotes sinalizadores *beacons*; sincronização de mensagens de *broadcast* em redes PAN habilitadas para utilização de *beacons* e aprimoramentos de segurança.

Em 2011 [27] o padrão foi mais uma vez revisado e passou a incluir três alterações aprovadas após a versão de 2006. Este esforço acrescentou mais quatro opções de camadas físicas:

- Uso de *Chirp Spread Spectrum* (CSS), empregando modulação *Differential Quadrature Phase-Shift Keying* (DQPSK), operando na banda de 2.450 MHz;
- Banda ultra-larga (*ultra-wideband*, UWB) combinando a modulação *Burst Position Modulation* (BPM) e a modulação BPSK, operando nas bandas sub-GHz ou na banda de 3 a 10 GHz;
- Modulação *M-ary Phase-Shift Keying* (MPSK), operando na banda de 780 MHz;
- Modulação *Gaussian Frequency-Shift Keying* (GFSK), operando na banda de 950 MHz.

Além de uma nova característica da camada de acesso ao meio, para estimação de distâncias entre nós. Também é preciso destacar uma mudança na organização do padrão, que passou a ser organizado de modo que cada camada física passou a ter uma seção separada, enquanto que a camada de acesso ao meio foi dividida em descrição funcional, especificação de interface e especificação de segurança.

Em 2015 [11] uma nova versão base foi criada com o intuito de reunir as alterações aprovadas após a revisão de 2011, que engloba seis alterações na camada física e uma alteração na camada de acesso ao meio, além de retificações e esclarecimentos. Os recursos adicionados pelas emendas são os seguintes:

- Formatos de quadros aprimorados, mantendo compatibilidade com versões anteriores;
- Elementos de informação (*information elements*, IE);
- Salto de canais;
- Opções estendidas de *superframes*;
- Mecanismos para reduzir consumo de energia;
- Um *frame* de reconhecimento aprimorado que pode transportar dados e ser protegido;
- Acesso priorizado a canais;
- Uma variedade de novas opções de modulação, codificação e faixas de frequência na camada física, para suportar uma ampla variedade de aplicações específicas, como:

- Identificação por radiofrequência (*radio frequency identification*, RFID);
- Redes inteligentes de utilidades (redes SUN);
- Operação na faixa branca de televisão (*television white space*, TVWS);
- Monitoramento de infraestrutura crítica com baixo consumo de energia (*low-energy critical infrastructure monitoring*, LECIM);
- Comunicações e controle ferroviários (*rail communications and control*, RCC).

Em 2020 [2] foi lançada a mais recente versão base da norma, denominada IEEE 802.15.4-2020, que tem como objetivo corrigir erros, inconsistências e ambiguidades de versões anteriores, além de incorporar sete emendas já existentes e completas (802.15.4n, 802.15.4q, 802.15.4s, 802.15.4t, 802.15.4u, 802.15.4v e 802.15.4x). Duas camadas físicas e uma nova camada MAC foram adicionadas, com retificações e esclarecimentos. Os recursos adicionados pelas alterações incluem novas opções de modulação, codificação e faixas de frequência na camada física para oferecer suporte a uma ampla variedade de aplicações e necessidades, incluindo alterações nas modulações das redes SUN, a faixa de frequência médica da China (*china medical band*, CMB) e as modulações *Ternary Amplitude Shift Keying* (TASK) e *Rate Switch - Gaussian Frequency Shift Keying* (RS-GFSK). Já um dos aprimoramentos na camada MAC é um recurso de medição da qualidade das faixas de frequências em uso, além de definição de métricas de desempenho, tais como taxa de erro de entrega de pacotes e atraso na entrega de pacotes.

Outros avanços para o padrão IEEE 802.15.4 vêm sendo desenvolvidos pelos grupos de tarefa TG4w, TG4y e TG4z, que devem implementar, respectivamente, as futuras emendas que atualmente estão num estágio inicial, ou seja, de esboço. O TG4w está desenvolvendo o 802.15.4w, que trata da definição de uma LPWAN, sendo uma extensão da camada física LECIM (já definida em emenda do padrão). O TG4y, por sua vez, é responsável pelo 802.15.4y, que definirá extensões de segurança ao padrão 802.15.4, adicionando suporte ao padrão de criptografia *Advanced Encryption Standard - 256 bits - Counter with Cipher block chaining Message authentication code* (AES-256-CCM), além de um método de registro de autenticação e a inclusão de novos algoritmos de segurança. Finalmente, o TG4z é responsável pelo desenvolvimento de novas camadas físicas que aumentem a capacidade de dispositivos que utilizam a UWB para transmissão de dados, utilizando técnicas como o *Enhanced High Rate pulse* (HRP) e o *Low Rate Pulse* (LRP). A camada de acesso ao meio também sofrerá modificações para se adequar í s mudanças na camada física propostas por esta emenda.

2.2.1 Novas definições de camada física da emenda IEEE 802.15.4g

A emenda IEEE 802.15.4g, que foi consolidada no padrão IEEE 802.15.4-2015, especifica camadas físicas alternativas, além das existentes no padrão IEEE 802.15.4-2011. Além disso, a emenda também define modificações na camada de enlace necessárias para apoiar a implementação delas. As novas camadas físicas dão suporte principalmente à

implementação de redes sem fio inteligentes de utilidades em ambientes externos para medição inteligente (redes SUN), com baixa taxa de transferência de dados, em vários domínios regulatóios.

As novas camadas físicas para redes SUN são as seguintes:

- SUN-FSK - Chaveamento por deslocamento de frequência (*frequency shift keying*, FSK);
- SUN-OFDM - Multiplexação por divisão de frequência ortogonal (*orthogonal frequency division multiplexing*, OFDM);
- SUN-O-QPSK - Chaveamento por deslocamento de fase em quadratura (*offset quadrature phase-shift keying*, O-QPSK).

As camadas físicas das redes SUN suportam múltiplas taxas de transferência de dados em bandas de frequência que variam de 169 MHz a 2450 MHz. As três novas definições de camada física para redes SUN são detalhadas a seguir.

SUN-FSK

A modulação SUN-FSK tem duas vantagens principais: a primeira é a boa eficiência energética, devido ao envelope constante do sinal e a baixa complexidade de implementação, a segunda é a compatibilidade com sistemas legados [28]. Por exemplo, a maioria dos sistemas implantados para medição inteligente nos Estados Unidos é baseada em esquemas de modulação FSK, principalmente aqueles que usam a faixa de frequência de 902 a 928 MHz [29].

O SUN-FSK pode ser usado em várias faixas de frequência, o que o torna adequado para diferentes regiões. Três modos de operação diferentes são definidos para cada faixa de frequência. O modo de operação define parâmetros de modulação e canal, como o tipo de modulação (BFSK ou 4FSK), o espaçamento do canal e o índice de modulação.

A unidade de dados de serviço na camada física (*physical service data unit*, PSDU) pode ser opcionalmente processada por um codificador de correção antecipada de erros (*forward error correction*, FEC). Dois tipos de FEC podem ser aplicados: um código recursivo e sistemático (*recursive and systematic code*, RSC) ou um código não-recursivo e não sistemático (*non-recursive and non-systematic code*, NRNSC). O FEC deve ser empregado no cabeçalho da camada física (*physical header*, PHR) e nos bits do PSDU. A intercalação deve ser empregada em conjunto com a codificação NRNSC e também pode ser empregada com a codificação RSC [11].

A unidade de dados de protocolo da camada física (*physical protocol data unit*, PPDU) do SUN-FSK é composta por um cabeçalho de sincronização (*synchronization header*, SHR), com um preâmbulo e um delimitador de início de quadro (*start of frame delimiter*, SFD), um PHR de 16 bits e uma carga útil de até 2047 bytes. No PHR, é definido o comprimento da sequência de verificação de quadros (*frame check sequence*, FCS), assim como o comprimento do quadro. Há um bit para indicar a ocorrência de uma alternância de modo, que é um

mecanismo que permite alterar a taxa de símbolos ou o esquema de modulação usado pelos rádios [28].

SUN-OQPSK

A modulação OQPSK foi introduzida na primeira versão do padrão IEEE 802.15.4, em 2003, apenas para a faixa de 2,4 GHz e fornecendo uma taxa de bits de 250 kbps. No IEEE 802.15.4g, outros modos de uso foram definidos, permitindo o uso da modulação OQPSK também em outras faixas de frequência e para taxas de bits variando de 6,25 a 500 kbps. O SUN-OQPSK emprega o DSSS, que permite uma melhor resistência í interferência. Para algumas faixas, é possível usar um modo alternativo, chamado DSSS multiplexado (*multiplexed direct sequence spread spectrum*, MDSSS) [11].

O PPDU do quadro SUN-OQPSK é composto por um SHR, um PHR e a carga útil PHY (PSDU). A taxa de símbolos é definida como a taxa de bits do SHR [11]. Quatro modos de uso, com diferentes taxas de bit, são suportados para cada faixa de frequência, mas apenas o Modo 0 é obrigatóio.

O uso do FEC é obrigatóio para o campo PHR. Ao usar DSSS como modo de espalhamento, o FEC também deve ser aplicado aos bits do PSDU, com uma codificação convolucional. Para o modo de propagação definido como MDSSS, o FEC é opcional. A intercalação deve ser empregada em conjunto com o FEC, para melhorar a robustez contra erros de intermitência e interromper a correlação de bits consecutivos [11].

SUN-OFDM

O SUN-OFDM foi definido pela primeira vez na emenda IEEE 802.15.4g e depois incorporado no IEEE 802.15.4-2015. Foi definido para fornecer altas taxas de dados e longo alcance, enquanto lida com problemas de interferência e atenuação por multipercursos [30]. Diferente do FSK e O-QPSK, o OFDM não vinha sendo amplamente utilizado para comunicações sem fio de baixa potência até recentemente, devido aos rigorosos requisitos de processamento, memóia e consumo de energia [31].

O SUN-OFDM pode ser usado em diferentes faixas de frequência (sub-GHz e 2,4 GHz) e, no padrão IEEE 802.15.4-2015, foram definidas configurações que oferecem taxas de bit entre o 50 kbps e 800 kbps. O espaçamento da subportadora é constante e igual a 10416-2/3 Hz, enquanto a taxa de símbolos é igual a 8-1/3 ksymbol/s (um período de símbolo de 120 μ s) [31]. No padrão IEEE 802.15.4-2020, foram definidas novas configurações que oferecem taxas de bit entre 12,5 kbps e 2.400 kbps [2].

Quatro opções de uso diferentes são definidas para o SUN-OFDM, cada uma com um número diferente de tons ativos. Para cada opção, um conjunto de esquemas de modulação e codificação (*modulation coding scheme*, MCS), numerados de 0 a 6, podem ser usados. Se um dispositivo suportar uma determinada opção, ele deverá suportar todas as modulações BPSK e QPSK e todos os MCS para essa opção. O suporte à modulação de amplitude em quadratura (*quadrature amplitude modulation*, QAM) é opcional [11].

O MCS determina o esquema usado para modular as subportadoras (BPSK, QPSK ou 16-QAM), se a repetição de frequência é aplicada (ou seja, 4x, 2x ou sem repetição de frequência) e a taxa de codificação FEC (1/2 ou 3/4). Assim, a taxa de dados efetiva depende do MCS usado. Por exemplo, para a Opção 2, que usa 52 tons ativos, a taxa de dados efetiva pode variar de 50 kbps, ao usar BPSK, uma taxa de codificação de 1/2 e repetição de frequência 4x, a 1.200 kbps, ao usar o 16-QAM, com uma taxa de codificação de 1/2 e sem repetição de frequência.

O PPDU do SUN-OFDM é composto por um SHR, um PHR e a carga útil. O SHR possui um preâmbulo e o PHR contém as informações sobre a configuração usada para transmitir a carga útil. O SHR e o PHR são transmitidos usando o MCS mais baixo suportado para a opção que está sendo usada. O comprimento máximo da carga útil é de 2047 bytes, o que permite transmitir um pacote IPv6 completo sem fragmentação [31].

2.2.2 Novas definições de camada de enlace da emenda IEEE 802.15.4e

Em 2008, o IEEE criou o grupo de trabalho 802.15 TG4e com o objetivo de redesenhar a camada de enlace do protocolo IEEE 802.15.4 existente, a fim de superar suas limitações. O objetivo era definir uma camada de enlace de baixa potência com saltos múltiplos, capaz de atender às necessidades emergentes de aplicações industriais. O resultado final foi o IEEE 802.15.4e, aprovado em 2012, que é baseado em muitas ideias dos padrões existentes para aplicações industriais (ou seja, WirelessHART e ISA100.11a), incluindo janelas de acesso (compartilhadas ou dedicadas), comunicação multicanal e salto em frequência. Especificamente, o IEEE 802.15.4e estende o padrão 802.15.4 anterior, introduzindo novos modos de comportamento na camada de enlace, projetados para suportar domínios de aplicações específicos, além de aprimoramentos funcionais gerais. A seguir é apresentada uma descrição mais detalhada sobre os aprimoramentos funcionais gerais e os modos de comportamento da camada de enlace, com base no trabalho de Guglielmo et.al [32].

Aprimoramentos funcionais gerais

O padrão IEEE 802.15.4e define seis aprimoramentos funcionais gerais, que são listados a seguir.

- Modo de baixa energia (*low energy*, LE): mecanismo destinado a aplicações que podem trocar latência por eficiência energética. Permite que um nó opere em um ciclo de trabalho muito baixo (por exemplo, 1% ou menos), enquanto parece estar sempre ativo para as camadas superiores. Esse mecanismo é importante para o paradigma da Internet das Coisas, em que os protocolos da Internet foram projetados assumindo que os nós estão sempre ativos;
- Elementos de informação (*information elements*, IE): mecanismo extensível para trocar informações em uma subcamada de enlace;

- Pacotes sinalizadores aprimorados (*enhanced beacons*, EB): extensão dos quadros de *beacons* do padrão IEEE 802.15.4 que fornecem uma maior flexibilidade. Eles permitem criar quadros específicos de aplicações, incluindo elementos de informação relevantes;

- Quadros multi-uso (*multipurpose frame*, MF): mecanismo que fornece um quadro com formato flexível que pode endereçar várias operações da camada de enlace baseado em elementos de informação;

- Métrica de desempenho da camada de enlace (*medium access control performance metric*, MAC-PM): mecanismo para fornecer *feedback* apropriado sobre a qualidade do canal para a rede e camadas superiores, buscando tomar decisões mais apropriadas. Por exemplo, o protocolo IP pode implementar fragmentação dinâmica de datagramas, dependendo das condições do canal;

- Associação Rápida (*fast association*, FastA): procedimento que permite que um nó se associe em um tempo reduzido para aplicações críticas, em que a latência tem prioridade sobre a eficiência energética.

Modos de comportamento da camada de enlace

O padrão IEEE 802.15.4e define cinco novos modos de comportamento MAC, que são listados a seguir.

- Salto de canal sincronizado com o tempo (*time slotted channel hopping*, TSCH): destina-se a domínios de aplicação como automação industrial e controle de processos. Esse modo fornece suporte para comunicações utilizando múltiplos canais e acesso múltiplo por divisão de tempo (*time division multiple access*, TDMA);

- Extensão multicanal determinística e síncrona (*deterministic and synchronous multi-channel extension*, DSME): destina-se a aplicações industriais e comerciais com requisitos rigorosos em termos de latência e confiabilidade. Para esse fim, combina acesso ao meio baseado em contenção e divisão de tempo e oferece dois modos diferentes de diversidade de canais (adaptação de canal e o salto de canal). Ele é projetado especificamente para redes em malha com múltiplos saltos;

- Rede determinística de baixa latência (*low latency deterministic network*, LLDN): projetado para redes de salto único e canal único, destina-se í automação de fábricas e indústrias, onde as aplicações exigem latência muito baixa ($\leq$ 10 ms);

- Adaptação multicanal assíncrona (*asynchronous multi-channel adaptation*, AMCA). É focada em aplicações em que grandes implantações são necessárias, como redes de utilidades inteligentes, redes de monitoramento de infraestrutura e redes de controle de processos. Nessas redes, utilizar um canal comum e único de comunicação pode não permitir a conexão de todos os nós na mesma rede PAN. Além disso, a variação da qualidade do canal é geralmente grande e a assimetria de enlace pode ocorrer entre

dois nós vizinhos. O modo AMCA depende de adaptação multicanal assíncrona e pode ser usado somente em PANs não habilitadas para uso de *beacons*. Em uma rede AMCA, cada dispositivo seleciona o canal com a melhor qualidade como canal de escuta e começa a escutar nessa frequência. No momento em que dois nós precisam trocar pacotes, o dispositivo emissor muda para o canal de escuta designado do dispositivo receptor.

- Sinalizador de identificação por radiofrequência (*radio frequency identification blink*, BLINK). Destina-se a aplicações para identificação de itens ou pessoas, localização e rastreamento. Especificamente, ele permite que um nó comunique sua identificação (ID) para outros nós, sem associação prévia e sem reconhecimento. Pacotes BLINK geralmente são enviados para os dispositivos, usando o protocolo *ALOHA*.

2.3 Resultados experimentais da implantação de redes IEEE 802.15.4g

Em [33] é descrita uma solução geral adotando o padrão de especificação de mensagem de linha de distribuição / especificação complementar para medição de energia (*distribution line message specification / companion specification for energy metering*, DLMS/COSEM) para desenvolvimento de aplicações de medição para rede elétrica inteligente (*smart grid*, SG). O transceptor utilizado na solução é o Microchip AT86RF215, compatível com os padrões IEEE 802.15.4-2011 e IEEE 802.15.4g-2012, operando em bandas de frequência sub-GHz e 2,4 GHz, com uma potência máxima de transmissão de 14 dBm e sensibilidade de até -123 dBm. Foram utilizadas antenas omnidirecionais com ganho de 2 dBi para a banda de 915 MHz. Os transceptores foram configurados para operar no canal 7 da banda de 915 MHz com o esquema de modulação O-QPSK e com taxa de transmissão da dados de 250 kbps. A potência de transmissão foi definida em 14 dBm e a sensibilidade do RF215, para este esquema de modulação, é de -102 dBm.

Os testes experimentais foram conduzidos durante 20 dias, observando-se que a taxa de perda de pacote aumenta drasticamente de acordo com o aumento do número de saltos entre os nós, variando entre um valor baixo, próximo de 0%, quando a transmissão utiliza apenas 1 salto, até um valor alto, em torno de 90%, quando a transmissão realiza mais de 4 saltos para atingir o nó destino. Mesmo nesse último caso, com uma alta taxa de perda de pacote, a taxa média de falhas das sessões DLMS foi de apenas 0,25% (utilizando o protocolo TCP), e 0,47% (utilizando o protocolo UDP). Também foi observada a influência dos materiais armazenados no armazém no tempo total de transmissão (*round-trip time*, RTT), durante o período de teste. De maneira geral, o RTT foi baixo, e aumentou em torno de 15 ms a cada novo salto.

Em [9] [4] foram realizados experimentos utilizando nós que consistiam em um conjunto

[4] Os autores disponibilizaram o código fonte utilizado no experimento em `https://github.com/`

equipado com um Raspberry Pi 3 (com uma distribuição Linux Debian instalada), uma placa de rádio ATREB215-XPRO-A, que possui um chip de rádio AT86RF215 e é equipada com dois conectores coaxiais de rádio frequência de semi-precisão SMA (*subminiature version A*) conectados a antenas com ganho de 2 dBi. Também possui um módulo GPS (*Global Positioning System*) da Adafruit (Ultimate GPS), construído com chip MTK3339 e com um conector uFL externo para conexão de uma antena GPS. Por meio de um barramento de interface periférica serial (*serial peripheral interface*, SPI), o Raspberry Pi pode acionar a placa de rádio, modificando suas configurações a cada teste a ser realizado e também pode receber o horário de *greenwich* (*greenwich mean time*, GMT) e a posição geográfica, através do módulo GPS.

Foi possível observar que os enlaces de rádio com maior alcance foram obtidos utilizando as modulações FSK-FEC e O-QPSK. Esse comportamento era esperado, pois essas configurações de camada física possuem a maior sensibilidade entre todas as disponíveis, fornecendo até 141 dB de *link budget*.

Já as configurações que usam OFDM são mais robustas contra interferências e efeitos de atenuação por multipercursos. O OFDM1-100 MCS0 usa 104 sub-portadoras com 4 repetições de frequência e oferece taxa de bits de 100 kbps, proporcionando um alto nível de robustez. O OFDM2-50 MCS0 tem a mesma característica de 4 repetições de frequência, mas utiliza apenas metade das subportadoras, oferecendo consequentemente a metade da taxa de bits (50 kbps). Cada canal OFDM Opção 1 ocupa 1,2 MHz e as 4 subportadoras iguais são separadas pela mesma distância em frequência. Mesmo se 3 subportadoras forem comprometidas durante a transmissão de um símbolo no caminho, o receptor ainda pode recuperar as informações. Apesar dessa robustez, a taxa de entrega de pacote (*packet delivery ratio*, PDR) para esta configuração foi de 31% para pacotes curtos e de apenas 5% para pacotes longos, considerando uma distância de 700 m. Para o OFDM2-100, o PDR aumentou para 92% e 47%, respectivamente. Essa configuração de camada física possui sensibilidade 2 dB maior em relação ao OFDM1-100.

A razão pela qual as configurações que usam OFDM não conseguem comunicações a longa distâncias é que elas apresentam uma menor sensibilidade e uma potência de transmissão menor (11 dBm para o OFDM e 14 dBm para FSK e O-QPSK). Esses valores de sensibilidade e potência máxima de transmissão são diretamente dependentes do hardware e pode variar dependendo do fabricante.

Em [34], inicialmente uma visão geral dos diferentes protocolos para redes LPWAN foi elaborada, incluindo uma discussão sobre o uso dessas tecnologias para aplicações de monitoramento em ambientes industriais. Na segunda parte do estudo, foram apresentados resultados de um experimento para avaliação de uma aplicação de monitoramento industrial com duração de várias semanas, com foco em três tecnologias LPWAN: LoRa©, IEEE 802.15.4g e um esquema de comunicação de banda estreita proprietário (modo de longo alcance implementado no transceptor TI CC1310). A avaliação mostrou que não há um

`openwsn-berkeley/range_test`

protocolo único que supere todos os outros protocolos em termos de todas as métricas de avaliação (confiabilidade, alcance, energia) consideradas no estudo.

LoRa© fornece o maior alcance de comunicação de todos os três protocolos. No entanto, o consumo de energia de LoRa© é uma ordem de magnitude maior que a do padrão IEEE 802.15.4g (com modulação 2-FSK), que fornece um *link budget* menor, ou seja, o gasto final de potência de enlace necessário para envio de um pacote de rádio é 80 vezes menor, quando utilizado o protocolo IEEE 802.15.4g em comparação com os outros protocolos do teste. Assim para a escolha de um protocolo LPWAN adequado é necessário analisar quais os requisitos concorrentes da aplicação (alcance de comunicação, consumo de energia, capacidade de rede).

Nos experimentos descritos por em [35], foram utilizados nós sensores da plataforma Zolertia, equipados com antenas omnidirecionais com ganho de 2 dBi. Os nós sensores são baseados no microcontrolador ARM Cortex-M3 (*system on chip*, SoC), e executam o sistema operacional Contiki. Para a comunicação sem fio, foi utilizada uma interface de rádio baseada no transceptor CC1120, da Texas Instruments, configurada para operar na banda de 863-950 MHz. Durante os experimentos, foi selecionada uma configuração do padrão IEEE 802.15.4g, mais especificamente a que opera na banda de 868 MHz, com modulação 2-GFSK e possui 33 canais disponíveis. Nesta configuração, a taxa de bits é de 50 kbps. A potência de transmissão selecionada foi o valor padrão definido no driver do CC1200 no Contiki, que é a potência máxima de transmissão permitida (14 dBm).

As medições mostraram que o alcance da comunicação depende fortemente dos ambientes onde os nós estão instalados. Por exemplo, a presença de edifícios, árvores e outros objetos podem atenuar significativamente ou impedir a propagação do sinal. Especificamente, descobriram que o alcance de comunicação é de aproximadamente 800 m em uma área rural, 250 m em um ambiente semi-rural, e cerca de 150 m em um ambiente urbano. Embora esses resultados dependam das condições específicas dos experimentos e não permitem obter conclusões mais gerais, os valores medidos mostram um limitado alcance da comunicação, especialmente no meio urbano. Mesmo no meio rural, o alcance da comunicação foi inferior a 1 km. Portanto, quando as distâncias são maiores do que os intervalos medidos, a comunicação direta não é suficiente e redes com múltiplo saltos podem ser necessárias.

Em [31] é descrita uma avaliação experimental pra comparar a robustez contra interferência das camadas físicas OQPSK-DSSS (IEEE 802.15.4-2003) e SUN-OFDM (IEEE 802.15.4-2015). Condições de interferência foram geradas em uma configuração controlada para avaliar a relação sinal-ruído (*signal-to-interference ratio*, SIR), exigida por cada modulação, de modo a atingir um PDR $>$ 80%.

Foram utilizados três dispositivos OpenMote-B, um atuando como transmissor, outro como receptor e o terceiro como interferente. O OpenMote-B é equipado com um chip Texas Instruments CC2538 (SoC) e um transceptor de rádio Microchip AT86RF215. Para coordenar as ações entre os diferentes dispositivos, um computador executando o Ubuntu 18.04 LTS

foi conectado ás placas OpenMote-B via USB e executou um script Python que orquestrou todo o experimento.

Os resultados mostram que o SUN-OFDM fornece pelo menos 6 dB de proteção adicional, em relação ao OQPSK-DSSS, independentemente do tipo de interferência e do comprimento do pacote. Além disso, o SUN-OFDM ocupa apenas 1,2 MHz de largura de banda, enquanto o OQPSK-DSSS ocupa 5 MHz.

2.4 Novos desafios para a pesquisa em redes IEEE 802.15.4g

Esquemas de diversidade são amplamente utilizados em redes sem fio para aumentar a confiabilidade de comunicação e para lidar com as variações temporais e espaciais na qualidade dos enlaces. Diferentes estratégias podem ser adotadas em diferentes camadas. Na camada física, diversidade de código e de antena são mecanismos bem estabelecidos, mas não são amplamente empregados em redes sem fio de baixa potência. Na camada de enlace, replicação de pacote no tempo e em diferentes canais são esquemas empregados em diversos padrões, como em alguns modos do IEEE 802.15.4e. Devido í possibilidade de usar vários tipos de modulação com um único transceptor, considerando as novas definições da emenda IEEE 802.15.4g, a diversidade de modulação em camada de enlace, para transmissão de pacotes consecutivos com esquemas de modulação diferentes, pode ser outra técnica empregada em redes IEEE 802.15.4 para aumento de confiabilidade.

Esta seção apresenta um conjunto de artigos recentes que visam propor técnicas para aumento de confiabilidade por meio de técnicas de diversidade em redes que usam as definições do IEEE 802.15.4g e, consequentemente, compatíveis com o padrão IEEE 802.15.4-2020.

2.4.1 Seleção dinâmica de canais

A capacidade dos nós sensores sem fio de alternar dinamicamente sua frequência operacional oferece a oportunidade de aumentar a confiabilidade dessas redes. A alocação dinâmica de canais (*dynamic channel allocation*, DCA) ou seleção dinâmica de canais foi desenvolvida para equilibrar recursos espectrais entre vários nós da rede sem fio, buscando atingir a máxima eficiência espectral do sistema. A pesquisa em DCA tem sido ativa para sistemas celulares, redes locais sem fio (*wireless local area network*, WLAN) e mais recentemente para as redes de sensores sem fio [36].

Como exemplos de trabalhos recentes que exploraram seleção dinâmica de canais em redes LPWAN, [37] e [38] descrevem experimentos que foram conduzidos com dispositivos para IoT, com o objetivo principal de avaliar um algoritmo proposto em trabalhos anteriores dos autores, que utilizam aprendizado de máquina na busca de solução de problemas de coexistência de vários sistemas de IoT, tais como sistemas baseados em redes Wi-

SUN, Sigfox© e LoRaWAN©, se comunicando nas mesmas faixas de frequência. Tais trabalhos [39][40], modelam o problema de seleção de canais distribuídos em redes de sensores sem fio como um problema *Multi-Armed Bandit*, MAB). Os canais que os nós poderiam utilizar foram considerados como as máquinas caça-niqueis do problema MAB, assim como os quadros de confirmação (*acknowledgement*, ACK) do *gateway*, quando a transmissão foi bem sucedida, foram considerados como recompensa para o MAB. Para a solução deste problema se adotou a dinâmica do cabo-de-guerra (*tug-of-war*, TOW), que é um algoritmo de reforço e que utiliza poucos cálculos, o que justifica a possibilidade de sua implementação em dispositivos IoT.

O desempenho da solução proposta foi avaliado usando um protótipo de IoT cognitiva instalado em um ambiente em conjunto com vários outros sistemas IoT (Wi-SUN, SigFox© e LoRaWAN©), que coexistiram e competiram pelas mesmas bandas de frequência. Nos dois experimentos descritos nos artigos, foram utilizados 5 Gateways Wi-SUN, 30 nós protótipos de IoT cognitiva (802.15.4g), 10 nós Sigfox© e 4 nós LoRaWAN©. Os canais 25, 28, 31, 34, 37 foram atribuídos respectivamente a cada um dos 5 *gateways*.

Os pesquisadores concluíram que o desempenho da rede, medido por métricas definidas no artigo, taxa de sucesso de transmissão de quadros (*frame success rate*, FSR) e o índice de justiça (*fairness index,* FI), melhorou quando o algoritmo de seleção de canais proposto foi utilizado.

2.4.2 Seleção dinâmica de modulação

A seleção dinâmica de modulação, também conhecida como modulação adaptativa, é uma técnica que modifica o esquema de modulação com base nas características do canal de transmissão ou do requisito de taxa de transferência da aplicação. Esta técnica tem um benefício significativo de manter o rendimento (medido em termos de probabilidade de erro) da rede em um nível adequado. Outro fator de grande importância no uso da modulação adaptativa é a eficiência na estimação da qualidade do canal. Por fim, modulação adaptativa pode ser utilizada para otimizar o consumo de energia, estendendo a vida útil das baterias dos nós, tornando mais eficiente e aumentando o tempo de vida de funcionamento da rede de sensores sem fio [41].

Em [42] buscou-se responder à pergunta: "É possível melhorar a taxa de entrega de pacote combinando várias modulações do padrão IEEE 802.15.4-2015?". Desta forma, o conceito de diversidade de modulação pacote a pacote na camada de enlace foi considerado, usando as múltiplas modulações do padrão 802.15.4g SUN para transmissão de pacotes consecutivos, de modo a melhorar a confiabilidade. Os resultados foram obtidos através de uma instalação experimental real, em que verificou-se que replicando transmissões de pacotes (entre uma e três replicações) com a mesma modulação SUN pode-se aumentar o PDR de 85,0% / 84,6% / 71,3% para 94,2% / 94,1% / 86,0% usando SUN-FSK, SUN-OQPSK e SUN-OFDM, respectivamente. Combinando o mesmo número de transmissões de pacotes com diversidade de modulação pode-se aumentar ainda mais o PDR médio, para

97,1%, indicando seu potencial de utilização como uma ferramenta para ajudar a atender aos requisitos de confiabilidade das aplicações industriais.

Em [43] é descrito o conceito de diversidade de modulação adaptativa em camada de enlace, que permite selecionar dinamicamente modulações diferentes, para melhorar a confiabilidade do enlace. Os autores apresentaram três estratégias para seleção adaptativa de modulação validaram-nas usando os dados derivados de uma instalação experimental real que utilizou as três modulações IEEE 802.15.4g SUN (ou seja, SUN-FSK, SUN-OQPSK e SUN-ODFM) em um ambiente industrial. Os resultados mostraram que, usando as estratégias de diversidade de modulação adaptativa, é possível melhorar a confiabilidade do enlace, independentemente das condições dos nós considerados no experimento. De acordo com os resultados da simulação baseada em *trace*, realizada com base em um conjunto de dados obtidos em um experimento real, as estratégias de diversidade de modulação propostas (utilizando uma modulação, duas modulações ou todas as três modulações definidas no padrão IEEE 802.15.4g a cada momento) permitiram aumentar o PDR em relação ao uso de uma única modulação e também em relação ao uso de técnicas de diversidade de modulação cegas, em que as modulações são escolhidas aleatoriamente ou de forma sequencial.

Finalmente, é importante destacar o trabalho descrito em [44] introduziu os conceitos da técnica de diversidade de modulação em redes IEEE 802.15.4g e também serviu como base para os dois artigos anteriormente citados, uma vez que eles utilizam os dados coletados no experimento real descrito nesse artigo. O experimento considerou uma rede com 11 nós finais, que utilizaram os três diferentes esquemas de modulação SUN definidos no padrão IEEE 802.15.4g. Os nós foram implantados em um armazém de aproximadamente 110 m^2, durante 99 dias, e o conjunto de dados resultante contém um total de 10.710.868 medições de indicador de intensidade do sinal recebido (*received signal strength indication*, RSSI), de valores de *clear channel assessment*, CCA) e PDR. Os resultados analisados mostram uma alta variabilidade no RSSI médio (ou seja, entre -82,1 dBm e -101,7 dBm) e CCA (ou seja, entre -111,2 dBm e -119,9 dBm), causada pelos efeitos da propagação de caminhos múltiplos e interferência externa. Apesar de estar acima do limite de sensibilidade para cada modulação, esses valores resultam em valores médios baixos de PDR (ou seja, entre 65,9% e 87,4%), indicando que são necessários esquemas adicionais na comunicações em redes sem fio de baixa potência para atender aos requisitos de confiabilidade de aplicações industriais. Para tal propósito, foi introduzido o conceito de diversidade de modulação, que pode ser combinada com a repetição de pacotes para atender a esses requisitos (ou seja, PDR $>$ 99%), minimizando o gasto de energia dos nós e atendendo í s restrições regulatóias.

2.4.3 Técnicas para mitigação de interferências em redes IEEE 802.15.4g

A interferência é um fenômeno fundamental em redes de comunicação sem fio. É resultado da superposição de transmissões e da própria natureza da transmissão sem fio, em um espectro compartilhado entre vários equipamentos. Como a interferência reduz o

rendimento da rede sem fio, é essencial entender e gerenciar as interferências para obter um melhor nível de confiabilidade na rede [45]. Recursos, tais como tempo, frequência e código devem ser compartilhados por vários usuários em redes sem fio. Em sistemas de comunicações convencionais, o objetivo dos projetos é permitir que os diversos equipamentos compartilhem o meio sem fio com o mínimo ou nenhuma interferência. Diversos esforços são realizados para evitar, mitigar ou cancelar as interferências. Por exemplo, para suportar múltiplos usuários, métodos de acesso ortogonal em tempo, frequência, código, bem como domínios espaciais têm sido usados em diferentes gerações de sistemas celulares. Assim, uma série de novos desafios surgem para aplicação de tais técnicas de mitigação de interferência em redes com um grande número de nós com baixa potência, ou seja, as redes de sensores sem fio [46].

Em [47] foi investigada a influência da interferência de redes LoRa© nas redes IEEE 802.15.4g (Wi-SUN). Em particular, foi observado que o mecanismo de CCA do padrão IEEE 802.15.4 não detecta de forma confiável as transmissões LoRa© interferentes. Por isso, foi proposto um mecanismo aprimorado de CCA baseado em uma rede neural Perceptron multicamadas e demonstrado que ele reduz significativamente o número de transmissões sem êxito, mantendo-se compatível com o padrão IEEE 802.15.4. As principais contribuições do trabalho foram:

- Um conjunto de dados exclusivo de mais de 640.000 interações LoRa© coletadas em um banco de testes automatizado sob condições de interferência controlada;
- Evidências de que o CCA do IEEE 802.15.4 não consegue detectar com segurança as interferências ocasionadas pelos dispositivos LoRa©. No conjunto de dados gerado, a decisão da CCA foi correta em apenas 43% dos casos;
- Um mecanismo CCA aprimorado, baseado em uma rede neural perceptron multicamadas. O CCA aprimorado aumentou a precisão da decisão do CCA em 30 pontos percentuais e reduziu o número de falhas de transmissões IEEE 802.15.4g, por um fator maior do que três;
- Compatibilidade de tempo e interoperabilidade com os mecanismos existentes de CCA e CSMA no padrão IEEE 802.15.4. Além disso, o CCA aprimorado não depende de amostragem estendida do canal, evitando um consumo maior de energia.

2.4.4 Outras técnicas para camada de enlace ou superiores em redes IEEE 802.15.4g

Em [48] um protocolo de camada de enlace do tipo escuta por amostragem coordenada *(coordinated sampled listening*, CSL) é analisado tanto teoricamente como experimentalmente. O CSL é um modo de controle de acesso ao meio, com baixo consumo de energia, onde os dispositivos receptores monitoram periodicamente os canais para verificar transmissões recebidas em ciclos curtos de trabalho (*low duty cycles*, LDC). [12]

Primeiramente, os autores projetaram e implementaram um protocolo CSL inicial. Em seguida, após a análise teóica que avaliou as características de comunicação do CSL, o protocolo foi aprimorado considerando o efeito do ACK, baseado em um trabalho anterior do grupo [49], realizado em uma rede com topologia estrela. Para validar a análise teóica foram realizadas simulações e os resultados foram comparados com os resultados teóicos.

Em um momento posterior, o protocolo CSL foi implementado em dispositivos Wi-SUN compatíveis com o padrão IEEE 802.15.4g, equipados com módulos de rádio BP35C0 (ROHM Co., Ltd.) que segue o padrão STD-T108 da associação das indústrias e empresas de rádio (*association of radio industries and businesses*, ARIB) [50]. Além disso, o protocolo CSL foi testado experimentalmente usando 30 esses módulos Wi-SUN em uma câmara de rádio anecóica. Os resultados experimentais e teóicos indicam que o protocolo CSL pode alcançar alta taxa de sucesso de comunicação, em torno de 90%, quando a taxa de geração de dados é inferior a $5,0 \times 10^{-3} s^{-1}$, mesmo no pior dos casos, quando todos terminais não podem se detectar.

As redes baseadas no protocolo TSCH são construídas sobre uma camada física e são, portanto, limitadas pelas características da camada física escolhida em termos de taxa de transferência de dados, confiabilidade e eficiência energética. Tradicionalmente, o TSCH usa *slots* de duração fixa, grandes o suficiente para enviar um pacote de qualquer tamanho permitido, de acordo com a taxa de transferência de dados. Para combater essas limitações e melhorar o desempenho e flexibilidade da rede em desafiadores ambientes industriais, o trabalho descrito em [51] explora a utilização de vários esquemas de modulação e codificação (MSC), além de uma inovação relacionada aos *slots* de comunicação em uma rede TSCH.

Para evitar desperdício de tempo de comunicação, quando são utilizados simultaneamente vários MCS, com diferentes taxas de transferência de dados, foi introduzido o conceito de união de *slots*, que permite a criação de *slots* de diferentes tamanhos, com duração adaptada ás taxas de transferência de dados de cada MCS escolhido. Posteriormente, foi utilizado um modelo de programação linear inteira mista, e descrito formalmente o conceito de união de *slots* TSCH, buscando maximizar a taxa de entrega de pacotes na rede enquanto o consumo de energia é minimizado, limitando o tempo em que os rádios permanecem ativos. Tal modelo foi utilizado nas simulações TSCH visando otimizar as alocações de *slots* e chegaram até uma configuração ideal, usando a abordagem de união de *slots*. Além disso, os resultados confirmam a vantagem de escalabilidade da abordagem de união de *slots*, em termos de taxa de entrega de pacotes, e também em relação ao consumo de energia.

2.5 Implementação de redes IEEE 802.15.4g usando a plataforma OpenMote-B

Esta seção descreve como implementar uma rede IEEE 802.15.4g simples utilizando a plataforma OpenMote-B [13, 52], que é mostrada na Figura 2.5.

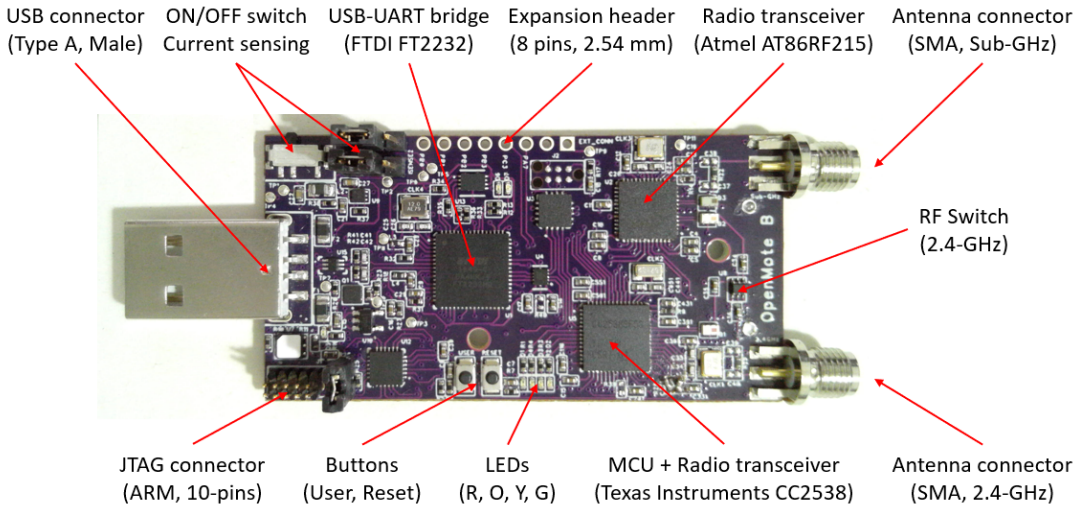


Figura 2.5 – Placa do OpenMote-B [44].

A placa do OpenMote-B possui dimensões 64 × 33 mm e é equipada com o SoC (*System on Chip*) CC2538 da Texas Instruments e um transceptor Microchip AT86RF215. O CC2538 [53] inclui um microcontrolador ARM Cortex-M3 (32 MHz, 32 kB RAM, 512 kB Flash) e um transceptor integrado compatível com o padrão IEEE 802.15.4-2006. O AT86RF215 [54] é um transceptor *dual-band* (sub-GHz e 2,4 GHz), compatível com o padrão IEEE 802.15.4-2015, que suporta as diversas opções de camada física definidas na emenda IEEE 802.15.4g-2012 (i.e., SUN-FSK, SUN-OQPSK, and SUN-OFDM). Por meio dessa plataforma e da base de código disponibilizada é possível implementar redes que usam as diferentes modulações definidas no IEEE 802.15.4g, bem como avaliar experimentalmente novas técnicas de diversidade propostas para esse tipo de rede.

Este tutorial apresenta uma aplicação simples, que pode ser usada como ponto de partida para a implementação de redes de sensores ou novos mecanismos e protocolos com base no padrão IEEE 802.15.4g. A aplicação descrita tem como objetivo realizar a comunicação entre dois dispositivos OpenMote-B. O primeiro realiza a função de enviar pacotes usando uma determinada modulação, frequência e largura de banda, que pode ser configurada no código, utilizando o transceptor AT86RF215. Pode-se ainda escolher se a comunicação ocorrerá na banda Sub-GHz ou na faixa de 2,4 GHz. O segundo dispositivo OpenMote-B recebe os pacotes e envia as informações relacionadas a eles pela porta USB.

No código de exemplo disponível no repositório, o rádio é configurado para se comunicar na banda Sub-GHz utilizando a modulação OFDM, com taxa de bits de 50 kbps e ocupando uma largura de banda de 800 kHz. No entanto, outras configurações de camada física podem ser usadas realizando poucas modificações no código.

2.5.1 Instalação dos pacotes de software requeridos

O primeiro passo para a utilização do OpenMote-B e do código de exemplo é instalar o conjunto de ferramentas necessárias para compilar código C/C++ para o microcontrolador ARM Cortex-M3, que é a arquitetura do SoC CC2538 presente na plataforma. Para isso,

é necessário instalar os pacotes *gcc-arm-none-eabi* e *binutils-arm-none-eabi*, por meio do seguinte comando no terminal[5]:

```
$ sudo apt install gcc-arm-none-eabi binutils-arm-none-eabi
```

Após a instalação desses pacotes, pode-se prosseguir com a instalação dos programas auxiliares e módulos Python:

```
$ sudo apt install build-essential git scons python3-pip
$ sudo pip install intelhex pyserial
```

2.5.2 Clonagem e configuração do repositório

Após a conclusão de todas as instalações, pode-se realizar a programação, compilação e gravação de programas para o OpenMote-B. Para isso, deve-se clonar o repositório que contém os projetos de exemplo para o OpenMote-B e submódulos necessários. A seguir é mostrada a linha de comando necessária para clonar o código do repositório[6], bem como as informações que são colocadas no terminal, indicando que o procedimento ocorreu com sucesso.

```
$ git clone https://github.com/GComPI-IFPB/openmote-sbrt.git
Cloning into 'openmote-sbrt'...
remote: Enumerating objects: 365, done.
remote: Counting objects: 100% (365/365), done.
remote: Compressing objects: 100% (238/238), done.
remote: Total 365 (delta 114), reused 365 (delta 114), pack-reused 0
Receiving objects: 100% (365/365), 628.78 KiB | 1.34 MiB/s, done.
Resolving deltas: 100% (114/114), done.
```

Após finalizar a clonagem do repositório, deve-se entrar na pasta que foi criada (denominada "openmote-sbrt") e digitar o comando *git submodule update –init –recursive*, como mostrado a seguir:

```
$ cd openmote-sbrt
$ git submodule update --init --recursive
Submodule 'tools/cc2538-bsl' (https://github.com/JelmerT/cc2538-bsl) registered for path 'tools/cc2538
-bsl'
Cloning into '.../openmote-sbrt/tools/cc2538-bsl'...
Submodule path 'tools/cc2538-bsl': checked out '...'
```

2.5.3 Escrita e compilação do código do transmissor

Seguindo o caminho "openmote-sbrt/projects/at86rf215/at86rf215-sensors-tx/src/" é possível encontrar o código fonte *main.cpp*, que utiliza o transceptor AT86RF215 para

[5]Os passos descritos neste tutorial foram testados utilizando o sistema operacional Ubuntu 20.04 LTS.

[6]O repositório está disponível em `https://github.com/GComPI-IFPB/openmote-sbrt.git`.

realizar a comunicação sem fio. No arquivo *main.cpp* encontra-se o código responsável por configurar o tipo de modulação, largura de banda e frequência de comunicação que serão utilizados pelo transceptor. Essas configurações devem ser escolhidas a partir de um conjunto de valores pré-definidos por meio de enumerações, uma para cada configuração de camada física suportada pelo transceptor. O trecho de código a seguir mostra como realizar as definições dos parâmetros do rádio:

```
#define RADIO_CORE ( At86rf215::CORE_RF09 )
#define RADIO_SETTINGS ( &radio_settings[CONFIG_OFDM2_MCS0] )
#define RADIO_FREQUENCY ( &frequency_settings_09[FREQUENCY_09_OFDM2] )
#define RADIO_CHANNEL ( 0 )
#define RADIO_TX_POWER ( At86rf215::TransmitPower::TX_POWER_MAX )
```

No diretório "openmote-sbrt/drivers/at86rf215/" pode-se encontrar os arquivos *At86rf215_conf.cpp* e *At86rf215_conf.h*, que contêm as declarações das enumerações, que podem ser usadas para configurar o rádio para usar uma das opções de camada física disponíveis.

Para garantir que o rádio funcionará corretamente é necessário manter as funções de inicialização, configuração e ativação do rádio no corpo do código fonte na seguinte ordem:

```
// Set radio callbacks and enable interrupts
at86rf215.setTxCallbacks(RADIO_CORE, &radio_tx_init_cb, &radio_tx_done_cb);
at86rf215.enableInterrupts();

// Turn AT86RF215 radio on
at86rf215.on();

// Wake up and configure radio
at86rf215.wakeup(RADIO_CORE);
at86rf215.configure(RADIO_CORE, RADIO_SETTINGS, RADIO_FREQUENCY, RADIO_CHANNEL);
at86rf215.setTransmitPower(RADIO_CORE, RADIO_TX_POWER);

// Prepare radio packet
tx_buffer_len = prepare_packet(radio_buffer, eui48_address, packet_counter, sensor_data);

// Load packet to radio
at86rf215.loadPacket(RADIO_CORE, radio_buffer, tx_buffer_len);

// Transmit packet
at86rf215.transmit(RADIO_CORE);

// Turn AT86RF215 radio off
at86rf215.off();
```

A função *prepare_packet* deve ser escrita pelo usuário. Nos códigos existentes no repositório, é possível observar melhor o uso dos métodos do rádio. Um exemplo de implementação da função *prepare_packet* também é fornecido.

Para compilar o projeto é necessário voltar à pasta raiz do repositório (denominada "openmote-sbrt"), usando o terminal, e executar o comando da ferramenta SCons, como demonstrado a seguir:

```
$ scons board=openmote—b project=at86rf215—sensors—tx compiler=gcc verbose=0
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
.
.
.
scons: done building targets.
```

Com a compilação concluída, pode-se gravar o programa por meio dos seguintes passos: conecte a placa do OpenMote-B em uma porta USB e utilize o comando SCons com o parâmetro *bootload* indicando o dispositivo a ser usado (/dev/ttyUSB1 nesse exemplo) para realizar a gravação, como mostrado na linha de comando a seguir:

```
$ scons board=openmote—b project=at86rf215—sensors—tx compiler=gcc verbose=0 bootload=/dev/
ttyUSB1
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
.
.
.
Starting booloading on /dev/tyyUSB1
Done bootloading on /dev/ttyUSB1
scons: done building targets.
```

Se todas as mensagens no terminal forem exibidas corretamente, significa que o processo foi concluído e a imagem foi devidamente gravada.

2.5.4 Escrita e compilação do código do receptor

Seguindo o caminho "openmote-sbrt/projects/at86rf215/at86rf215-sensors-rx/src/" é possível encontrar o código fonte *main.cpp*, que segue as mesmas especificações para configuração do rádio encontradas no arquivo *main.cpp* do diretório "at86rf215-sensors-tx/src/", que possui o código do transmissor.

Há diferença nos métodos utilizados a fim de configurar o transceptor para função de receptor, bem como enviar as informações dos pacotes recebidos pela porta USB. A seguir o exemplo de código:

```
// Turn AT86RF215 radio on
at86rf215.on();

// Check AT86RF215 radio
status = at86rf215.check();
```

```
if (!status)
{
    // Blink red LED
    board.error();
}

// Set radio callbacks and enable interrupts
at86rf215.setRxCallbacks(RADIO_CORE, &radio_rx_init_cb, &radio_rx_done_cb);
at86rf215.enableInterrupts();

// Wake up and configure radio
at86rf215.wakeup(RADIO_CORE);
at86rf215.configure(RADIO_CORE, RADIO_SETTINGS, RADIO_FREQUENCY, RADIO_CHANNEL);
at86rf215.setTransmitPower(RADIO_CORE, RADIO_TX_POWER);

At86rf215::RadioResult result;
int8_t rssi, lqi;
bool crc;
bool received;

// Initialize packet pointer and length
uint8_t packet_ptr = radio_buffer;
uint16_t packet_len = radio_buffer_len;

// Try to receive a packet
at86rf215.receive(RADIO_CORE);

// Get packet from radio
result = at86rf215.getPacket(RADIO_CORE, packet_ptr, &packet_len, &rssi, &lqi, &crc);

// Check packet has been received successfully
if (result == At86rf215::RadioResult::Success && crc == true)
    {
        uint16_t length;

        // Prepare serial buffer
        length = prepare_serial(serial_buffer, packet_ptr, packet_len, lqi);

        // Send packet via Serial
        serial.write(serial_buffer, length, true);
    }

uint16_t length;
// Prepare serial buffer
length = prepare_serial(serial_buffer, packet_ptr, packet_len, lqi);

// Send packet via Serial
serial.write(serial_buffer, length, true);
```

```
// Turn AT86RF215 radio off
at86rf215.off();
```

Para compilar e gravar o projeto para função de receptor, basta executar os comandos SCons, como mostrado para o código do transmissor, alterando o parâmetro *project* para "at86rf215-sensors-rx", de modo que o comando a ser digitado fica como a seguir:

```
$ scons board=openmote—b project=at86rf215—sensors—rx compile=gcc verbose=0
```

Para a gravação do programa no OpenMote-B, também deve-se seguir os mesmos passos descritos para o transmissor, apenas substituindo "at86rf215-sensors-tx" por "at86rf215-sensors-rx".

2.6 Conclusões

No decorrer deste capítulo, foi apresentada ao leitor uma ampla visão das redes de sensores sem fio e posteriormente uma visão mais detalhada sobre redes de longo alcance e baixa potência, conhecidas pelo acrónimo LPWAN. Abordou-se também o desenvolvimento, ao longo do tempo, de um padrão criado pelo IEEE e já bem difundido, o IEEE 802.15.4, que serve como base para diversas tecnologias de redes de sensores sem fio. Em seguida, foi realizada uma discussão mais detalhada sobre as definições incluídas a partir da emenda 802.15.4g, por meio do fornecimento de referenciais teóicos, em como a descrição de diversos artigos que descrevem estudos experimentais com essa tecnologia. Por fim, foi apresentada uma plataforma IoT (o OpenMote-B), que pode ser utilizada na condução de experimentos práticos com redes que usam as opções de camada física definidas na emenda IEEE 802.15.4g.

Em uma pesquisa sobre IoT, no site da Forbes, existe a previsão de conexão em rede de mais de 75 bilhões de dispositivos IoT até 2025 [14]. Tal nível de adoção sustentará o crescimento constante das novas implantações de nós sensores em todo mundo, durante os próximos anos, corroborando assim a importância da discussão realizada no presente capítulo, sobre os desafios e as dificuldades que os pesquisadores desta área enfrentam, bem como da descrição de novas oportunidades de pesquisa que possam preencher as atuais lacunas de conhecimento sobre o assunto.

Referências Bibliográficas

[1] Institute of electrical and electronics engineers - ieee. https://www.ieee.org/. Acessado em: Setembro de 2020.

[2] Ieee standard for low-rate wireless networks. *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pages 1–800, 2020.

[3] Amina Seferagić, Jeroen Famaey, Eli De Poorter, and Jeroen Hoebeke. Survey on wireless technology trade-offs for the industrial internet of things. *Sensors*, 20(2):488, 2020.

[4] IEC62591. Industrial networks - wireless communication network and communication profiles - wirelesshart. Technical report, International Electrotechnical Commission, Geneva, Switzerland, 2016.

[5] IEC62734. Industrial networks wireless communication network and communication profiles isa 100.11a. Technical report, International Electrotechnical Commission, Geneva, Switzerland, 2014.

[6] 6tisch - ipv6 over the tsch mode of ieee 802.15.4e. https://datatracker.ietf.org/wg/6tisch/documents/. Acessado em: Setembro de 2020.

[7] Wafaa Anani, Abdelkader Ouda, and Ali Hamou. A survey of wireless communications for iot echo-systems. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, pages 1–6. IEEE, 2019.

[8] IEEE. Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans) amendment 3: Physical layer (phy) specifications for low-data-rate, wireless, smart metering utility networks. *IEEE Std 802.15.4g-2012 (Amendment to IEEE Std 802.15.4-2011)*, pages 1–252, 2012.

[9] Jonathan Muñoz, Tengfei Chang, Xavier Vilajosana, and Thomas Watteyne. Evaluation of ieee802. 15.4 g for environmental observations. *Sensors*, 18(10):3468, 2018.

[10] Wi-sun alliance. https://wi-sun.org/. Acessado em: Setembro de 2020.

[11] IEEE. Ieee standard for low-rate wireless networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pages 1–709, 2016.

[12] IEEE. Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans) amendment 1: Mac sublayer. *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pages 1–225, 2012.

[13] Xavier Vilajosana, Pere Tuset, Thomas Watteyne, and Kris Pister. Openmote: Open-source prototyping platform for the industrial iot. In *International Conference on Ad Hoc Networks*, pages 211–222. Springer, 2015.

[14] Bharat S Chaudhari, Marco Zennaro, and Suresh Borkar. Lpwan technologies: Emerging application characteristics, requirements, and design considerations. *Future Internet*, 12(3):46, 2020.

[15] Figure with wsn radio technologies. https://embeddedams.nl/different-ways-to-connect-iot-devices-to-transmit-and-receive-data/. Acessado em: Setembro de 2020.

[16] Zigbee alliance. https://zigbeealliance.org/solution/zigbee/. Acessado em: Setembro de 2020.

[17] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. A comparative study of lpwan technologies for large-scale iot deployment. *ICT express*, 5(1):1–7, 2019.

[18] Etsi - european telecommunications standards institute. https://www.etsi.org/. Acessado em: Setembro de 2020.

[19] 3gpp - 3rd generation partnership project. https://www.3gpp.org/. Acessado em: Setembro de 2020.

[20] Wsig - wsigweightless special interest group. http://www.weightless.org/. Acessado em: Setembro de 2020.

[21] Lora alliance. https://lora-alliance.org/. Acessado em: Setembro de 2020.

[22] Dash7 alliance. https://dash7-alliance.org/. Acessado em: Setembro de 2020.

[23] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2):855–873, 2017.

[24] Jose A Gutierrez, Marco Naeve, Ed Callaway, Monique Bourgeois, Vinay Mitter, and Bob Heile. Ieee 802.15. 4: a developing standard for low-power low-cost wireless personal area networks. *IEEE network*, 15(5):12–19, 2001.

[25] Ieee standard for telecommunications and information exchange between systems - lan/man specific requirements - part 15: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpan). *IEEE Std 802.15.4-2003*, pages 1–680, 2003.

[26] Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans). *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pages 1–320, 2006.

[27] Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans). *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pages 1–314, 2011.

[28] Chin-Sean Sum, Ming-Tuo Zhou, Fumihide Kojima, and Hiroshi Harada. Experimental Performance Evaluation of Multihop IEEE 802.15.4/4g/4e Smart Utility Networks in Outdoor Environment. *Wireless Communications and Mobile Computing*, 2017:1–13, 2017.

[29] Kuor-Hsin Chang and B. Mason. The ieee 802.15.4g standard for smart metering utility networks. In *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pages 476–480, Nov 2012.

[30] Jonathan Muñoz, Paul Muhlethaler, Xavier Vilajosana, and Thomas Watteyne. Why channel hopping makes sense, even with ieee802. 15.4 ofdm at 2.4 ghz. In *2018 Global Internet of Things Summit (GIoTS)*, pages 1–7. IEEE, 2018.

[31] Pere Tuset-Peira, Francisco Vazquez-Gallego, Jonathan Munoz, Thomas Watteyne, Jesus Alonso-Zarate, and Xavier Vilajosana. Experimental interference robustness evaluation of ieee 802.15.4-2015 oqpsk-dsss and sun-ofdm physical layers for industrial communications. *Electronics*, 8(9):1045, Sep 2019.

[32] Domenico De Guglielmo, Simone Brienza, and Giuseppe Anastasi. Ieee 802.15. 4e: A survey. *Computer Communications*, 88:1–24, 2016.

[33] Jaime Dias, Filipe Ribeiro, Rui Campos, Manuel Ricardo, Luıs Martins, Fernando Gomes, and Antonio Carrapatoso. Evaluation of an rpl/6lowpan/ieee 802.15. 4g solution for smart metering in an industrial environment. In *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 1–4. IEEE, 2016.

[34] Philipp Sommer, Yannick Maret, and Dacfey Dzung. Low-power wide-area networks for industrial sensing applications. In *2018 IEEE International Conference on Industrial Internet (ICII)*, pages 23–32. IEEE, 2018.

[35] Francesca Righetti, Carlo Vallati, Daniela Comola, and Giuseppe Anastasi. Performance measurements of ieee 802.15. 4g wireless networks. In *2019 IEEE 20th International Symposium on"A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 1–6. IEEE, 2019.

[36] Richard E Cagley, Scott A McNally, and Michael R Wiatt. Dynamic channel allocation for dynamic spectrum use in wireless sensor networks. In *MILCOM 2006-2006 IEEE Military Communications conference*, pages 1–5. IEEE, 2006.

[37] So Hasegawa, Song-Ju Kim, Yozo Shoji, and Mikio Hasegawa. Performance evaluation of machine learning based channel selection algorithm implemented on iot sensor devices in coexisting iot networks. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–5. IEEE, 2020.

[38] So Hasegawa, Ryoma Kitagawa, Takumi Ito, Takashi Nakajima, Song-Ju Kim, Yozo Shoji, and Mikio Hasegawa. Performance evaluation of machine learning based channel selection algorithm implemented on iot sensor devices and its application to wireless sensor network for building monitoring system. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 161–166. IEEE, 2020.

[39] Jing Ma, Tomoya Nagatsuma, Song-Ju Kim, and Mikio Hasegawa. A machine-learning-based channel assignment algorithm for iot. In *2019 International Conference on*

Artificial Intelligence in Information and Communication (ICAIIC), pages 1–6. IEEE, 2019.

[40] Jing Ma, So Hasegawa, Song-Ju Kim, and Mikio Hasegawa. A reinforcement-learning-based distributed resource selection algorithm for massive iot. *Applied Sciences*, 9(18):3730, 2019.

[41] G Thirunavukkarasu, G Murugesan, R Soundharya, K Vijiyalakshmi, and N Tamil. Adaptive modulation for wireless sensor networks. In *2018 International Conference on Intelligent Computing and Communication for Smart World (I2C2SW)*, pages 185–189. IEEE, 2018.

[42] Pere Tuset-Peiró, Ferran Adelantado, Xavier Vilajosana, and Ruan Delgado Gomes. Determinism through modulation diversity: Can combining multiple ieee 802.15. 4-2015 sun modulations improve communication reliability? In *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020.

[43] Ruan Delgado Gomes, Pere Tuset-Peiró, and Xavier Vilajosana. Improving link reliability of ieee 802.15.4g sun with adaptive modulation diversity. In *2020 IEEE 31th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2020.

[44] Pere Tuset-Peiró, Ruan D Gomes, Pascal Thubert, Eva Cuerva, Eduard Egusquiza, and Xavier Vilajosana. A dataset to evaluate ieee 802.15. 4g sun for dependable low-power wireless communications in industrial scenarios. *Data*, 5(3):64, 2020.

[45] Namyoon Lee and Robert W Heath Jr. Advanced interference management technique: Potentials and limitations. *IEEE Wireless Communications*, 23(3):30–38, 2016.

[46] Gan Zheng, Ioannis Krikidis, Christos Masouros, Stelios Timotheou, Dimitris-Alexandros Toumpakaris, and Zhiguo Ding. Rethinking the role of interference in wireless networks. *IEEE Communications Magazine*, 52(11):152–158, 2014.

[47] Charalampos Orfanidis, Laura Marie Feeney, Martin Jacobsson, and Per Gunningberg. Cross-technology clear channel assessment for low-power wide area networks. In *The 16th IEEE International Conference on Mobile Ad-Hoc and Smart Systems*, 2019.

[48] Kai Samejima, Ryota Okumura, Keiichi Mizutani, and Hiroshi Harada. Evaluation of csl-based low power mac protocol for wireless smart metering networks. In *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2020.

[49] Kai Samejima, Ryota Okumura, Takuya Habara, Keiichi Mizutani, and Hiroshi Harada. Practical analysis of csl low power mac protocol based on ieee 802.15. 4e frame structure. In *2018 21st International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 636–640. IEEE, 2018.

[50] Arib - association of radio industries and businesses. https://www.arib.or.jp/english/index.html. Acessado em: Setembro de 2020.

[51] Glenn Daneels, Carmen Delgado, Steven Latré, and Jeroen Famaey. Towards slot bonding for adaptive mcs in ieee 802.15. 4e tsch networks. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2020.

[52] Pere Tuset-Peiró, Xavier Vilajosana, and Thomas Watteyne. Openmote+: A range-agile multi-radio mote. In *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, EWSN '16, pages 333 – 334, USA, 2016. Junction Publishing.

[53] Texas Instruments. Cc2538 powerful wireless microcontroller system-on-chip for 2.4-ghz ieee 802.15.4, 6lowpan, and zigbee applications (rev. swrs096d), 2015.

[54] Atmel. At86rf215 device family: Sub-1ghz/2.4ghz transceiver and i/q radio for ieee std 802.15.4-2015 (rev. 42415e), 2016.

CAPÍTULO

3 Novas plataformas de comunicação para Internet das Coisas - experiências e práticas

Juliana F. Ibiapina, Victória T. Oliveira, Leila Maria R. de Sousa, Carlos Eduardo da S. Brandão, Francisco Jeferson da S. Pontes, Maria Alice J. de Sousa, Mateus P. Alves, Wendley S. Silva (UFC)

3.1 Introdução

Os primeiros computadores eletrônicos surgiram na década de 1950, logo depois, na década de 1980, os protocolos *Transmission Control Protocol* (TCP) e *Internet Protocol* (IP) foram implementados e com isso veio o uso comercial da Internet e sua popularização. Rapidamente a Internet evoluiu de uma rede de documentos HTML (*Hyper Text Markup Language*) estáticos para a Web 2.0, proporcionando interação e colaboração entre os usuários por meio dos serviços de redes sociais, *blogs* e *wikis* - tecnologias que se tornaram essenciais nas interações sociais modernas bem como nos negócios globais [1].

Em paralelo ao crescimento da Internet, as tecnologias de *Wireless Sensors Network* (WSN) foram se desenvolvendo e a convergência dessas duas áreas resultou na criação do conceito denominado *machine-to-machine*, o qual define uma comunicação entre máquinas com pouca ou nenhuma intervenção humana. Tal conceito se baseia na ideia de que máquinas conectadas têm maior valor que máquinas isoladas porque tal conexão proporciona o desenvolvimento de automação e inteligência para as aplicações [2].

Em 1991, Mark Weiser definiu o termo computação ubíqua, afirmando que mais e mais objetos cotidianos estariam conectados à internet [3]. Essa definição foi a semente do que posteriormente viria a ser conhecida como *Internet of Things* (IoT). A geração atual presenciou uma rápida proliferação dos dispositivos de *Internet of Things*. O relatório da Business Insider[1], publicado em março de 2020, prevê que mais de 41 bilhões de dispositivos

[1]https://www.businessinsider.com/internet-of-things-report

IoT estarão conectados em 2027, comparado a 8 bilhões em 2019. Pesquisas indicam que a taxa de adoção das tecnologias IoT é cinco vezes mais rápida que a do crescimento da eletricidade e telefonia [3], demonstrando sua importância. Apesar de ser um campo de conhecimento considerado novo, a IoT é tida como umas das iniciativas tecnológicas mais importantes da atualidade [4].

O termo *Internet of Things* já foi definido diversas vezes na literatura e não há um entendimento comum sobre o que ele abrange [4]. Apesar de tantas variações, percebe-se um objetivo em comum em todas elas. Em geral, o termo *Internet of Things* (IoT) refere-se à rede de objetos físicos dotados de comunicação, sensoreamento e interação com o mundo real [5]. A IoT transforma objetos tradicionais em objetos inteligentes integrando-os às tecnologias de computação ubíqua e pervasiva, dispositivos embarcados, tecnologia de comunicação, redes de sensores, protocolos de Internet e aplicações para os usuários [6].

A variedade de dispositivos e de protocolos de comunicação é uma das principais características da IoT, e conforme citado anteriormente, diversas tecnologias embarcadas foram desenvolvidas para permitir que objetos ("*things*") possam coletar e compartilhar dados. Além disso, a grande quantidade de objetos conectados à rede gera novos desafios, exemplo disso é o protocolo IPv6, criado porque o IPv4 não mais suportaria a quantidade de dispositivos conectados. Essas características da IoT fazem com que uma arquitetura flexível seja fundamental para seu desenvolvimento.

Na literatura, muitas arquiteturas para IoT foram propostas ao longo dos anos e ainda não convergiram para um modelo de referência [6]. No trabalho [6] os autores propõem uma estrutura que divide um projeto em seis blocos/elementos necessários para que as funcionalidades essenciais da IoT sejam entregues pelo mesmo. São eles: identificação, sensoreamento, comunicação, computação, serviços e semântica, os quais serão explicados a seguir.

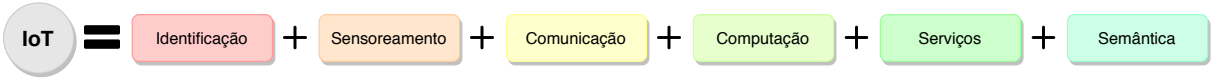


Figura 3.1 – Elementos da IoT [6] [7]

O primeiro elemento é chamado de **identificação**, sua função é atribuir identificadores únicos aos objetos, sendo uma etapa fundamental para os demais etapas. A identificação é possível por meio de tecnologias de comunicação sem fio como RFID (*Radio Frequency Identification*) e NFC (*Near Field Communication*). O próximo elemento é denominado de **sensoreamento**, mas pode incluir tanto sensores como atuadores. O papel dos sensores é coletar dados sobre o contexto e as condições em que se encontram os objetos, posteriormente encaminhando o que foi coletado para centros de processamento e armazenamento, já os atuadores são os responsáveis por reagir de acordo com os dados recebidos e, quando necessário, podem manipular o ambiente [8].

O elemento de **comunicação** é responsável por conectar os objetos inteligentes a internet por meio de protocolos de comunicação como WiFi, Bluetooth, IEEE 802.15.4 e RFID. O quarto elemento é o de **computação**, ele é responsável por todo processamento

local necessário, isso se dá por meio de microcontroladores, processadores e FPGAs (*Field-Programmable Gate Array*). Como próximo elemento tem-se os **serviços**, este dividido entre as classes: Serviços de Identificação, Serviços de Agregação de Dados e Serviços de Ubiquidade. Por fim, o sexto elemento é a **semântica**, responsável por gerar conhecimento embasado nos dados recolhidos, possibilitando a utilização mais eficiente dos recursos IoT disponíveis, fundamental para aprimorar as aplicações. Podem ser utilizadas técnicas como *Resource Description Framework* (RDF), *Web Ontology Language* (OWL) e *Efficient XMl Interchange* (EXI) [8].

A IoT propicia a automatização de vários setores da economia e da vida, tornando possível interligar objetos em pequenas e grandes escalas. São exemplos de aplicações IoT:

Smart home: oferece ao usuário a automação de sua residência através de acesso remoto;

Healthcare: possibilita monitoramento e acompanhamento de pacientes em suas casas, provendo assistência médica remota por meio de sensores que coletam informações do paciente;

Smart Grid: automatiza o setor elétrico por meio de leitores inteligentes *(smart meters)* instalados nas residências, comércios ou industrias;

Smart manufacturing: automação de processos fabris reais sobre os dispositivos industriais;

Smart farming: são fazendas automatizadas que monitoram seus cultivos e suas criações por meio de sensores para coleta de dados e máquinas automatizadas.

Desenvolver um sistema baseado em IoT é uma tarefa que envolve a utilização de diversas ferramentas e tecnologias relacionando-se também a tópicos como limitação energética, capacidade de armazenamento e processamento bem como variados tipos de padrões de comunicação. Assim, este trabalho apresentará as principais plataformas e protocolos de comunicação recém-lançados ou aprimorados nos últimos anos para a implementação de projetos de IoT, trazendo uma descrição detalhada de cada uma dessas tecnologias, além de práticas abordando as rotinas de comunicação entre elas. Dentre as ferramentas apresentadas estão: Raspberry Pi Zero, NodeMCU ESP32, LoRaWan, ESP-Now, Z-Wave e outras.

3.2 Plataformas de IoT

Esta seção é destinada a apresentar os principais dispositivos embarcados lançados nos últimos anos para o desenvolvimento de projetos IoT, incluindo os que serão utilizados nas práticas descritas nas próximas seções deste trabalho.

3.2.1 Raspberry Pi 4 e Raspberry Pi Zero

O Raspberry Pi é um *single-board computer* de baixo custo, que pode ser conectado a periféricos de computadores como monitor, teclado, mouse e câmera. Ele foi desenvolvido por uma equipe da Universidade de Cambridge em 2006 com a intenção de estimular o ensino de ciência da computação nas escolas [9, 10]. A Raspberry Pi *Foundation* dá suporte ao Raspbian, um sistema operacional de *software* livre baseado no Linux Debian e otimizado para o *hardware* do Raspberry. Além do Raspbian, outros sistemas operacionais também foram desenvolvidos para essa placa, como o Windows IoT Core e o Ubuntu Core.

O Raspberry Pi 4 B é o modelo mais recente dessa linha de *single-board computers* e o primeiro a trazer uma experiência completa de *desktop*. Os seus principais diferenciais em relação aos modelos anteriores incluem um processador *quad-core* de 64 bits de alto desempenho, duas portas micro-HDMI dando suporte para resoluções de 4K, até 8 Gb de RAM, dual LAN sem fio de 2,4/5,0 GHz, Bluetooth 5.0, Gigabit Ethernet e USB 3.0[1].

O Raspberry Pi Zero é um modelo menor da linha de *sungle-board computers* Raspberry Pi. Em relação ao modelo Pi 4, esta placa possui metade do tamanho e menor consumo energético. Entre suas principais especificações estão uma CPU *single-core* de 1 GHz, Bluetooth *Low Energy* (BLE) 4.1, 512 MB de RAM, uma porta mini HDMI, uma porta USB OTG (*On-The-Go*) e duas micro-USB, GPIO (*General Purpose Input/Output*) de 40 pinos e conector CSI (*Camera Serial Interface*)[2]. Esta pequena plataforma, apesar de possuir menor poder computacional, é suficiente para realizar diversos projetos de IoT.

Além de poder computacional, o Raspberry Pi também proporciona conexão direta com sensores e atuadores por meio da GPIO, o que torna possível a integração com módulos permitindo-lhe adaptar-se a diversas necessidades, porém possui a limitação de não suportar entradas/saídas analógicas. Existem casos em que a aplicação inclui uma estrutura de rede heterogênea e a flexibilidade e o poder computacional dessa plataforma são utilizados fazendo com que ela atue como elemento central [11].

O Raspberry Pi foi inicialmente criado para um propósito educacional, mas tem sido utilizado em diversas áreas de aplicações em IoT, mostrando ser uma das principais plataformas de desenvolvimento. Entre alguns dos seus usos pode-se citar projetos de processamento de imagens e reconhecimento facial [12, 13], automação residencial, *Smart Home* e segurança nesses ambientes [14, 15, 16] e monitoramento energético [17].

3.2.2 NodeMCU ESP32

O ESP32 é um sistema dual-core com duas CPUs *Architecture* de 32 bits Xtensa LX6. A memória embutida, memória externa e periféricos estão localizados no barramento de dados e/ou barramento de instruções dessas CPUs. O NodeMCU ESP32 é um sistema de baixa potência em um microcontrolador da Espressif com Wi-Fi 802.11 e módulo Bluetooth integrado. O chip opera a 160 MHz ou 240 MHz e possui desempenho de até 600 DMIPS

[1] https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/
[2] https://www.raspberrypi.org/products/raspberry-pi-zero/

[18]. O microcontrolador possui dois núcleos - PRO_CPU para protocolo e APP_CPU para aplicação, porém, as finalidades destes não são fixas. O espaço de endereço para ambos os barramentos de dados e instrução é de 4 GB e o espaço de endereço periférico é de 512 KB. Além disso, as memórias incorporadas são de 448 KB ROM, 520 KB SRAM e duas memórias RTC de 8 KB. A memória externa suporta até quatro vezes a Flash de 16 MB. O ESP32 é capaz de funcionar em ambientes com temperatura na faixa de -40°C até 125°C e pode funcionar como um sistema *standalone* ou como escravo de outro dispositivo [19].

O ESP32 é projetado para aplicativos móveis, eletrônicos vestíveis e *Internet of Things* (IoT). Ele apresenta todas as características de última geração dos chips de baixo consumo de energia, incluindo controle de *clock* de baixa granularidade, vários modos de potência e escalonamento de potência dinâmico. Por exemplo, em um cenário de aplicação de *hub* de sensor de IoT de baixa potência, o ESP32 é ativado periodicamente e apenas quando uma condição especificada é detectada. O ciclo de baixa carga é usado para minimizar a quantidade de energia que o chip gasta. A saída do amplificador de potência também é ajustável, contribuindo assim para um equilíbrio ideal entre faixa de comunicação, taxa de dados e consumo de energia [20].

A variedade de aplicação do ESP32 não se limita a projetos IoT. Em [20] há um exemplo de um projeto que visa construir um protótipo de um osciloscópio sem fio portátil baseado no ESP32 e um aplicativo de *smartphone* como unidade de exibição e controle. Em [21] é apresentado um sistema de bombeamento solar de água automatizado de baixo custo para irrigação em países em desenvolvimento. No sistema proposto em [22], sensores são utilizados para monitorar a qualidade do ar de cidades poluídas. Os dados medidos são colocados no servidor usando uma API REST (*Representational State Transfer*) e ESP32. Todas as leituras dos sensores são exibidas na página da *web* e no aplicativo móvel.

3.2.3 DragonBoard 410c

A DragonBoard 410c é uma placa de desenvolvimento do tamanho de um cartão de crédito lançada pela Qualcomm. Foi projetado para oferecer suporte rápido ao desenvolvimento de softwares e protótipos, para fins educacionais e para o uso em produtos comerciais que necessitem de computadores embarcados, como robôs, dispositivos médicos, *smart buildings* e outros [23].

Essa placa de desenvolvimento é compatível com a especificação *96Boards Consumer Edition* e possui um processador Qualcomm Snapdragon 410 Quad-core ARM-Cortex A53 com velocidade de *clock* de até 1,2 GHz por núcleo, capaz de operar em 32 bits e 64 bits. Essa placa suporta Android 5.1, distribuições Linux baseadas no Debian e Win10 IoT Core, além de WLAN, Bluetooth e GPS (*Global Positioning System*). Ela foi projetada para suportar diversas funcionalidades, incluindo multimídia por meio da GPU Qualcomm Adreno 306, ISP (*Image Signal Processor*) integrado com suporte para câmera de até 13 MP e reprodução e captura de vídeo HD 1080p [3].

[3]https://www.arrow.com/en/reference-designs

Um exemplo de aplicação da Dragon Board em um sistema, é o de monitoramento de espectro de baixo custo, o monitoramento do espectro consiste em fazer medições do ambiente por meio de uma antena para determinar o nível de ocupação de porções específicas do espectro. O intuito de conhecer os padrões de utilização do espectro é possibilitar aos órgãos que regulamentam o uso desse recurso podem propor novas aplicações e descubram buracos de espectro para explorá-los posteriormente. A função da Dragonboard 410C3 neste sistema é de processar e analisar as amostras para determinar o estado do canal, em outras palavras se o mesmo é um canal vazio ou um canal ocupado [24]. O diretório a seguir contém um conteúdo complementar sobre a DragonBoard e seu funcionamento com outras plataformas, disponível em: https://github.com/nuclicufc/pratica_Dragonboard.

3.3 Protocolos de IoT

Devido ao aumento de dispositivos conectados a Internet, faz-se necessário o estudo e desenvolvimento de protocolos novos para a utilização em IoT. Esse fenômeno ocorre, sobretudo pela heterogeneidade dos dispositivos utilizados em ambientes IoT. Em decorrência, faz-se essencial o estudo e o desenvolvimento de protocolos que possam, por exemplo, suprir limitações impostas pelos dispositivos, como ocorre em muitos casos. Como consequência, vários grupos foram criados para fornecer protocolos de apoio à Internet das Coisas, incluindo alguns liderados pelo *Institute of Electrical and Electronics Engineers* (IEEE), *World Wide Web Consortium* (W3C), *EPCglobal*, *Internet Engineering Task Force* (IETF) e o *European Telecommunications Standards Institute* (ETSI) [25].

Alguns dos principais protocolos utilizados em ambientes IoT podem ser vistos na Tabela 3.1, separados em 4 principais grupos [25], sendo eles: protocolos de aplicação, protocolos de descoberta de serviços, protocolos de infraestrutura e outros protocolos influentes.

Tabela 3.1 – Protocolos relacionados à IoT [25] [26]

<table>
<tr><td colspan="2">Protocolos de aplicação</td><td>ESP-Now</td><td>MQTT-SN</td><td>LWM2M</td></tr>
<tr><td colspan="2">Protocolos de descoberta de serviços</td><td colspan="2">mDNS</td><td>DNS-SD</td></tr>
<tr><td rowspan="4">Protocolos de infraestrutura</td><td>Camada de Transporte</td><td colspan="2">QUIC</td><td>M3DA</td></tr>
<tr><td>Camada de Rede</td><td colspan="2">6LoWPAN</td><td>IPv4/IPv6</td></tr>
<tr><td>Camada de Enlace</td><td>IEEE 802.15</td><td>IEEE 802.11</td><td>LoraWAN</td></tr>
<tr><td>Camada Física</td><td>Z-Wave</td><td>IEEE 802.15</td><td>IEEE 802.11</td></tr>
<tr><td colspan="2">Outros protocolos influentes</td><td colspan="2">IEEE 1888.3</td><td>IPSec</td></tr>
</table>

3.3.1 Protocolos de aplicação

O HTTP (*Hypertext Transfer Protocol*) foi projetado para redes com computadores pessoais, com maior poder de processamento se comparado aos dispositivos utilizados em redes IoT. Sendo assim, surgem alguns protocolos na camada de aplicação como alternativa ao HTTP, como, por exemplo, o MQTT-SN, LWM2M e ESP-NOW, ambos projetados para troca

de informação entre dispositivos com baixo poder computacional. A seguir, alguns destes protocolos serão detalhados [25] [27].

ESP-Now

O protocolo ESP-Now foi publicado pela primeira vez em 2018 para a plataforma ESP32. é um protocolo *peer to peer* (P2P), ou seja, implementa uma comunicação dispositivo para dispositivo, dispensando a necessidade de um roteador. Além disso, é um protocolo de baixo gasto energético; comparando-o com o Wi-Fi, por exemplo, é mais eficiente [28].

Os dispositivos conectados por meio de uma rede ESP-Now possuem três principais métodos de comunicação: *broadcast*, *unicast* e *multicast*. Os métodos *unicast* e *multicast* requerem emparelhamento inicial e limitam a quantidade de receptores a vinte. Já o *broadcast* não necessita de emparelhamento inicial e pode ter um número ilimitado de receptores [29].

LWM2M

O *Lightweight* M2M (LWM2M) é um padrão da *Open Mobile Alliance*, cuja finalidade é provê uma comunicação cliente/servidor rápida e implementável para fornecer serviços *Machine to Machine*(M2M) [30]. O LWM2M fornece uma interface de comunicação segura leve e compacta, juntamente com um modelo de dados eficiente, que juntos permitem o gerenciamento de dispositivos e a ativação de serviços para dispositivos M2M. O LWM2M é direcionado, em particular, a dispositivos restritos, por exemplo, dispositivos como microcontroladores de baixa potência e pequenas quantidades de memória Flash e memória RAM, em redes que requerem o uso eficiente de largura de banda. Ao mesmo tempo, o LWM2M também pode ser utilizado com dispositivos incorporados mais poderosos que se beneficiam de uma comunicação eficiente [31].

O modelo de comunicação LWM2M é baseado em métodos COAP, utilizando GET, PUT, POST e DELETE, fazendo a utilização do UDP na camada de transporte. Os *overheads* de mensagem codificados em binário serão de apenas alguns bytes e os objetos simples e planos com URI uniforme entre os dispositivos tornam o protocolo mais adequado para conectividade de dispositivo restrita e fácil gerenciamento [32].

O LWM2M *Enabler* define o protocolo de comunicação da camada de aplicativo entre um servidor e um cliente. O LWM2M *Server* está normalmente localizado em um data center público ou privado e pode ser hospedado pelo provedor de serviços M2M, provedor de serviços de rede ou provedor de serviços de aplicativos. O LWM2M *Client* reside no dispositivo e é normalmente integrado como uma biblioteca de software ou uma função embutida de um módulo ou dispositivo [32].

MQTT-SN

O MQTT *for Sensor Network*[4] (MQTT-SN) foi desenvolvido para ser o mais próximo possível do MQTT, porém adaptado para comunicação sem fio com restrições, como baixa largura de banda, falha de links etc. O MQTT-SN é um protocolo específico para redes de sensores e não dependente dos serviços de redes subjacentes para operar. Pode operar sobre qualquer camada de transporte, como ZigBee, Bluetooth, UDP, comunicação serial cabeada, entre outras. Qualquer rede que forneça um serviço de transferência de dados bidirecional entre qualquer nó e um *gateway* pode suportar o MQTT-SN.

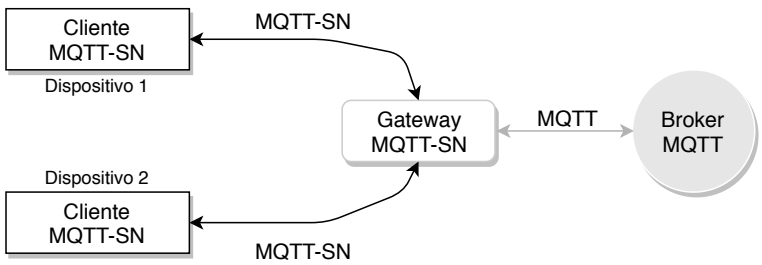
O MQTT-SN é otimizado para implementação em dispositivos de baixo custo e baixo consumo (alimentados por bateria), com processamento e recursos de armazenamento limitados [33]. Em comparação com MQTT, MQTT-SN é caracterizado pelas diferenças que podem ser observadas na Tabela 3.2.

Tabela 3.2 – Comparação entre o MQTT e o MQTT-SN

	MQTT	**MQTT-SN**
Comunicação	TCP-IP	UDP, não IP
Rede	Ethernet, WIFI, 3G	Zigbee, Bluetooth, RF, etc
Tamanho mínimo de mensagem	2 bytes - PING	1 byte
Tamanho máximo de mensagem	<= 24 MB	<128 bytes
Alimentação por bateria	-	Sim
Clientes em modo Sleep	-	Sim
QoS (Quality of Service) -1	-	Sim
Tópico	Nome	Nome e ID

Conforme podemos observar na figura 1, a arquitetura é composta por 3 tipos de componentes: Clientes, *Gateways* e um *Broker*. Os clientes MQTT são conectados ao *broker* MQTT via *gateways* MQTT usando o protocolo MQTT-SN. Há também casos onde o cliente acessa o *broker* através de *forwarders*, nesse caso o *forwarder* apenas encapsula os pacotes MQTT-SN e envia para o *Gateway* MQTT-SN.

Figura 3.2 – Arquitetura.



[4] $http://www.mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf$

3.3.2 Protocolos de descoberta de serviços

A alta escalabilidade da IoT demanda um mecanismo de gerenciamento de recursos que seja apto a se registrar e descobrir recursos e serviços de forma autoconfigurada, eficiente e dinâmica. Os protocolos mais utilizados nesta área são o DNS de *multicast* (mDNS) e DNS *Service Discovery* (DNS-SD), que podem descobrir recursos e serviços oferecidos pelos dispositivos IoT

DNS de *multicast* (mDNS)

O mDNS é um protocolo de descoberta de serviços desenvolvido pelo *Zeroconf IETF group* e sua finalidade é estabelecer os nomes de domínios sem a obrigação de um servidor DNS convencional [34]. O mDNS é a escolha apropriada para dispositivos embarcados fundamentados na Internet, pois não existe a necessidade de reconfiguração manual ou administração extra para gerenciar dispositivos, é capaz de executar sem infraestrutura, podendo continuar seu funcionamento mesmo se ocorrer uma falha de infraestrutura.

O protocolo mDNS opera da seguinte forma: cada nó que ingressa na rede transmite uma mensagem contendo sua descrição de serviço para os outros nós na rede. Todos os nós clientes que recebem esse pacote registram a descrição do serviço e seu provedor correspondente em uma tabela de consulta de serviço local. Quando um serviço é desejado, o cliente procura a descrição do serviço em sua tabela de consulta. Quando um cliente encontra o endereço de um provedor de serviços em sua tabela de consulta, ele envia uma mensagem unicast para o provedor de serviços solicitando o serviço desejado [35].

3.3.3 Protocolos de infraestrutura

Os protocolos de infraestrutura são fundamentais para estabelecer a comunicação subjacente necessária para as aplicações IoT. O roteamento é um elemento-chave da infraestrutura IoT e muitos outros parâmetros desses sistemas, como confiabilidade, escalabilidade e desempenho, que dependem fortemente dessa tecnologia. Portanto, existe uma necessidade de mais investigações sobre melhorias e otimizações de protocolos de roteamento para atender aos requisitos de IoT [25].

Bluetooth

O Bluetooth é uma tecnologia de comunicação sem fio desenvolvida para conectar dispositivos de variadas funções como celulares, televisores, computadores, microcontroladores,entre outros. Vantagens que a tecnologia oferece, além da transmissão de dados, é o baixo custo financeiro para sua implementação e o baixo consumo energético. Seu chip é facilmente implementado a dispositivos eletrônicos, o que possibilita criações de redes *Ad-Hoc* [36]. O Bluetooth é baseado no padrão IEEE 802.15, definindo uma rede PAN(Personal Area Network) operável em um raio médio de 10 metros [37].

A arquitetura do Bluetooth é definido por dois padrões de rede, sendo eles, *piconets* e *scatternet*. Uma rede Bluetooth é chamada de *piconet* quando é composta por até 8 estações (dispositivos), sendo um considerado primário (mestre) e os outros secundários (escravos). Esta comunicação pode ser ponto a ponto ou multiponto. Uma rede Bluetooth scatternet ocorre a partir da combinação de duas ou mais *piconets*. A arquitetura desta rede é semelhante a uma árvore, onde uma estação secundária pode ser a primária para outra *piconet* [37].

O Bluetooth não utiliza o modelo OSI para definir suas camadas, sua pilha de protocolos é definida pelas seguintes camadas: Camada física de rádio, camada de enlace de dados ou banda base, camada de *middleware* (L2CAP), camada de aplicação.

Protocolo IEEE 802.11

Em 1990, a IEEE fundou um grupo de pesquisa, tendo como objetivo principal o desenvolvimento de uma camada Física e uma camada de Enlace para as redes sem fio. Em 1997, o IEEE publicou um padrão específico para redes sem fio, denominado IEEE 802.11, sendo ele o primeiro padrão construído destinado para as redes sem fio [38].

Os padrões das redes sem fio determinam as camadas 1 e 2 do modelo OSI, sendo elas, a camada física e a camada de enlace de dados, respectivamente. A camada física tem como utilidade codificar e decodificar os sinais, a geração e remoção de parâmetros para sincronização, recepção e transmissão de bits e inclui especificações do meio de transmissão [39].

A camada de enlace de dados é dividida em duas camadas, o LLC (*Logical Link Control*) e o MAC (*Media Access Control*). Cada uma delas tem funções diferentes, a camada LLC tem como função prover interface para camadas superiores e executa controle de fluxo e erro de pacotes. Já a camada MAC, na transmissão, reúne os dados dentro de um pacote com endereços e campos, realizando a detecção de erros. Na recepção, tem como função abrir os pacotes e detectar erros e, por último, mas não menos importante, ter o controle de acesso ao meio de transmissão [39].

LoRaWAN

LoRaWAN é uma das tecnologias *Low Power Wide Area Network* (LPWAN) que oferece comunicação de baixa potência e baixa taxa de dados em uma ampla cobertura de área. é um protocolo MAC de código aberto padronizado pela LoRa Alliance[5] executado sobre a camada física LoRa. Foi projetado principalmente para redes de sensores, em que os sensores trocam pacotes com o servidor com uma baixa taxa de dados e intervalos de tempo relativamente longos[40].

A arquitetura de rede LoRaWAN tem uma topologia em estrela, na qual os dispositivos finais podem apenas comunicar-se com os *gateways* LoRaWAN e não diretamente entre si. A rede é composta por três componentes principais: dispositivo final, que podem ser definidos como sensores de consumo de energia reduzido que realizam comunicação com os

[5]https://lora-alliance.org

gateways por meio do LoRa; *gateway*, um dispositivo intermediário que encaminha os pacotes provenientes dos dispositivos finais para um servidor de rede. Podem haver vários em uma única implementação LoRa, e o mesmo pacote de dados pode ser recebido e encaminhado por mais de um *gateway*; e servidor de rede, responsável por desfazer duplicada e decodificar os pacotes recebidos, além disso gera pacotes que devem ser reenviados aos dispositivos finais. Vários *gateways* são conectados a um servidor de rede central. Eles são responsáveis apenas pelo encaminhamento dos pacotes de dados brutos dos dispositivos finais para o servidor de rede, encapsulando-os em pacotes UDP/IP[40, 41].

3.4 Práticas

3.4.1 ESP-Now com ESP8266: Enviando dados para múltiplas placas (*One-to-many*).

ESP-Now é um protocolo desenvolvido pela *Espressif*, que permite que vários dispositivos se comuniquem entre si sem usar *Wi-Fi*, permite a transmissão de pacotes curtos e pode ser usado com placas ESP8266 e ESP32. O protocolo é semelhante ao *Wireless* de baixa potência de 2.4 GHz. Antes da comunicação é necessário que os dispositivos sejam emparelhados, a comunicação é segura e não há necessidade de *Handshake*. Isso significa que depois de emparelhar um dispositivo a conexão é persistente. ESP-Now é muito versátil pode ser usado com comunicação unilateral ou bidirecional em configurações diferentes [42].

A seguinte prática mostra com usar o protocolo ESP-Now para trocar dados entre placas ESP8266 NodeMCU programadas usando Arduino IDE [43]. Ao usar o Arduino IDE a única modificação necessária é a instalação do complemento para usar a placa ESP8266. A biblioteca utilizada espnow.h[6], está presente no complemento para a placa ESP8266 NodeMCU e não precisa ser instalada separadamente. O código da prática está diponível em: `https://github.com/nuclicufc/ESP-Now_with_ESP8266`.

Visão geral do projeto:

- Um ESP8266 atua como remetente;
- Várias placas ESP8266 atuam como receptores. Nesta configuração um ESP8266 envia dados para três outros, simultaneamente. Você deve ser capaz de adicionar mais placas à sua configuração
- O remetente ESP8266 recebe uma mensagem de confirmação se as mensagens forem entregues com êxito. Você sabe quais dispositivos receberam a mensagem e quais não; Como exemplo, valores aleatórios diferentes serão trocados entre as placas.

[6]https://github.com/esp8266/Arduino/blob/master/tools/sdk/include/espnow.h

Obtendo o endereço MAC das placas

Para se comunicar via ESP-Now, você precisa saber o endereço MAC do receptor ESP8266. é assim que você sabe para qual dispositivo enviará as informações. Cada ESP8266 tem um endereço MAC exclusivo e é assim que identificamos cada placa para enviar dados a ela usando ESP-NOW. Para obter o endereço MAC da sua placa, faça upload do código presente na pasta MAC do repositório. O endereço MAC será mostrado no monitor serial do Arduino IDE. Esse processo deve ser repetido para cada placa receptora.

NodeMCU ESP8266 Principal(*Sender*)

No *sketch* há quatro blocos importantes para o funcionamento da aplicação:

1. Inicializar o ESP-Now;
2. Registrar uma função de retorno ao enviar dados - a função OnDataSent() será executada quando uma mensagem for enviada. Isso pode dizer se a mensagem foi entregue com sucesso ou não;
3. Adicionar um dispositivo(receptor);
4. Enviar uma mensagem para o dispositivo.

Para iniciar o ESP-Now basta usar a função esp_now_init(). é necessário definir esta placa como remetente para isso o argumento ESP_NOW_ROLE_CONTROLLER, precisa ser usado como argumento da função esp_now_set_self_role(). Depois disso, precisamos emparelhar com outros dispositivos ESP-Now para enviar dados.

No exemplo, são enviados dados para três placas. Para enviar uma mensagem diferente para cada dispositivo, você precisa criar uma estrutura de dados para cada um deles. Essa estrutura se chamará test_struct e ela contém três variáveis inteiras. Você pode alterar isso para enviar os tipos de variáveis que desejar. Crie uma nova variável do tipo test_struct que é chamada de *test* que armazenará os valores das variáveis. Neste caso, estamos enviando o mesmo tipo de estrutura para os dispositivos.

Em seguida, defina a função OnDataSent(). Esta é uma função de retorno de chamada que será executada quando uma mensagem for enviada. Neste caso, esta função imprime se a mensagem foi entregue com sucesso ou não e para qual endereço MAC. Então, é possível saber quais dispositivos receberam a mensagem ou não.

Uma lista vinculada das informações do dispositivo local e as informações do dispositivo usado como receptor será mantida na camada de baixo nível do ESP-Now. As informações dos dispositivos são usadas para enviar e receber dados. ESP-Now mantém as informações essenciais, como endereço MAC e chave na camada inferior.

Em seguida, atribua valores diferentes às variáveis de cada estrutura. Neste exemplo, estamos apenas configurando-os para números aleatórios. Finalmente, você precisa chamar a função esp_now_send() para cada receptor. Esta função enviará os dados armazenados nas variáveis de teste.

ESP8266 NodeMCU(*receivers*)

No lado do receptor, o *sketch* deve incluir:

1. Inicializar o ESP-Now;
2. Registre-se para receber uma função de retorno de chamada (OnDataRecv). Esta é uma função que será executada quando uma mensagem for recebida
3. Dentro dessa função de retorno de chamada, salve a mensagem em uma variável para executar qualquer tarefa com essa informação.

ESP-Now funciona com funções de retorno que são chamadas quando um dispositivo recebe uma mensagem ou quando uma mensagem é enviada (você obtém se a mensagem foi entregue com sucesso ou se falhou).

Com todas as placas ligadas, abra o Arduino IDE Serial Monitor da porta COM à qual o remetente está conectado. Você deve começar a receber mensagens de "*Delivery Success*" com o endereço MAC do destinatário correspondente no Monitor Serial.

3.4.2 Monitoramento de sensor através de um servidor Web com ESP32 LoRa

A prática propõe a construção de um sistema de monitoramento de sensores com um servidor *Web*[44] utilizando uma placa WiFi LoRa 32(V2), placa de desenvolvimento produzida pela empresa Heltec Automation que possui um módulo ESP32, equipado com *WiFi* e *Bluetooth Low Energy* (BLE). O diferencial da placa está na tecnologia LoRa com *chip* SX1276, que permite a realização de transmissões sem fio na frequência de 915 MHz (América) e de 868 MHz (Europa) com um baixo consumo energético, longo alcance e grande capacidade anti-interferência. Além disso possui um *display* OLED de 0,96 polegadas com resolução de 128x64 *pixels* e tem um circuito de carga e descarga de bateria de lítio.

O sistema de monitoramento pode ser analisado em duas partes: o transmissor, composto por uma placa WiFi LoRa 32(V2) e o módulo com sensor DHT11, onde são obtidos os valores de temperatura e umidade do ambiente que serão enviados através do protocolo LoRaWAN para o receptor; e o receptor que obtém essa mensagem e os exibe em uma página *Web* através de um servidor hospedado localmente na própria placa. O diretório com os códigos a serem enviados para as placas e demais arquivos está disponível em: `https://github.com/nuclicufc/webServer_LoRa`.

Preparando o Arduino IDE

Para a programação do código é utilizado a IDE (*Integrated Development Environment*) do Arduino[7] e é necessária a instalação de dois pacotes: o pacote de suporte à placa ESP32, que fornece o necessário para a programação da maioria das placas baseadas em ESP32; e o

[7]https://www.arduino.cc/en/Main/software

uploader de arquivos ESP32, que permite acesso ao SPIFFS (*Serial Peripheral Interface Flash File System*), um sistema de arquivos leve para microcontroladores que permite acessar a memória *flash*, semelhante a um sistema de arquivos de computadores pessoais e tem como função carregar arquivos de configuração, ou arquivos de um servidor *Web*, como é o caso apresentado. Assim, em vez de escrever um código HTML para o servidor como uma *string* diretamente no código, podem ser escritos o HTML e CSS em um arquivo separado e salvo no sistema de arquivos ESP32.

Para a instalação do pacote de suporte à placa ESP32[8] é necessário acessar a janela de preferências do Arduino IDE em: Arquivo > Preferências. Na tela exibida há um campo denominado ”URLs Adicionais para Gerenciadores de Placas” que deve ser preenchido por: `https://dl.espressif.com/dl/package_esp32_index.json` e finalizar o processo clicando em ”OK”. Em seguida, é necessário acessar a janela de gerenciador de placas em: Ferramentas > Placa: ... > Gerenciador de Placas, na qual deve-se pesquisar ”ESP32” e instalar a opção denominada ”esp32 by Espressif Systems”. Após a instalação deve selecionar a placa utilizada em: Ferramentas > Placa: ... > ESP32 Arduino > Heltec WiFi LoRa 32(V2).

Após a instalação do pacote de suporte à placa, agora é a vez de instalar o *uploader* do sistema de arquivos ESP32. Primeiro é necessário baixar o arquivo ESP32FS-1.0.zip[9]. Finalizado o *download*, é necessário acessar a pasta *tools* no diretório em que o Arduino IDE foi instalado, que por padrão fica em: C:\Program Files (x86)\Arduino\tools, descompactar o arquivo baixado anteriormente e reiniciar a IDE. Se a instalação ocorrer com sucesso, na aba ”Ferramentas” terá a opção: ”*ESP32 Sketch Data Upload*”.

Bibliotecas

Algumas bibliotecas devem ser instaladas no Arduino IDE para auxiliar a execução da prática. Para o transmissor, onde ocorrem as leituras de temperatura e umidade a partir do sensor DHT e o envio desses dados para o receptor, é necessária a instalação das seguintes bibliotecas em: Ferramentas > Gerenciador de Bibliotecas.

- LoRa por Sandeep Mistry, para lidar com a comunicação entre transmissor e receptor;
- Adafruit SSD1306 por Adafruit, usada para a interação com o OLED;
- DHT sensor library por Adafruit, envolvida na obtenção das informações de umidade e temperatura.

Para o receptor, onde são recebidos os valores de temperatura e umidade do transmissor e os exibe, juntamente com o tempo da última leitura e o RSSI (*Received Signal Strength Indication*) em uma página *Web* hospedada na própria placa, é necessária a instalação das bibliotecas:

[8]https://www.electronics-lab.com/project/getting-started-esp32/
[9]https://github.com/me-no-dev/arduino-esp32fs-plugin/releases/

- NTPClient forked por Taranis[10], que facilita a obtenção da data e hora de rede;
- ESPAsyncWebServer[11] e Async TCP[12], que em conjunto facilitam a implementação de um servidor *Web* assíncrono no ESP32.

O *download* das bibliotecas é por meio de repositórios no GitHub em: Code > Download ZIP e deve ser salvo no diretório de bibliotecas padrão do Arduino, normalmente em: C:\Users\<Nome>\Documents\Arduino\libraries. A instalação é finalizada no próprio Arduino IDE, em: Sketch > Incluir Biblioteca > Adicionar Biblioteca .ZIP e na janela aberta deve-se procurar os arquivos no diretório em que foram baixados, selecionar o desejado e clicar em ”Abrir”.

Transmissor

O transmissor, componente do sistema responsável pela coleta e envio dos dados, necessita de uma conexão entre o módulo de sensor DHT11 e a placa WiFi LoRa 32(V2), como mostrado na Figura 3.3.

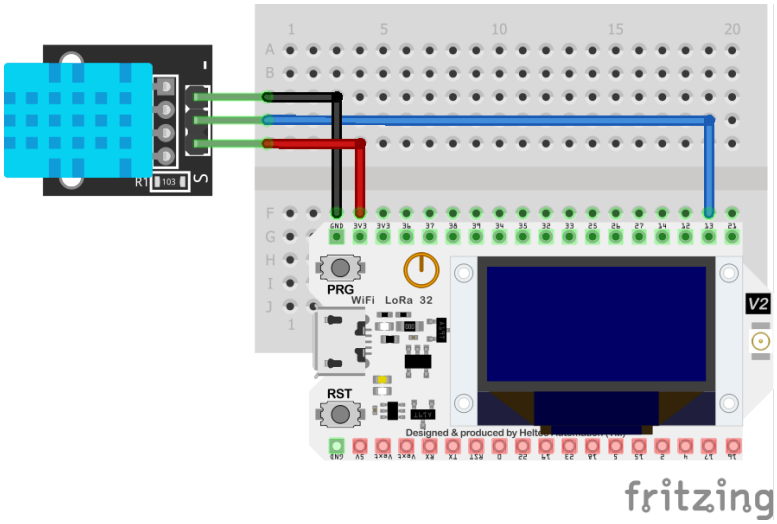


Figura 3.3 – Diagrama esquemático do transmissor

O código é iniciado com a declaração das bibliotecas para se trabalhar com o LoRa, bibliotecas para o *display* OLED e a biblioteca para o sensor DHT11. Em seguida são definidos os pinos utilizados pelo *chip* LoRa; a banda, no caso em análise é 915 MHz, frequência destinada a América; os pinos OLED e as dimensões da tela; o pino usado pelo módulo do sensor e qual sensor, ou seja, DHT11. As estruturas seguintes são: uma instância para o sensor com o pino utilizado e o seu tipo; a declaração de variáveis para a contagem de pacotes enviados e os valores coletados de temperatura e umidade; e a instância para o *display* com as dimensões já definidas.

O próximo passo é a criação das funções auxiliares: startOLED() e startLoRa(), funções padrões para a inicialização do *display* OLED e do módulo LoRa, respectivamente;

[10]https://github.com/taranais/NTPClient
[11]https://github.com/me-no-dev/ESPAsyncWebServer
[12]https://github.com/me-no-dev/AsyncTCP

getReadings(), usada para a obtenção dos valores de temperatura e umidade a partir do DHT11; e sendReadings(), onde ocorre o envio dos dados obtidos da função anterior organizando-os em uma única *string* denominada de LoRaMessage contendo o contador readingID, temperatura e umidade separados pelos marcadores "/" e "&". Além do envio dos dados, a função também os organiza para a exibição no *display*.

Há duas funções principais no escopo do código: setup() e loop(). Na função setup() são inicializados o *Serial Monitor* com *baud rate* de 115200, o DHT11, o OLED e LoRa através das funções startOLED() e startLoRa(), respectivamente. Na função loop() são chamadas as funções getReadings(), sendReadings() e delay(), esta última com 10000 milissegundos como parâmetro para o envio de cada pacote.

Receptor

O componente receptor coleta os dados enviados pelo transmissor e os exibe, juntamente ao tempo em que foi recebido e o RSSI em uma página *Web*. São necessários três arquivos diferentes: o código para a placa, o código HTML para a construção da página e a imagem utilizada que devem estar organizados como mostrado na Figura 3.4.

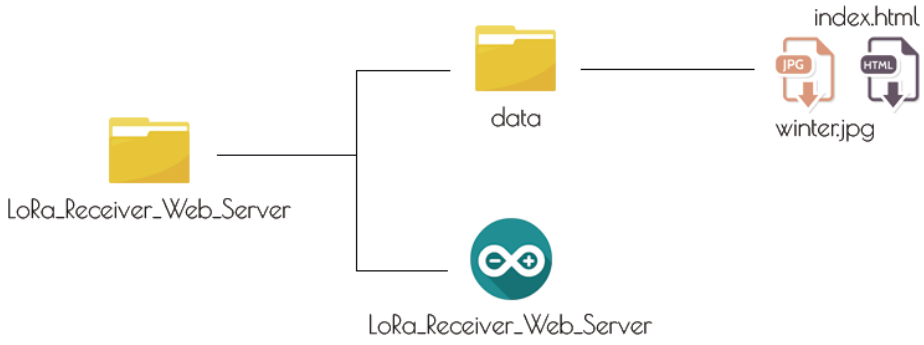


Figura 3.4 – Estrutura de organização de arquivos do receptor

Fonte: Adaptado de [44]

O arquivo HTML nomeado por index.html é composto por três elementos: o código HTML que cria a estrutura básica da página; trechos de estilo em CSS delimitado pelas *tags* <style></style>; e o trecho em Javascript, delimitado por <script></script>, responsável pela atualização dos campos a cada novo pacote recebido.

O código responsável pela recepção, assim como o anterior, é iniciado com a declaração das bibliotecas para execução com o LoRa e com o WiFi, bibliotecas para o *display* OLED e as bibliotecas para obter o tempo do NTP Server. Em seguida, são definidos os pinos utilizados pelo *chip* LoRa, a banda, os pinos do OLED e suas dimensões. As estruturas seguintes são campos que devem ser preenchidos com as credenciais de rede a ser usada, a definição do NTP Client e a declaração de variáveis para o armazenamento de data, hora e variáveis para o armazenamento dos dados recebidos do transmissor; a criação de um objeto AsyncWebServer na porta 80 e a instância para o *display* com as dimensões já definidas.

O próximo passo é a criação das funções auxiliares: startOLED() e startLoRa(), supracitadas na seção referente ao transmissor; processor(), usada para enviar os valores recebidos para os campos reservados no arquivo HTML; connectWiFi(), que usa as credenciais de rede para conectar a placa à rede local , também inicia o servidor e imprime o endereço de IP para acessar a página; getLoRaData(), onde ocorre a leitura e separação dos dados recebidos usando como mecanismo os marcadores ”/” e ”&” definidos no código do transmissor; e getTimeStamp(), utilizado para a obtenção e organização de data e hora do NTP Client.

Para finalizar, há as funções setup(), onde é inicializado o *Serial Monitor* com *baud rate* de 115200, OLED e LoRa através das funções startOLED() e startLoRa(), respectivamente; a inicialização do sistema SPIFFS juntamente com a implementação da rotina do servidor assim como sua inicialização e de um NTP Client com a compensação de fuso horário, no caso do Brasil, -3; e loop(), onde ocorre uma verificação se existem pacotes LoRa disponíveis, se sim, são chamadas as funções getLoRaData() e getTimeStamp().

Execução dos códigos

Para a execução do código do transmissor basta conectar a placa ao computador, confirmar a seleção da placa (Heltec WiFi LoRa 32(V2)), da porta e clicar em ”carregar”. O resultado pode ser observado tanto no *display* OLED quanto no *Monitor Serial*. O mecanismo para o receptor é um pouco diferente. Primeiro, deve-se realizar o *upload* da pasta *data* em: Ferramentas > ESP32 Sketch Data Upload. Após isso, o código deve ser enviado para a placa assim como o código do transmissor. Por fim, deve-se abrir o *Monitor Serial* para a obtenção do endereço de IP para acessar a página de monitoramento, que pode ser visualizado em qualquer navegador na mesma rede do ESP.

3.4.3 Servidor LwM2M com aplicação Web em Raspberry Zero para gerenciamento de clientes LwM2M

Lightweight M2M é um conjunto de protocolos definidos pela Open Mobile Alliance (OMA) para comunicação e gerenciamento de dispositivos máquina a máquina (M2M) ou IoT. Na seguinte prática é proposta a criação de um servidor LwM2M com uma interface de acesso aos dados dos clientes por uma aplicação web em um Raspberry Pi Zero W, criar dois clientes LwM2Ms e estabelecer desses clientes com o servidor. Ademais, a estrutura de funcionamento dessa prática pode ser observada na Figura 3.5.

O LwM2M tem algumas implementações de código aberto, nessa prática serão utilizados os ambientes Eclipse Leshan^{TM} [13] e Eclipse Wakaama[14]. O primeiro implementa tanto Cliente LwM2M quanto Servidor LwM2M em Java, e além disso, tem incluso uma aplicação Web para gerenciamentos dos dispositivos, objetos e instancias dos clientes. Já o segundo, oferece os recursos necessários para implementar cliente e/ou servidor em C.

[13]https://github.com/eclipse/leshan
[14]https://github.com/eclipse/wakaama

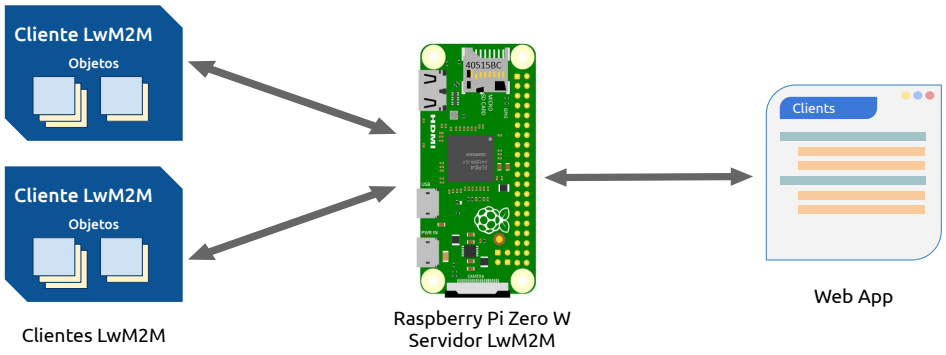


Figura 3.5 – Arquitetura de implementação

Requisitos necessários para a prática

- Raspberry Pi Zero W: Nesse dispositivo, o Servidor LwM2M e o servidor Web para hospedar a aplicação serão criados. Serão necessários os dispositivos de entrada e saída de um computador padrão (*mouse*, teclado e monitor).
- Computador pessoal Desktop/notebook: dedicado a execução dos clientes LwM2Ms e acesso da interface Web.
- Ponto de acesso Wi-Fi: Necessário para criar uma rede local entre os dois dispositivos.

Preparando o Raspberry Pi Zero W

1. é necessário um cartão microSD com memória igual ou superior a 8 Gb, ele será a memória secundária.
2. Instalar o Raspberry Pi OS[15] ou similar.
   - Baixar a imagem do Sistema Operacional (SO).
   - Gravar a imagem no cartão microSD.
   - Inserir o microSD no Raspberry, conectar os periféricos (monitor, mouse e teclado) e a fonte de alimentação.
   - O dispositivo iniciará, e quando requerido, inserir o usuário e senha. Até o momento o padrão é "pi"e "raspberry", respectivamente.
3. A fim de executar os *scripts* em Java da Eclipse Leshan é necessário instalar a versão 8 do java jdk, pois a versão instalada por padrão é a 11, entretanto, não é suportada pela arquitetura da placa.
   - Instalar o java jdk-8 pelo terminal executando o seguinte comando:

```
$ sudo apt—get install openjdk—8—jdk
```

[15] https://www.raspberrypi.org/downloads/raspberry-pi-os/

- Definir manualmente a versão padrão do Java, executando o comando:

```
$ sudo update−alternatives −−config
```

Então, é necessário definir a versão 8 em modo manual.

- Teste do Java:

```
$ java −version
```

- é preciso instalar o gerenciador de projetos Apache Maven[16], uma ferramenta usada para construir e gerenciar qualquer projeto baseado em Java.

```
$ sudo apt−get −y install maven
```

- Instalar o sistema de controle de versões Git:

```
$ sudo apt−get install git
```

Preparando o Desktop/Notebook

1. Para executar um cliente usando Leshan:

- Instalar o java jdk pelo terminal executando o seguinte comando:

```
$ sudo apt−get install openjdk− XX−jdk
```

substitua XX pela versão de preferência.

- é preciso instalar o gerenciador de projetos Apache Maven, uma ferramenta usada para construir e gerenciar qualquer projeto baseado em Java.

```
$ sudo apt−get −y install maven
```

- Instalar o sistema de controle de versões Git:

```
$ sudo apt−get install git
```

- Instalar compilador de C/C++ e CMmake[17]:

```
$ sudo apt−get install g++
```

```
$ sudo apt−get install cmake
```

Executar o servidor LwM2M e Web App no Raspberry Pi Zero:

- Obter o Eclipse Leshan:

```
$ git clone https://github.com/eclipse/leshan.git
```

[16]http://maven.apache.org/
[17]https://cmake.org/

O Leshan fornece uma demonstração de um servidor, que além de criar o servidor LwM2M, também cria um servidor Web e hospeda uma aplicação para obter a lista de clientes conectados e interagir com os objetos e recursos. Para executar o servidor LwM2M de demonstração da biblioteca, basta utilizar os seguintes comandos na pasta raiz do Leshan:

- $ mvn clean install
- $ java -jar leshan-server-demo/target/leshan-server-demo-*-SNAPSHOT-jar-with-dependencies.jar
- O servidor LwM2M fica disponível no endereço:
  `coap://raspberrylocalhost:5683`
- O servidor Web fica disponível no endereço:
  `http://raspberrylocalhostt:8080`
- Para obter o endereço local (localhost) da raspberry, execute no terminal do Desktop/Notebook:
  $ ping raspberrypi.local

Feito isso, é necessário criar clientes LwM2M no Desktop/Notebook ou similar, e registrar os clientes no servidor LwM2M em execução. E para isso, pode-se usar o Eclipse Leshan, Eclipse Wakaama ou biblioteca similar que ofereça esse recurso.

Cliente LwM2M, usando Eclipse Leshan

1. Obter o Eclipse Leshan:
   $ git clone https://github.com/eclipse/leshan.git
2. Configurando o cliente LwM2M:
   Com um editor de texto ou IDE, abrir o arquivo LeshanClientDemo.java no diretório:
   - /leshan/leshan-client-demo/src/main/java/org/eclipse/leshan/client/demo

   Nesse exemplo o nome(endpoint) do cliente, é definido como o nome de usuário do sistema operacional do Desktop/Notebook, caso não seja possível obter o usuário é utilizado o DEFAULT_ENDPOINT = "LeshanClientDemo". Para definir manualmente basta atribuir uma *string* com o nome desejado a variável *endpoint* após o trecho de código que faz a definição e atribuição a *endpoint*, para essa prática foi definido como "cliente_leshan"

   ServerURI é o endereço do servidor LwM2M. Por padrão, o algoritmo tenta estabelecer conexão com um servidor "127.0.0.1", ou seja, no mesmo dispositivo, então para evitar essa tentativa é preciso alterar as *strings* coap://localhost:âXX e âXX coaps://localhost:âXX no trecho de código que declara a variável serverURI, substituindo o "localhost"pelo endereço IP do servidor LwM2M, que é o mesmo do Raspberry.

3. Executar o cliente LwM2M:
   Usar os seguintes comandos na pasta raiz do Leshan:

   - $ mvn clean install
   - $ java -jar leshan-client-demo/target/leshan-client-demo-*-SNAPSHOT-jar-with-dependencies.jar

O cliente demo cria inicialmente quatro objetos por padrão, LwM2M *Server* (Id: 1), *Device* (Id: 3), *Location* (Id: 6) e *Temperature* (Id: 3303) e, além disso, apresenta uma interface de linha de comando básica no terminal de execução, capaz de criar, excluir e modificar os objetos e recursos. Os modelos de objetos disponíveis estão definidos em "public final static String[] modelPaths = new String[] {...}". Após o registro do cliente LwM2M no servidor LwM2M, os dados dos objetos são atualizados os dados periodicamente.

Cliente LwM2M, usando Eclipse Wakaama

Wakaama não é uma biblioteca, mas arquivos a serem construídos com um aplicativo, ele usa CMake > = 3. Veja no diretório [pasta raiz do wakaama] / examples / server / o arquivo CMakeLists.txt, para um exemplo de como incluí-lo em um projeto. Além disso, dispõe de várias opções de compilação, oferece um cliente para testes que implementa 9 objetos, *Security* (id: 0), Server (id: 1), Access Control (id: 2), Device (id: 3), contendo valores codificados do Cliente Exemplo LWM2M do Apêndice E da Especificação Técnica LWM2M [45], Connectivity Monitoring (id: 4), Firmware Update (id: 5), Location Object (id: 6), Connectivity Statistics Object (id: 7) e Test (id: 31024), e apresenta uma interface de linha de comando básica no terminal de execução, basta digitar ’help’ para obter uma lista de comandos suportados.

- Obter Eclipse Wakaama:
  $ git clone https://github.com/eclipse/wakaama.git

O exemplo de cliente está localizado em [pasta raiz do wakaama]/examples/client/, entretanto, para executar de forma correta é preciso fazer modificações. Então, com um editor de texto, abrir o arquivo lwm2mcliente.c. No trecho de código: #define DEFAULT_SERVER_IPV4 "127.0.0.1", é preciso alterar o endereço local para o correspondente ao servidor LwM2M. Na função principal, pode-se alterar o nome (endpoint) do cliente no trecho: char * name = ”testlwm2mclient”, para a prática foi definido o endpoint como "cliente_wakaama". Além disso, é fundamental configurar a conexão IPv4/IPv6, por padrão a configuração é IPv6, então no trecho: data.addressFamily = AF_INET6; é preciso substituir AF_INET6 por AF_INET, para definir a conexão como IPv4.

Feito isso, basta compilar e executar o cliente. No diretório [pasta raiz do wakaama]/examples/client/, executar os seguintes comandos:
$ cmake [pasta raiz do wakaama]/examples/client/

```
$ make
$ ./lwm2mclient
```

Web App

Ao executar o servidor LwM2M, o servidor Web também inicia, então a aplicação Web para visualizar/modificar os objetos dos clientes fica disponível em: `http://raspberrylocalhost:8080/`, assim como na Figura

3.4.4 Análise de Tráfego do Protocolo MQTT-SN

As seções anteriores trouxeram uma visão geral sobre as bases da IoT, em especial do protocolo MQTT-SN. Esta seção, por outro lado, tem como objetivo mostrar ao leitor na prática alguns dos conceitos vistos anteriormente, dando uma ênfase maior na análise de tráfego do protocolo MQTT-SN. Para realizar os experimentos será utilizado o microcomputador *single-board* Raspberry Pi Zero com sistema operacional Raspbian[18], em sua versão com *desktop*.

Com o intuito de se obter um cenário mais próximo de uma rede IoT real, em que grande parte das "coisas sofrem com severas restrições de poder computacional, no experimento será utilizado o microcontrolador NodeMCU ESP32. Toda a análise de tráfego será feita com auxílio da ferramenta de análise de rede Wireshark[19].

Para a realização dos experimentos o primeiro passo é a instalação do analisador de protocolos Wireshark. No Raspbian, abra o terminal e digite o seguinte comando:

```
$ sudo apt—get install wireshark
```

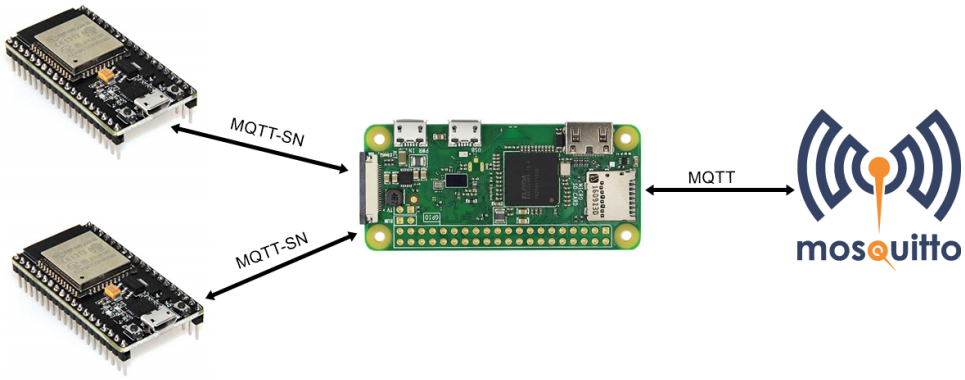
Após a instalação do Wireshark, para o experimento, é necessário a instalação de um *gateway* e um servidor *broker* (disponível para download no GitHub[20]).

O Raspberry Pi Zero será utilizado como *gateway* e como *broker* MQTT-SN, e será também o responsável pela captura do tráfego, auxiliado pelo analisador de protocolos Wireshark. Neste experimento será analisado o processo de comunicação MQTT entre dois ESP32, simulando um ambiente de IoT, como pode ser observado na Figura 3.6.

Após a configuração do *gateway* e do *broker* no Raspberry Pi, o servidor já estará rodando automaticamente na porta TCP padrão, 1883. O próximo passo é a preparação dos clientes. Após a instalação do servidor *broker* foi realizada a preparação dos clientes, para isto a biblioteca Arduino MqttSnClient (disponível para download no GitHub[21]) foi instalada nos NodeMCUs. No Arduino IDE, após a instalação da biblioteca, foram criados códigos para o experimento do Protocolo MQTT-SN. O código pode ser encontrado no link: `https://github.com/nuclicufc/minicursoSBrT2020_MQTTSN`. Altere o IP na linha 14 e a porta na linha 15 de acordo com a faixa de endereços IP da sua rede, por

[18]https://www.raspberrypi.org/downloads/raspbian/
[19]https://www.wireshark.org/
[20]https://github.com/eclipse/paho.mqtt-sn.embedded-c/tree/master/MQTTSNGateway
[21]https://github.com/S3ler/arduino-mqtt-sn-client

Figura 3.6 – Cenário simulando um ambiente de IoT [Elaborada pelos autores].



exemplo, para uma rede que utiliza endereços na faixa de “192.168.0.x” é possível utilizar os valores mostrados no Algoritmo 2.

Algorithm 1 Linhas 14 e 15 do exemplo do código.

```
IPAddress gatewayIPAddress(192, 168, 0, 100);
uint16_t localUdpPort = 8888;
```

O endereço do “server”, na linha 14, deve ser substituído pelo endereço IP atribuído ao *gateway*. Feito isso, inicie a captura de pacotes no Wireshark (filtrando por “mqttt"), carregue as alterações para a placa ESP32. Neste momento, já será possível observar alguns pacotes no Wireshark, como é mostrado na Figura 3.7.

Figura 3.7 – Captura de pacotes após a conexão do ESP32 na Rede [Elaborada pelos autores].

No.	Time	Source	Destination	Protocol	Length	Info
13…	40.96324…	137.135.83.217	192.168.1.10	MQTT	71	Subscribe Ack (id=512)
15…	44.56853…	192.168.1.10	137.135.83.217	MQTT	93	Connect Command
16…	44.73265…	137.135.83.217	192.168.1.10	MQTT	70	Connect Ack
47…	153.6234…	192.168.1.10	137.135.83.217	MQTT	93	Connect Command
48…	153.7944…	137.135.83.217	192.168.1.10	MQTT	70	Connect Ack
48…	154.1223…	192.168.1.10	137.135.83.217	MQTT	99	Subscribe Request (id=512) [ESP8266/WiFi/Udp/subscribe]
48…	154.2884…	137.135.83.217	192.168.1.10	MQTT	71	Subscribe Ack (id=512)
56…	172.2607…	192.168.1.10	137.135.83.217	MQTT	93	Connect Command
56…	172.4395…	137.135.83.217	192.168.1.10	MQTT	70	Connect Ack
63…	216.5407…	192.168.1.10	137.135.83.217	MQTT	74	Publish Message []

O primeiro pacote MQTT capturado pelo Wireshark é a mensagem de solicitação de conexão. Analisando melhor esta solicitação podemos perceber algumas informações relevantes para a comunicação. A primeira informação importante que este pacote fornece é a versão do protocolo na qual a comunicação se estabelecerá. Como pode ser visto em “Version: 4”, na Figura 3.8, o cliente 1 está solicitando comunicação com a versão 3.1.1 do protocolo MQTT, que conforme especificado em [46] é representado pelo número 4, o cliente 2 também esta utilizando a versão 3.1.1, como visto na Figura 3.9. A solicitação de conexão deve conter, ainda, outras informações importantes, como usuário, senha, *Client ID*, *Keep Alive*, entre outras. No exemplo não foi utilizado usuário e senha, entretanto, em uma implementação prática a utilização destes campos é extremamente recomendada. Ainda nas Figuras 3.8 e 3.9, é possível ver o campo *Client ID*, onde é informada uma identificação para o dispositivo solicitante da conexão, (neste caso MqttSnCliente1 e MqttSnCliente2, respectivamente), para isto foi utilizado o protocolo mDNS, esta deve ser única para que não ocorram conflitos.

Figura 3.8 – Pacote de solicitação de conexão do ESP32 - Cliente 2 [Elaborada pelos autores].

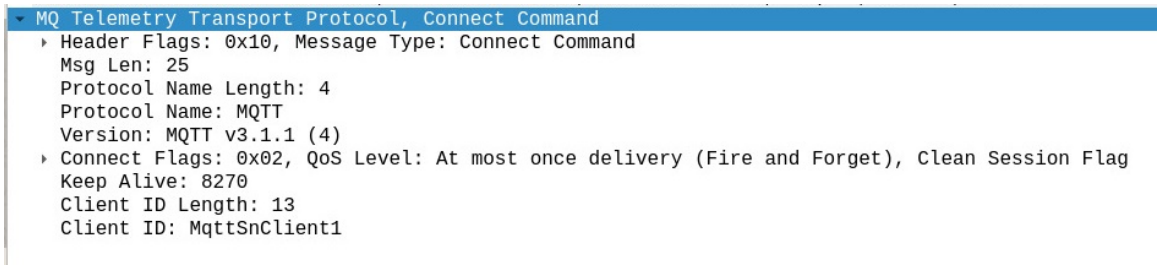
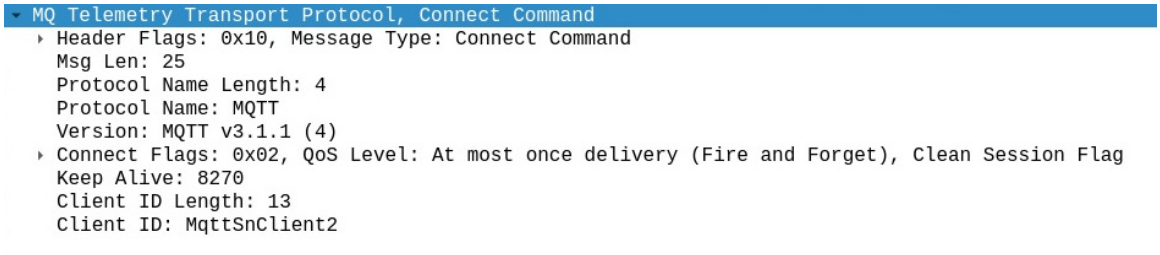


Figura 3.9 – Pacote de solicitação de conexão do ESP32 - Cliente 1 [Elaborada pelos autores].



Pode-se observar nas Figuras 3.8 e 3.9, que o nível de QoS utilizado foi o 0 (zero). Este nível pode ser alterado na linha 27 do código, a fim de analisar o tráfego de rede para demostração de cada nível de QoS. Para este trabalho, a nível de demostração, utilizamos só o QoS nível 0 (zero).

Algorithm 2 Linha 27 do exemplo código.

```
int8_t qos = 0; //QoS MqttSN
```

Nas Figuras 3.10 e 3.11, no topo da próxima página, pode-se observar o Cliente 2, que estava escrito no tópico “ESP8266/WiFi/Udp/subscribe”, configurado previamente no código, recebendo as informações publicadas pelo Cliente 1 no tópico.

Figura 3.10 – Tela de recebimento da mensagem - Wireshark [Elaborada pelos autores].

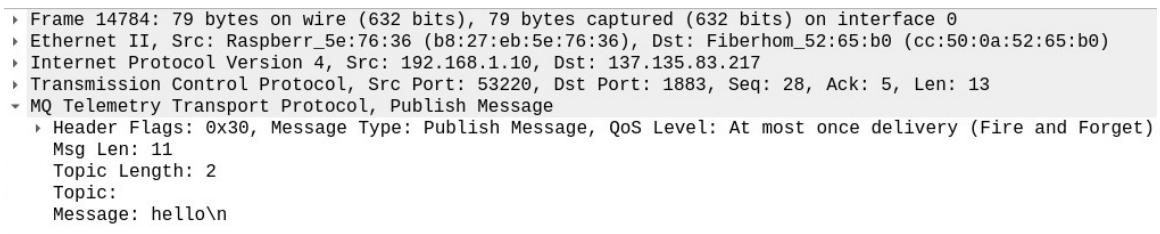
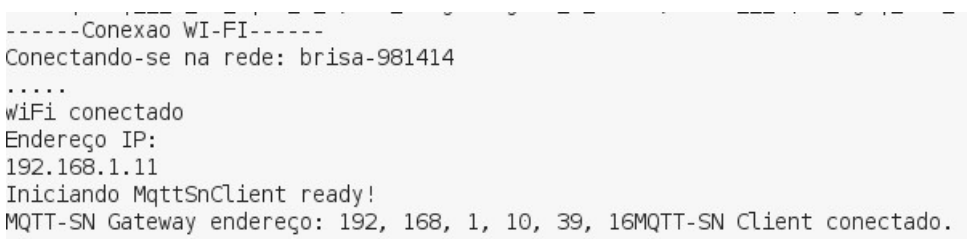


Figura 3.11 – Tela de recebimento da mensagem - IDE Arduino [Elaborada pelos autores].



3.5 Conclusão

O desenvolvimento de um projeto de IoT envolve o uso de diversas tecnologias, que variam conforme os critérios e necessidades de cada aplicação. O Raspberry Pi 4, o Raspberry Pi Zero, o NodeMCU ESP32 e a DragonBoard são plataformas de desenvolvimento lançadas nos últimos anos com grande potencial de aplicação em diversas áreas da IoT, tendo como principais diferenciais seu poder computacional embarcado bem como a flexibilidade para adaptar-se a diversos contextos.

Seguindo a mesma necessidade de fácil adaptação e flexibilidade, característica intrínseca da IoT, muitos protocolos de comunicação precisaram ser desenvolvidos ou adaptados para o uso nesse novo cenário, incluindo protocolos de aplicação, de descoberta de serviços ou de infraestrutura. As práticas desenvolvidas neste trabalho demonstraram o uso dos protocolos ESP-Now, LoRaWAN, LwM2M e MQTT-SN para o estabelecimento de comunicação entre as plataformas anteriormente apresentadas. Tais rotinas de comunicação podem ser utilizadas e/ou adaptadas para aplicação em diversos cenários, como automação residencial ou *Smart Home*, *Healthcare*, *Smart Grid* e outros.

Referências Bibliográficas

[1] Andrew Whitmore, Anurag Agarwal, and Li Da Xu. The internet of things: A survey of topics and trends. *Information systems frontiers*, 17(2):261–274, 2015.

[2] Min Chen, Jiafu Wan, and Fang Li. Machine-to-machine communications: Architectures, standards and applications. *Ksii transactions on internet & information systems*, 6(2), 2012.

[3] Ammar Rayes and Samer Salam. *Internet of Things From Hype to Reality*. 2019.

[4] Jeretta Horn Nord, Alex Koohang, and Joanna Paliszkiewicz. The internet of things: Review and theoretical framework. *Expert Systems with Applications*, 133:97–108, 2019.

[5] Jonathan Tournier, François Lesueur, Frédéric Le Mouël, Laurent Guyon, and Hicham Ben-Hassine. A survey of iot protocols and their security issues through the lens of a generic iot stack. *Internet of Things*, page 100264, 2020.

[6] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, 17(4):2347–2376, 2015.

[7] Syllas Rangel C. Magalh aes, Victória Tomá Oliveira, Francisco Evangelista N. Filho, Magdiel Campelo Alves de Sousa, Jermana Lopes de Moraes, and Wendley S. Silva. Monitoramento lógico e fásico do tâfego em redes de internet das coisas. *Livro de Minicursos SBRT 2018*, (4):120–151, 2018.

[8] Bruno P Santos, Lucas AM Silva, CSFS Celes, João B Borges, Bruna S Peres Neto, Marcos Augusto M Vieira, Luiz Filipe M Vieira, Olga N Goussevskaia, and A Loureiro. Internet das coisas: da teoria à prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuıdos*, 31, 2016.

[9] Umit Isikdag. Internet of things: Single-board computers. In *Enhanced Building Information Models*, pages 43–53. Springer, 2015.

[10] Sarthak Jain, Anant Vaibhav, and Lovely Goyal. Raspberry pi based interactive home automation system through e-mail. In *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, pages 277–280. IEEE, 2014.

[11] Sérgio de Oliveira. *Internet das coisas com ESP8266, Arduino e Raspberry PI*. Novatec Editora, 2017.

[12] Gagandeep Singh Nagpal, Gagandeep Singh, Jappreet Singh, and Nishant Yadav. Facial detection and recognition using opencv on raspberry pi zero. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 945–950. IEEE, 2018.

[13] D Sri Sai Mahesh, T Maneesh Reddy, A Sai Yaswanth, C Joshitha, and S Sudarshan Reddy. Facial detection and recognition system on raspberry pi with enhanced security. In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pages 1–5. IEEE, 2020.

[14] Bahast Ali and Xiaochun Cheng. Security solution based on raspberry pi and iot. In *International Symposium on Cyberspace Safety and Security*, pages 162–171. Springer, 2019.

[15] Chinmay Bepery, Sudipto Baral, Animesh Khashkel, and Farhad Hossain. Advanced home automation system using raspberry-pi and arduino. *International Journal of Computer Science and Engineering*, 8(8):1–10, 2019.

[16] Shubhang Khattar, Anisha Sachdeva, Rishi Kumar, and Richa Gupta. Smart home with virtual assistant using raspberry pi. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 576–579. IEEE, 2019.

[17] Mani Dheeraj Mudaliar and N Sivakumar. Iot based real time energy monitoring system using raspberry pi. *Internet of Things*, page 100292, 2020.

[18] Abu Hassan Abdullah, Sukhairi Sudin, Muhammad Ikmal Mat Ajit, Fathinul Sahir Ahmad Saad, Kamarulzaman Kamaruddin, Firdaus Ghazali, Zahari Awang Ahmad, and Muhammad Aizat Abu Bakar. Development of esp32-based wi-fi electronic nose system for monitoring lpg leakage at gas cylinder refurbish plant. In *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, pages 1–5. IEEE, 2018.

[19] S Espressif. Esp32 datasheet. *IotYbased microcontroller*, 2017.

[20] Alexander Maier, Andrew Sharp, and Yuriy Vagapov. Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things. In *2017 Internet Technologies and Applications (ITA)*, pages 143–148. IEEE, 2017.

[21] Shatadru Bipasha Biswas and M Tariq Iqbal. Solar water pumping system control using a low cost esp32 microcontroller. In *2018 IEEE Canadian conference on electrical & computer engineering (CCECE)*, pages 1–5. IEEE, 2018.

[22] Borade Samar Sarjerao and Amara Prakasarao. A low cost smart pollution measurement system using rest api and esp32. In *2018 3rd International Conference for Convergence in Technology (I2CT)*, pages 1–5. IEEE, 2018.

[23] Ioana-Maria Culic and Alexandru Radovici. Extending the wyliodrin platform for windows 10 iot core. In *2016 15th RoEduNet Conference: Networking in Education and Research*, pages 1–5. IEEE, 2016.

[24] Leidy Alexandra Sarmiento e Cuervo Victor Manuel Martinez Moncayo, Hector I Reyes e Ortiz. Sistema de monitoramento de espectro de baixo custo baseado em dragonboard 410c e dongle rtl-sdr 2832u. *Scientia et technica*, 23:34–40, 2018.

[25] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015.

[26] Syllas Rangel C. Magalh aes, Victória Tomá Oliveira, Francisco Evangelista N. Filho, and Wendley S. Silva. Monitoramento de trÃifego em redes de internet das coisas. *Encontro Unificado de ComputaÃ§Ã£o*, 10(11):831–855, 2017.

[27] Marcelo Maestrelli e Thiago Maia Celso Tinoco. A internet das coisas. `https://www.gta.ufrj.br/ensino/eel878/redes1-2016-1/16_1_2/IOT/iot.html?`, 2016.

[28] Sistemas de comunicação de voz de baixo custo descentralizado baseado em esp-now para edifícios. In *Simpósio Internacional de Engenharia Elétrica e Eletrônica 2019 (ISEE)*.

[29] Tu Ngoc Hoang, Su-Tran Van, and BD Nguyen. Esp-now based decentralized low cost voice communication systems for buildings. In *2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, pages 108–112. IEEE, 2019.

[30] Suhas Rao, Devaiah Chendanda, Chetan Deshpande, and Vishwas Lakkundi. Implementing lwm2m in constrained iot devices. In *2015 IEEE Conference on Wireless Sensors (ICWiSe)*, pages 52–57. IEEE, 2015.

[31] David Tracey and Cormac Sreenan. Oma lwm2m in a holistic architecture for the internet of things. In *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, pages 198–203. IEEE, 2017.

[32] Suhas Rao, Devaiah Chendanda, Chetan Deshpande, and Vishwas Lakkundi. Implementing lwm2m in constrained iot devices. In *2015 IEEE Conference on Wireless Sensors (ICWiSe)*, pages 52–57. IEEE, 2015.

[33] Andy Stanford-Clark and Hong Linh Truong. Mqtt for sensor networks (mqtt-sn) protocol specification. `http://www.mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf`, 2013.

[34] M. Krochmal S. Cheshire. Multicast dns. *Apple Inc.*, 2013.

[35] M. Krochmal S. Cheshire. Dns-based service discovery. *Apple Inc.*, 2013.

[36] Elvio João Leonardo and Ailton Akira Shinoda. Revisão de protocolos de acesso ao meio em redes ad-hoc sem fio. *Semina: Ciências Exatas e Tecnológicas*, 25(2):172–182, 2004.

[37] Behrouz A Forouzan. *Comunicação de dados e redes de computadores*. AMGH Editora, 2009.

[38] Jo ao Wilson Vieira Rocha. Redes wlan de alta velocidade i: Características. `http://www.teleco.com.br/tutoriais/tutorialredeswlanI/default.asp`, 2006. [Online; acesso em 16-Outubro-2018].

[39] Luis Guilherme Uzeda Garcia. Redes 802-11 (camada de enlace). `https://www.gta.ufrj.br/grad/01_2/802-mac/R802_11.htm`, 2018. [Online; acesso em 11-Outubro-2018].

[40] Jetmir Haxhibeqiri, Eli De Poorter, Ingrid Moerman, and Jeroen Hoebeke. A survey of lorawan for iot: From technology to application. *Sensors*, 18(11):3995, 2018.

[41] Aloÿs Augustin, Jiazi Yi, Thomas Clausen, and William Mark Townsley. A study of lora: Long range & low power networks for the internet of things. *Sensors*, 16(9):1466, 2016.

[42] S Espressif. Esp-now user guide. 2016.

[43] Esp-now with esp8266: Send data to multiple boards (one-to-many)). `https://randomnerdtutorials.com/esp-now-one-to-many-esp8266-nodemcu/`, 2020.

[44] Esp32 lora sensor monitoring with web server (long range communication). `https://randomnerdtutorials.com/esp32-lora-sensor-web-server/`, 2019.

[45] Open Mobile Alliance. Lightweight machine to machine technical especification. http://openmobilealliance.org/release/LightweightM2M/V1_0_2-20180209-A/OMA-TS-LightweightM2M-V1_0_2-20180209-A.pdf, 2018. [Online; acesso em 01-Outubro-2018].

[46] OASIS Standard. Mqtt version 3.1.1. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html, 2014.

CAPÍTULO

4 Fundamentos de Testes de Compatibilidade Eletromagnética Conduzido e Irradiado

Andreia Aparecida de Castro Alves (Rohde & Schwarz do Brasil Ltda.)

Introdução

Atualmente, os dispositivos eletrônicos desempenham um papel importante em quase todos os aspectos da vida cotidiana, tornando-se praticamente indispensáveis nos negócios, na indústria, na medicina, na educação e até mesmo em casa. Todos esses aparelhos têm capacidade de interferir ou sofrer interferência de outros instrumentos eletrônicos. As interferências eletromagnéticas entre dispositivos podem ocorrer por meio de emissões irradiadas (pelo ar) ou conduzidas (via cabos) [1]. Os efeitos das interferências podem variar de perturbações relativamente mínimas, falha do dispositivo e até danos permanentes. Dessa forma, é essencial que sejam feitos esforços para identificar, medir, limitar ou eliminar interferências entre aparelhos.

Compatibilidade eletromagnética (EMC-*Electromagnetic compatibility*) é o termo usado para descrever a capacidade dos componentes eletrônicos funcionarem adequadamente em um ambiente eletromagnético definido. A importância dos testes de EMC deu origem a vários padrões e regulamentos governamentais, militares e industriais [2]. Em muitas partes do mundo, os produtos eletrônicos não podem ser comercializados sem antes demonstrarem a aderência aos padrões relevantes de EMC e, portanto, muitas empresas incorporaram os testes de EMC em seu ciclo de desenvolvimento de produtos.

Existem três elementos essenciais para um problema de EMC, conforme ilustrado na Figura 4.1. Há uma fonte de energia eletromagnética, um receptor e um caminho entre eles que acopla a energia da fonte interferente ao receptor. As fontes interferentes podem ser naturais, como raios, radiação solar, radiação cósmica e descargas eletrostáticas, ou fontes artificiais como transmissores de RF, fornos micro-ondas, instrumentos cirúrgicos, entre outros. O caminho pelo qual a energia eletromagnética é percorrido pode ser cabos, antenas,

aterramento comum, indutâncias e capacitâncias parasitas. Já nos receptores, podem ser incluidos todos os mais diversos tipos de equipamentos eletrônicos.

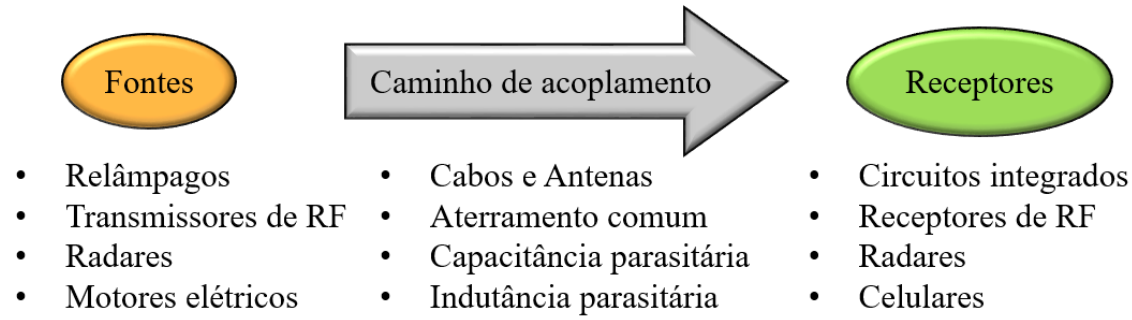


Figura 4.1 – Elementos de um problema de EMC.

Os padrões de EMC são definidos e redigidos por organizações e comitês internacionais, nacionais ou regionais. Eles definem os métodos de teste de EMC, especificam os limites e níveis mínimos para emissões elétricas e eletromagnéticas de produtos eletromecânicos e eletrônicos. Outro objetivo das normas e padrões é verificar a susceptibilidade dos produtos eletrônicos em relação aos sinais de RF, descargas atmosféricas e eletricidade estática. Além disso, os padrões EMC definem os níveis de potência de espúrios fora da faixa de frequência para garantir que não haja interferência na comunicação essencial por rádio (por exemplo, serviços de bombeiros ou rádio policial).

Uma revisão abrangente das normas internacionais é fornecida em [3]. O dispositivo deve atender aos requisitos definidos por organizações de padronização, como IEC, CISPR, ISO, IEEE, CENELEC, ETSI, FCC, ANSI , RTCA ou o comitê MIL-STD. A IEC61000-3-2 [4], refere-se a emissões de até 2 kHz, enquanto, as CISPR11 e EN55011 [5], definem limites de emissão na faixa de frequência de 9 kHz a 150 kHz para indução. A IEC61000-6-3 [6], considera as emissões de luz em aplicações industriais, comerciais e residenciais até 400 GHz, mas não atende às faixa de kHz. Além disso, a EN55022 [7], estabelece os limites de emissão de radiofrequência de 9 kHz a 400 GHz para equipamentos de tecnologia da informação.

Para colocar um produto eletrônico no mercado é necessário que o dispositivo cumpra seus respectivos padrões de EMC durante seu desenvolvimento, projeto e fabricação e, portanto, muitas empresas incorporaram esses testes em seu ciclo de desenvolvimento. Isso significa mais algumas condições de contorno para os projetistas de dispositivos eletro/eletrônicos.

Diversos estudos de EMC estão sendo apresentados na literatura com o objetivo de reduzir interferências eletromagnéticas, aumentar a imunidade dos equipamentos e reduzir custos e testes de certificação e pré-certificação [8, 9, 10, 11, 12, 13, 14, 15]. Recentemente, Chunyu Wu e colaboradores [8], analisaram e modelaram os níveis de emissão conduzida da fonte de alimentação CA e CC abaixo de 1 MHz em uma TV LED. Le Yang e colaboradores [9], desenvolveram uma técnica de previsão de emissão com base no princípio operacional dos analisadores de espectro EMC. Comparado com a literatura existente, o filtro IF e o detector de envelope foram modelados com uma maior precisão. Elmar Griese [10], propôs uma

nova tecnologia de interconexão híbrida elétrico/óptica capaz de reduzir as interferências eletromagnéticas e melhorar o desempenho das placas de circuitos impressos. Frédéric Pythoud e E. Tas [11], desenvolveram um protótipo para testes de imunidade irradiada. O aparelho foi usado para uma comparação Inter laboratorial e métodos de avaliações foram desenvolvidas para realizar uma proficiência de avaliação de teste de acordo com a norma ISO 13528 [16]. Hermann Nzalli e colaboradores [12], criaram um novo método baseado em técnicas guiadas por simulação para simplificar a avaliação da dependência da carga nos testes de imunidade para injeção direta de potência.

A investigação de problemas de EMC envolve a medição de formas de onda complexas que variam consideravelmente na amplitude e no tempo. Métodos de medição foram criados para fornecer resultados consistentes e repetíveis que, quando possível, têm alguma relação com a interferência causada na recepção. Os instrumentos de medição no domínio da frequência são baseados em projetos de receptor de rádio. Os circuitos de detecção e medição mais comuns empregam detectores de pico e quase-pico.

Os testes e medições de EMC são divididos em duas categorias: interferência eletromagnética (EMI *Electromagnetic interference*) e susceptibilidade eletromagnéticas (EMS *Electromagnetic susceptibility*). Para cada um desses cenários tem-se duas configurações distintas, teste conduzido e teste irradiado [17, 18]. O objetivo desse minicurso é apresentar as configurações, principalmente os equipamentos, que são recomendações na implementação de sistemas EMC. Serão descritos os cenários de testes EMI e EMS, conduzido e irradiado, bem como os ambientes de teste recomendados pelas normas.

Testes e medições EMC

Essa Seção apresenta as 4 principais configurações de teste EMC de pré-conformidade considerando um aparelho de telecomunicações como o dispositivo sob teste (DUT - *device under test*). Para cada *setup* serão ilustrados os principais equipamentos necessários. Os testes de pré-certificação podem impedir visitas repetidas a um laboratório de certificação completo, economizando recursos. Resultados limpos nas fases iniciais aumentam a chance do produto passar no teste formal desde a primeira vez. A Figura 4.2 apresenta de forma qualitativa, a complexidade e o custo de cada um dos 4 cenários. É possível concluir que os testes de EMI são sempre menos complexos em relação aos testes de EMS, e testes do tipo irradiado exigem sempre uma maior complexidade e o uso de câmaras anecóicas maiores [19].

Os instrumentos de pré-conformidade foram desenvolvidos para permitir que os engenheiros avaliem seu projeto em intervalos durante o processo de desenvolvimento. A medição comparativa é outra ferramenta importante - como quando um produto em uma variedade de unidades semelhantes é aprovado por um laboratório de teste reconhecido e usado como um “produto de ouro” com o qual os outros podem ser comparados. Esses truques deram origem a uma gama crescente de equipamentos de teste de pré-conformidade

para atender aos orçamentos da maioria das empresas. Assim, as empresas devem poder otimizar a aquisição de equipamentos de teste de EMC para atender às suas necessidades específicas.

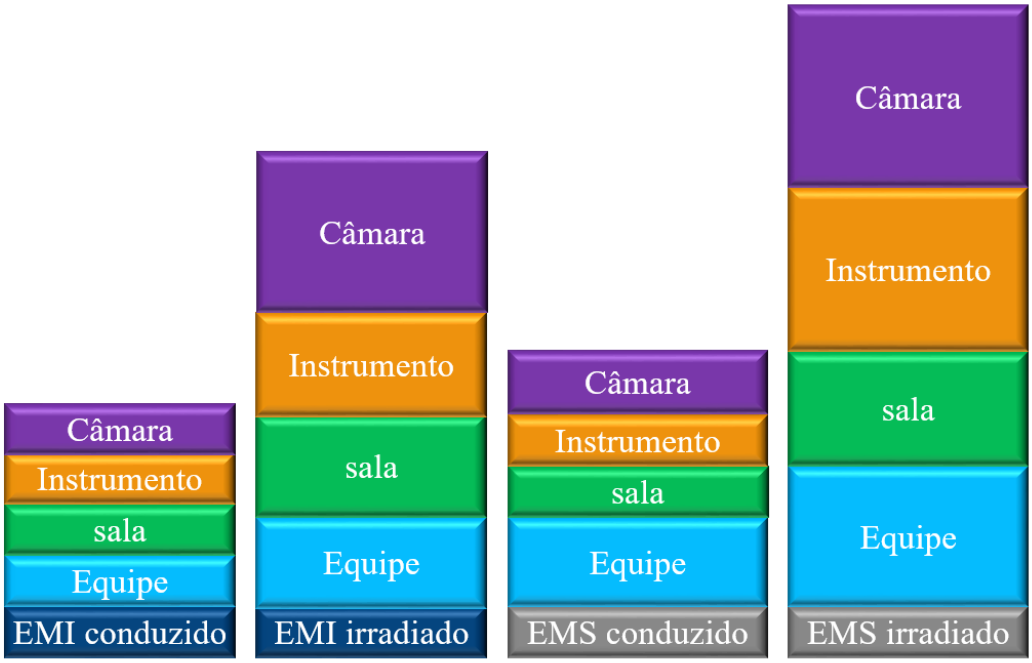


Figura 4.2 – Ilustração qualitativa da diferença dos testes de EMC, tomando como referência custo e complexidade.

4.0.1 EMI

Teste de emissão (também chamado de teste de interferência) envolve a medição de sinais eletromagnéticos (não intencionalmente) emitidos pelo DUT para determinar se essas emissões excedem os limites permitidos, potencialmente causando problemas para outros dispositivos próximos.

O teste de EMI difere da análise tradicional de espectro de várias maneiras, mas o diferenciador mais importante entre eles reside na natureza dos sinais que estão sendo medidos. Na análise tradicional de espectro, medimos sinais que o dispositivo em teste esta gerando. O principal objetivo do teste EMI é detectar, medir e localizar sinais cujos parâmetros básicos de RF, como frequência, nível, periodicidade etc, não são conhecidos ou conhecíveis a priori. Isso cria um problema óbvio em termos de otimizar a configuração de medição e evitar resultados imprecisos ou erros. Outro desafio é o desperdiço de tempo e recurso limitado ao procurar emissões espúrias que estão sendo geradas dentro de nossa configuração de medição e não pelo DUT. Por isso a necessidade de uma calibração confiável em todos os equipamentos, baseando-se sempre nas normas estabelecidas.

Uma falha na detecção e mitigação de emissões espúrias reais do DUT pode fazer com que o dispositivo falhe no teste de conformidade (demorado e caro) ou crie problemas sérios se o dispositivo for colocado em um ambiente “do mundo real”. A maioria dos problemas

envolvidos na detecção e quantificação precisa de emissões espúrias e indesejadas nos testes de EMI está relacionada ao nível imprevisível e/ou a mudança rápida dos sinais [10].

EMI conduzido

O *setup* comumente utilizado para uma medição de EMI conduzida é apresentado na Figura 4.3. O DUT é conectado a uma rede de estabilização de impedância de linha (LISN - *line impedance stabilization network*). A LISN é necessária para realizar uma medição de ruído de perturbação estável e reproduzível. Inserindo a LISN na linha de fonte de alimentação, a impedância da fonte de energia observada do lado do DUT é constante. Não há mudança de posição do DUT, mas o teste é realizado em cada uma das linhas de alimentação. Para fazer medições de EMI em cabos de dados que não possuem a LISN correta, o teste será realizado através de um transdutor diferente que pode ser uma sonda de corrente.

O analisador de espectro tem sido o instrumento tradicional de escolha da comunidade EMC. Houve uma mudança de analisadores de espectro para receptores EMI cujas arquiteturas são otimizadas para medições EMI. O equipamento EMI *test receiver*, é um analisador de espectro, configurado para realizar testes de conformidade ou pré-conformidade. A faixa de frequência observada nos testes é de 9 kHz até 30 MHz. Um software, por exemplo o R&S EMC32, pode ser utilizado para facilidar a vizualização dos resultados medidos e plotar curvas em comparação com as normas estabelecidas para o DUT em teste.

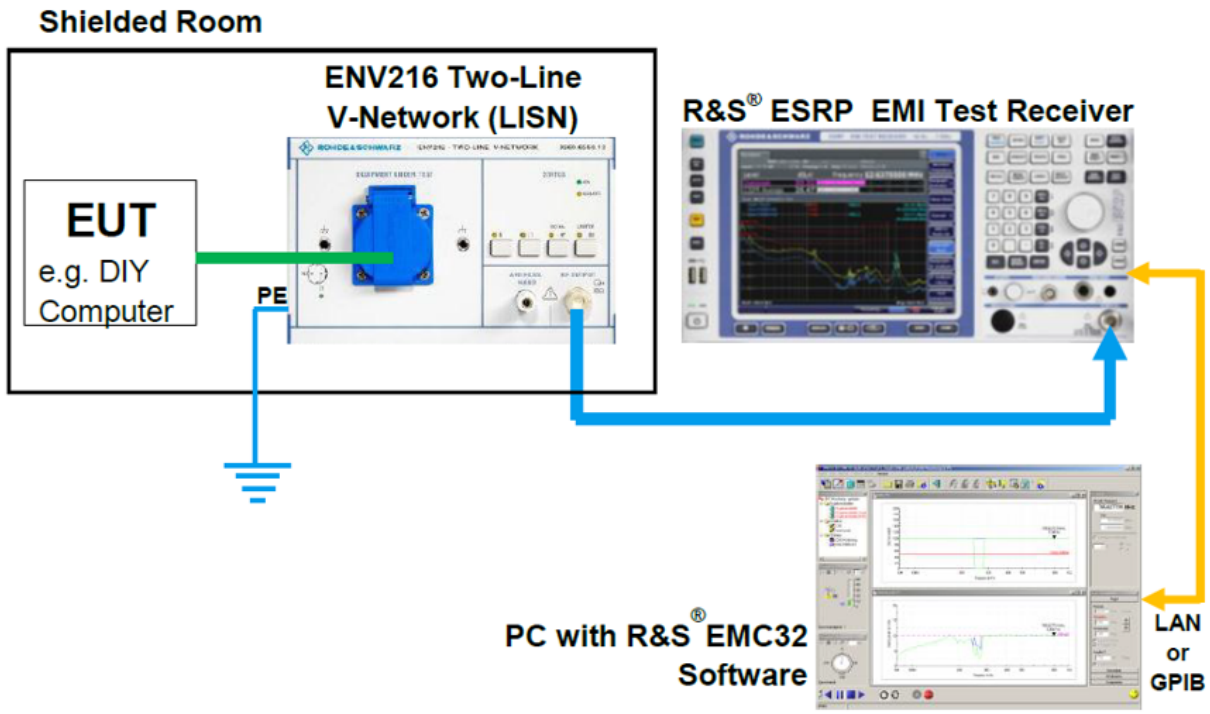


Figura 4.3 – Configuração de um teste EMI conduzido.

EMI irradiado

Conforme mostrado na Figura 4.4, o teste de EMI irradiado necessita de uma mesa giratória de material dielétrico para rotacionar o DUT. Essa condição garante a captura do sinal emitido em todas as direções. O ambiente deve ser uma câmara semianecóica, ou anacóica dependendo da norma. A blindagem é necessária para que o campo eletromagnético provindo do medido não seja resultado de ruídos do ambiente. Existem vários tipos de salas blindadas, incluíndo câmara semi-anecóica, câmara totalmente anecóica, sala blindada sem absorventes, sala de controle, sala para medições conduzidas, célula de teste blindadas como TEM e GTEM. Nesses ambientes é necessário considerar aspectos como tamanho da sala, tipos de absorção e arranjo, faixa de frequência, infraestrutura de rede, luz e ar condicionado. Adcionalmente, a interface com a instrumentação, como filtros, *feedthroughs*, extensão IEEE 488, acesso aos painéis, pontos de conexão, não podem invalidar a eficácia da blindagem. Devemos evitar sempre armazenar os equipamentos na câmara durante as medições.

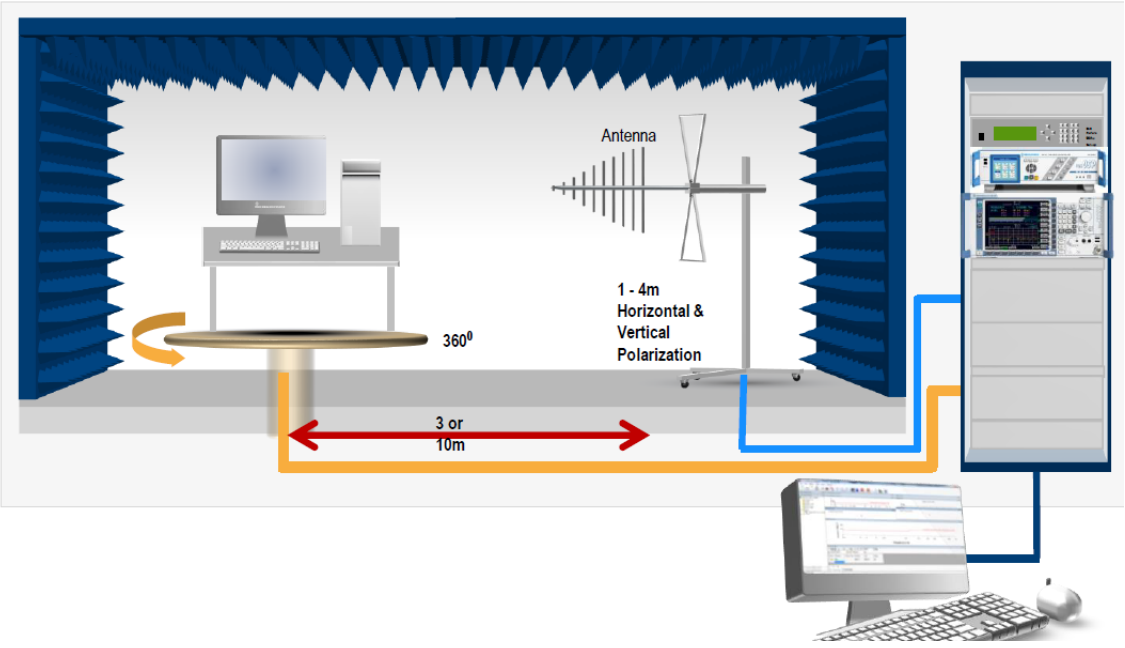


Figura 4.4 – Configuração de um teste EMI irradiado.

Geralmente posiciona-se a antena a 3 metros de distância do DUT. Diminuir essa distãncia e operar no campo próximo faz com que os resultados não se traduzam diretamente nos testes de campo distante.

O receptor deve estar calibrado para toda a faixa de frequência do teste e a RBW deve ser ajustada para 1 MHz. A faixa de frequência é de 30 MHz até 1/3/6/18 ou 40 GHz. Recentemente, os receptores de EMI adcionaram em seus testes a metodologia de Transformada rápida de Fourier (FFT). Trata-se da forma de processar o sinal medido. As medidas realizadas em receptores FFT foram inclusas nas normas como medida aceita para certificação. Antes, a FFT já era usada como uma medida prévia mais rápida, mas ainda era necessário uma medida em *stepped scan* (tradicional) final. A diferença, de forma resumida, é a capacidde de capturar mais sinais de uma só vez usando bandas maiores. Assim, ao invés

de ir varrendo (*scan*) frequência por frequência, é possível capturar uma “porçãoâ maior do sinal e depois via FFT obter as frequências individuais de interesse.

4.0.2 EMS

O objetivo do teste de imunidade (também chamado de teste de suscetibilidade), por outro lado, é verificar se um dispositivo é capaz de funcionar adequadamente, mesmo quando exposto a níveis (geralmente significativos) de energia de radiofrequência. Existem inúmeros exemplos bem divulgados de aparelhos eletrônicos que apresentaram mau funcionamento ou falharam quando expostos a altos níveis de energia de RF; em alguns casos, o mau funcionamento pode levar a lesões e até a morte.

EMS conduzido

Para medições EMS conduzidas de acordo com IEC/EN61000-4-6, existem três métodos possíveis de acoplamento com o DUT: CDN (*Coupling Decoupling Network*), BCI (*Bulk Current Injection*) e EM-Clamp. O método de teste mais comum é o acoplamento via CDN (Figura 4.5) pois requer menos energia e é facilmente plugado com carregadores comuns de celulares, por exemplo. O diagrama de blocos na Fig.5 mostra a configuração básica para medições de acordo com IEC/EN61000-4-6 com uma CDN.

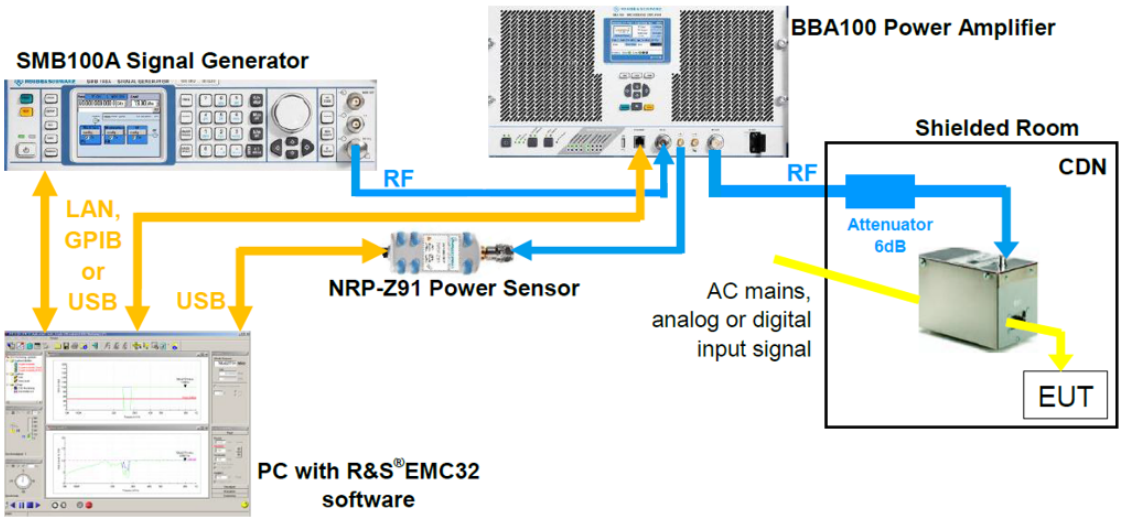


Figura 4.5 – Configuração de um teste EMS conduzido.

Os dois principais testes de EMS conduzida são transientes e interrupções de tensão. Um gerador ou simulador de transientes geralmente é usado junto com a CDN. Os testes transitórios são projetados para simular uma variedade de fenômenos que ocorrem nas linhas de energia e controle, dependendo do ambiente eletromagnético. Para variações e interrupções de tensão, o grau de variação na tensão será determinado com alguma porcentagem determinada pela norma. Variações em diferentes graus do sinal sinusoidal

também serão realizadas para encontrar o pior cenário possível. No caso de testes de interrupções, o tempo de interrupção será definido pela norma de acordo com o DUT.

EMS irradiado

EMS irradiada pode ser descrita como o *setup* mais complexo de EMC, como é ilustrado na Figura 4.6. Nessa configuração é necessário realizar os testes em uma câmara anecóica para impedir que o campo irradiado do teste "vaze" e interfira com outros equipamentos. Estão disponíveis vários tipos de absorvedores, e os tipos mais comuns são pirâmides de espuma carregadas de carbono ou ladrilhos de ferrite. Os requisitos básicos são um gerador de sinais, um amplificador de potência de banda larga, medidor de potência e um transdutor. O último é tipicamente um conjunto de antenas.

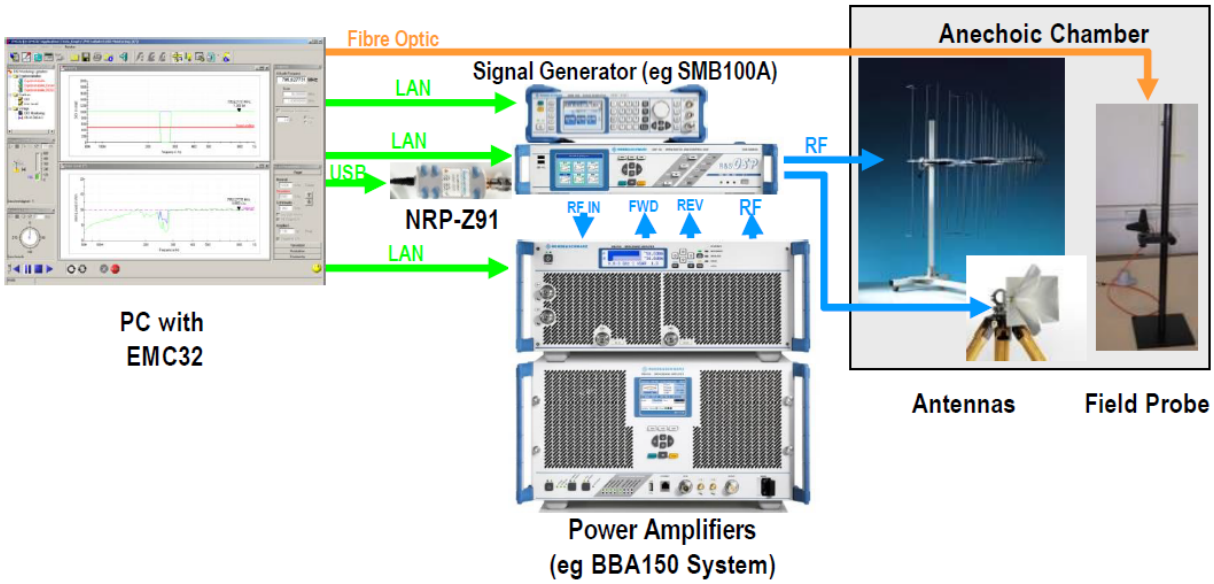


Figura 4.6 – Configuração de um teste EMS irradiado.

As emissões irradiadas representam o maior desafio nos testes de imunidade. A primeira razão para isso é que o teste de imunidade irradiada (ou suscetibilidade) frequentemente requer a criação de intensidades de campo elétrico muito altas, com valores típicos que variam de 3 a 200V/m. Dependendo da frequência, distância e tipo de antena, isso pode exigir uma potência de saída de centenas ou milhares de watts - muito maior do que a maioria dos geradores de sinal são capazes de criar. A segunda razão é que converter a potência de saída conduzida em um cabo em campos eletromagnéticos irradiados, o uso de antenas nunca é 100% eficiente, particularmente nas amplas faixas de frequência especificado na maioria dos padrões e regulamentos EMC. Por essas razões, os amplificadores EMC são uma parte indispensável dos testes de EMC.

Lembre-se de que, nos testes de imunidade irradiada, estamos tentando determinar as frequências e os níveis que afetam adversamente o funcionamento de um DUT. Se um amplificador está em compressão e gerando produtos de intermodulação, surge a possibilidade de que o DUT possa estar reagindo a energia nesses harmônicos e

produtos de intermodulação em vez de reagir (exclusivamente) a energia na frequência fundamental (pretendida). A presença de harmônicos e produtos de intermodulação pode, portanto, tornar muito difícil determinar quais componentes de frequência são responsáveis por comportamento indesejável do DUT. Observe também que, embora os harmônicos normalmente tenham uma potência menor do que a fundamental, o harmônico pode, na verdade, produzir uma intensidade de campo mais alta devido à resposta de frequência da antena. Por essas razões, os padrões EMC geralmente definem os níveis máximos de harmônicos em vários níveis de potência para um determinado cenário de teste.

É necessário calibrar a câmara anecóica transmitindo o sinal de RF e medindo por meio de uma field probe o campo em Volts/metro em pontos especificados para determinar se o campo atende aos critérios de uniformidade. Posteriormente, cria-se uma tabela do nível de sinal necessário em cada frequência de teste para atingir a intensidade de campo necessária quando a média é calculada sobre o volume de teste. O DUT não está presente durante a calibração da câmara, mas mesmo assim os níveis de campo acima do volume de teste podem variar em +6dB, -0dB do nominal. Os DUTs são testados colocando o volume de teste na câmara, com a antena transmissora na mesma posição da calibração e, em seguida, repetindo o arquivo de níveis, criado anteriormente.

Outro desafio no teste de imunidade é a extrema variabilidade da impedância de carga. Ao contrário de uma carga simulada puramente resistiva, a impedância das antenas usadas no teste de imunidade radiada normalmente varia enormemente em função da frequência. é difícil, senão impossível, projetar antenas que tenham um VSWR baixo que 2:0 mesmo em faixas de frequência moderadamente grandes. Muitos testes de imunidade irradiada abrangem uma faixa de frequência de vários megahertz a vários gigahertz e, mesmo em faixas de frequência muito menores, a impedância da antena pode variar amplamente. Mesmo para antenas bem projetadas e com manutenção adequada usadas em testes de imunidade radiada, o VSWR pode facilmente exceder 6:1 em sua faixa operacional nominal.

Uma solução potencial para atingir valores VSWR “razoáveis” nessas amplas faixas de frequência é separar a faixa de frequência total em subfaixas menores e usar uma antena diferente em cada uma dessas faixas. Com base em considerações práticas, o número de subfaixas é geralmente limitado a seis ou menos, dependendo do padrão. Este método é bastante comum em testes de imunidade radiada, embora exija a comutação manual ou eletrônica entre as antenas (com um impacto correspondente em tempo de teste e eficiência). No entanto, ainda pode ser difícil obter um “bom” VSWR mesmo em uma única subfaixa, especialmente quando o VSWR é devido a fatores externos à antena.

Outra solução potencial é alterar a impedância (aparente) da carga. Uma vez que o VSWR alto é causado por uma incompatibilidade de impedância significativa, as chamadas redes de casamento as vezes são usadas na tentativa de fazer a impedância de carga “combinar” com a fonte (ou saída do amplificador). Embora as redes casadas sejam usadas com sucesso em muitas aplicações, é difícil projetar redes casadas que sejam eficazes e/ou eficientes em faixas de frequência muito amplas.

Alguns produtos empregam constantes de tempo muito longas, e pode ser impraticável acelerá-las sem alterar suas características EMC; nesse caso, longos períodos de teste podem ser necessários. Portanto, é bastante comum criar software de produto especial para testes de imunidade irradiada, reduzir o tempo de teste e reduzir os custos de teste. No exemplo da Fig. 4, o software R&S EMC32 está sendo o auxiliador das medições.

Conclusões

O minicurso apresentou os fundamentos de EMC e descreveu as 4 principais configurações de testes de pré-conformidade. Para cada teste foram descritos os equipamentos, metodologia e as faixas de frequência de acordo com a CISPR. Adcionalmente, apresentamos para cada *setup* os principais desafios de medição.

Referências Bibliográficas

[1] *Testing for EMC Compliance: Approaches and Techniques*, pages 100–200. John Wiley & Sons, Ltd, 2004.

[2] P. Kotsampopoulos, A. Rigas, J. Kirchhof, G. Messinis, A. Dimeas, N. Hatziargyriou, V. Rogakos, and K. Andreadis. Emc issues in the interaction between smart meters and power-electronic interfaces. *IEEE Transactions on Power Delivery*, 32(2):822–831, 2017.

[3] Academy of EMC. The academy of emc, a free and independent source of information about electromagnetic compatibility. *Disponivel em: <https://www.academyofemc.com/emc-standards>*, Acessado em: junho de 2020.

[4] Released: 2019. En iec 61000-3-2: Electromagnetic compatibility (emc) - part 3-2: Limits for harmonic current emissions (equipment input current ≦16 a per phase). *European Standards*, page 1, 2019.

[5] Released: 2019. En 55011, industrial, scientific and medical equipment: Radiofrequency disturbance characteristics - limits and methods of measurement. *European Standards*, page 1, 2019.

[6] Released: 2020. Cispr/cis/h - limits for the protection of radio services. *International Standard*, page 27, 2020.

[7] Released: 2019. En 55022, information technology equipment - radio disturbance characteristics - limits and methods of measurement. *European Standards*, page 1, 2019.

[8] C. Wu, H. Kim, J. He, N. Erickson, S. Cho, D. Kim, Y. Hur, D. J. Pommerenke, and J. Fan. Analysis and modeling of conducted emi from an dc power supply in led tv up to 1 mhz. *IEEE Transactions on Electromagnetic Compatibility*, 61(6):2050–2059, 2019.

[9] L. Yang, S. Wang, H. Zhao, and Y. Zhi. Prediction and analysis of emi spectrum based on the operating principle of emc spectrum analyzers. *IEEE Transactions on Power Electronics*, 35(1):263–275, 2020.

[10] E. Griese. Reducing emc problems through an electrical/optical interconnection technology. *IEEE Transactions on Electromagnetic Compatibility*, 41(4):502–509, 1999.

[11] F. Pythoud and E. Tas. Design of a reference device for radiated immunity inter-laboratory comparison. In *2017 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, pages 1–4, 2017.

[12] H. Nzalli, W. Wilkening, and R. H. Jansen. Load dependency assessment of the emc immunity for integrated low side drivers. *IEEE Transactions on Electromagnetic Compatibility*, 56(3):581–588, 2014.

[13] D. Zhang, E. Cheng, H. Wan, X. Zhou, and Y. Chen. Prediction of electromagnetic compatibility for dynamic datalink of uav. *IEEE Transactions on Electromagnetic Compatibility*, 61(5):1474–1482, 2019.

[14] Z. Xu, B. Ravelo, O. Maurice, J. Gantet, and N. Marier. Radiated emc kron's model of 3-d multilayer pcb aggressed by broadband disturbance. *IEEE Transactions on Electromagnetic Compatibility*, 62(2):406–414, 2020.

[15] M. Koohestani, R. Perdriau, J. Levant, and M. Ramdani. A novel passive cost-effective technique to improve radiated immunity on pcbs. *IEEE Transactions on Electromagnetic Compatibility*, 61(6):1733–1739, 2019.

[16] ISO 13528. Statistical methods for use in proficiency testing by interlaboratory comparison. *British Standards Institution*, page 1, 1st ed.,2005.

[17] Nikolaos V. Kantartzis and Theodoros D. Tsiboukis. Modern emc analysis techniques volume i: Time-domain computational schemes. *Synthesis Lectures on Computational Electromagnetics*, 3(1):1–224, 2008.

[18] Mark I Montrose. *EMC and the printed circuit board: design, theory, and layout made simple*, volume 6. John Wiley & Sons, 2004.

[19] R. Taylor. Cutting costs of compliance [emc testing]. *IEE Review*, 41(4):SUPL19–SUPL20, 1995.

CAPÍTULO

5 Segurança em Aplicações de Internet das Coisas: Bluetooth Low Energy, casos de uso e vulnerabilidades

Ívilla I. S. Silva (UFPB), Aellison C. T. Santos (UFPB), Michael A. F. Monteiro (UFPB), Pablo H. R. Goes (UFPB), Renan M. Silva (UFPB), Vivek Nigam (UFPB-Fortiss) e Iguatemi E. Fonseca (UFPB)

Introdução

Em uma camada de alto nível, Internet das Coisas (*IoT - Internet of Things*) é o termo utilizado para descrever a capacidade dos dispositivos de se comunicarem entre si e de se conectarem com a internet. Por meio desta tecnologia, é possível criar uma rede de dispositivos capazes de operar de forma coordenada com pouca ou sem interferência humana. Devido a sua enorme gama de aplicações, a IoT se tornou uma das principais apostas para revolucionar a forma como interagimos com a tecnologia, tornando-a mais ubíqua, eficiente e autônoma.

Nota-se o potencial tecnológico e financeiro de aplicações IoT tendo em vista que, apesar da crise econômica causada pela pandemia devido ao COVID-19, estimativas indicam que a área voltará a ter um crescimento anual de dois dígitos em 2021 e que atingirá o crescimento anual esperado de 11,3% para o período de 2020-2024 [1]. O número de dispositivos conectados ao redor do mundo no final de 2018 era de aproximadamente 12 bilhões, sendo que estimativas apontam que em 2030 este número atingirá 50 bilhões [2]. Todos estes dispositivos estarão coletando, processando e enviando informações, logo será necessário criar mecanismos de segurança que consigam manter todos esses dados protegidos.

O objetivo deste minicurso é apresentar os principais conceitos envolvendo IoT. Na primeira parte será abordado o cenário atual de IoT no mundo, cobrindo as principais tecnologias e protocolos utilizados, casos de uso e segurança em aparelhos. Na segunda parte serão abordados os ataques mais comuns em dispositivos IoT, apresentando algumas

referenciais e cenários de ataque. Na terceira parte, será apresentado o protocolo *Bluetooth Low Energy* (BLE), que é uma das tecnologias de comunicação mais utilizada em dispositivos IoT atualmente. Na quarta parte, serão demonstrados alguns ataques que exploram as principais vulnerabilidades do BLE. Na quinta parte, serão apresentadas algumas contramedidas aos ataques abordados na parte anterior. E por último, a conclusão que buscará condensar os principais conceitos apresentados e estabelecer algumas indicações sobre o futuro.

O Cenário da Internet das Coisas

Com o rápido avanço da tecnologia nas últimas décadas, o conceito de dispositivos inteligentes ganhou muita força com o advento da Internet das Coisas. Sengupta et al. [3] define IoT como um grupo de objetos estáticos e/ou móveis interconectados, como dispositivos equipados com comunicação, sensores e módulos atuadores conectados pela Internet.

Aparelhos adeptos ao paradigma IoT, devido a sua simplicidade e facilidade de conexão, tem se tornado muito comuns entre usuários para troca de informações e encontram-se presentes em: celulares, laptops, automóveis, impressoras, teclados, headsets, caixas de som, monitores fitness e recentemente em dispositivos médicos e pessoais como smartwatches. De acordo com um relatório publicado pela Ericsson [3], o número de dispositivos IoT conectados em 2022 será de cerca de 18 bilhões.

Dentre várias tecnologias presentes em dispositivos IoT, podemos destacar o *Bluetooh Low Energy* (BLE). A tecnologia BLE surgiu a partir do Bluetooth 4.0, propiciando uma conexão com consumo e custo muito baixo em relação ao Bluetooth tradicional, mantendo o mesmo alcance. Apesar da simplicidade de conexão, mais que o habitual, Ray et al. [4] enfatiza que mesmo assim o BLE provê as propriedades de segurança que um protocolo de comunicação espera, como confidencialidade, integridade e disponibilidade.

Apesar de ser amplamente usado em dispositivos IoT e prover a segurança básica de comunicação, o BLE possui problemas de segurança na sua conexão, uma delas é relativa a forma de pareamento entre os dispositivos [5]. Até sua Versão 4.2, o BLE possuía três métodos de geração da STK (*Short Term Key*), que é criada no momento do pareamento. O "*Just Works*", em que um código 000000 é gerado de forma *default* para o STK, temos o "*Passkey Entry*" que é um código de 6 dígitos gerado aleatoriamente pelo dispositivo periférico e o "*Out of Band*" que usa uma senha de 128 bits e um protocolo diferente do BLE.

Saravanan et al. [6] mostra que o pareamento do tipo "*Just Works*" é a forma mais comum de conexão entre dois dispositivos BLE, devido a sua simplicidade, e foi modelado para realizar conexões nas quais pelo menos um dos dispositivos não apresenta teclado ou display para inserção de valores. Esse modelo de pareamento não possui proteção contra o ataque MitM (*Man in the Middle*), em comparação a outros métodos, na qual se exige um PIN.

Alguns ataques podem ser aplicados a dispositivos BLE que usam o método de pareamento "*Just Works*" [4] como ataques de negação de serviço (DoS), ataques Fuzzing, sniffing, e ataques em dispositivos que utilizam o protocolo *Secure Simple Pairing* (SSP) do Bluetooth. Dentro desse grupo podemos destacar ataques como MitM, Sniffing, DoS usando Spoofing dentre outros.

Outra tecnologia IoT atualmente de grande relevância é a Identificação por Rádio-Frequência (RFID - *Radio Frequency Identification*). Chunling [7] define RFID a partir de um conceito semelhante à tecnologia de código de barras, mas sem exigir uma visibilidade direta das entidades monitoradas. O RFID precisa de um equipamento de leitor e etiquetas ou cartões especiais anexados aos produtos para que os produtos sejam rastreados.

O RFID oferece benefícios, como rastreamento de ativos, monitoramento das condições de segurança e ajuda na prevenção da falsificação, desempenhando um dos papeis fundamentais na revolução tecnológica junto com a Internet e os dispositivos móveis.

As tecnologias RFID podem ser classificadas em três categorias: RFID passivo, RFID ativo e RFID semi-passivo. Com base na frequência de rádio usada, as tecnologias RFID passivas são geralmente categorizadas em RFID de baixa frequência (LF - *Low Frequency*), RFID de alta frequência (HF - *High Frequency*), RFID de Frequência Ultra-Alta (UHF - *Ultra High Frequency*) e RFID de micro-ondas.

Com seus conceitos fundamentados a mais de meio século, foram nos últimos anos que essa tecnologia ganhou impulso devido ao seu baixo custo e maior capacidade de processamento. Algumas aplicações podem ser encontradas usando a tecnologia RFID, como por exemplo, gerenciamento da cadeia de suprimentos, varejo, manutenção de aeronaves, anti-falsificação, manuseio de bagagem e saúde.

Outra tecnologia que se destaca é o Zigbee, estabelecida em 2002, que permite a formação de uma rede de área pessoal sem fio de baixa potência e baixa taxa de transferência de dados operando em bandas não licenciadas, ou seja, principalmente na banda de frequência de 2,4 GHz. Com um alcance de 10 a 100 metros, suporta três topologias de rede diferentes: árvore, estrela e malha. Atualmente está na sua Versão 3.0, permitindo maior interoperabilidade, como também a conectividade com a internet [8].

Zigbee especifica três tipos de dispositivos lógicos diferentes, o Zigbee Coordenador (ZC - *Zigbee Coordinator*), o Zigbee Roteador (ZR - *Zigbee Router*) e o Zigbee Dispositivo Final (ZED - Zigbee End-Device). O coordenador possui a função de coordenar a rede e ser uma ponte para outras redes. O Roteador tem a função de ser o nó intermediário entre o ZC e os dispositivos finais, como também permitir a entrada de outros roteadores e dispositivos finais na rede. E um dispositivo final, que não pode rotear tráfego e nem permitir a entrada de outros dispositivos na rede, comunica-se apenas com o seu nó pai, que geralmente é um roteador.

Em termos de segurança o Zigbee oferece duas camadas para prover a segurança necessária nas suas comunicações, a camada de rede e de aplicativo. Entre os seus serviços de segurança, estão o estabelecimento e transporte de chaves, confidencialidade e integridade

de quadros por meio do algoritmo de criptografia AES-CCM. O Zigbee pode ser suscetível a vários ataques dentre eles podemos citar: Sniffing, Replay, Jamming, etc.

Ainda sobre as tecnologias IoT podemos citar o ZWave, Kambourakis et al. [8] define ZWave como um protocolo de automação residencial criado em 2001. O ZWave permite a comunicação em uma rede mesh multi-hop de até 232 nós, que opera na banda de 800 - 900 MHz com um alcance físico de até 100 m. É apoiado pela Z-Wave Alliance, que fornece a certificação para os produtos.

Em termos de segurança, a versão inicial do Zwave, o S0, não exige nenhum tipo de criptografia, tornando-o vulnerável para ataques de injeção, manipulação de mensagens e espionagem. No entanto, uma camada de segurança pode ser implementada por meio do algoritmo de criptografia AES-128-CCM, no qual as respectivas chaves são geradas durante a fase de emparelhamento tendo como entrada os dados enviados aos nós pelo controlador primário, que se encarrega de gerenciar toda a rede. Na versão S0, o emparelhamento é protegido por uma chave padrão composta apenas de zeros, o que o torna bastante vulnerável e facilmente pode ser obtida a chave da rede. ZWave pode ser suscetível a diferentes ataques, entre eles podemos citar: *Replay Attack, Evesdropping Attack, Key Derivation*, etc.

A partir da explanação de algumas das várias tecnologias IoT que existem atualmente, pode-se notar que, dentre as características compartilhadas, há a preocupação com segurança de dados e protocolos em todas elas. Percebe-se que devido a presença de dispositivos IoT nas nossas casas e nossos locais de trabalho, a segurança e privacidade são fatores decisivos para a aceitação destes dispositivos no mercado. Ao tempo que a adoção de dispositivos IoT vêm crescendo de uma maneira muito rápida, a segurança se torna o assunto de extrema urgência.

Ataques e Vulnerabilidades em Tecnologias IoT

Com a passagem dos dias, observa-se novas formas de facilitar atividades serem descobertas e aderidas às nossas vidas por meio de novas tecnologias e paradigmas, como nos encontrando vivenciando na era IoT. Apesar de oferecer comodidade, esses dispositivos também possuem vulnerabilidades que geram riscos crescentes aos seus usuários. Como mostrado no estudo feito pela F-Secure [9], apenas no primeiro semestre de 2019 a quantidade de ciberataques a dispositivos IoT chegou a casa dos bilhões, três vezes mais do que o número de ataques no mesmo período do ano anterior.

A segurança de dispositivos IoT se tornou uma prioridade e tópico de pesquisas extensas [4] [10], e com base nestas pesquisas e outras mais, listamos algumas das metodologias de ataques mais comuns a dispositivos IoT com o propósito de apresentar as vulnerabilidades comumente exploradas.

5.0.1 Sniffing

A transmissão de pacotes feita por dispositivos IoT é realizada majoritariamente através de padrões de comunicação sem fio, o que oferece maior mobilidade para o usuário. Em contrapartida, tais padrões de comunicação abrem portas para diversos tipos de ataques, sendo um deles o *Sniffing*. Pelo fato dos pacotes trocados entre dispositivos IoT compartilharem o mesmo meio de transmissão que diversos outros, o ar, eles estão vulneráveis à Escuta Passiva (*Passive Eavesdropping*).

Devido aos recursos necessários para realizar a captura de pacotes, *Sniffing* é considerada uma prática simples e barata. O atacante precisa apenas de um dispositivo capaz de capturar pacotes difundidos no meio de transmissão que o dispositivo alvo usa. Os pacotes capturados podem então ser analisados com ferramentas de análise de pacotes como o *Wireshark* [11] para extração de informações relevantes para o atacante. Por si só o *Sniffing* é uma técnica que coloca em risco a confidencialidade dos dados transmitidos entre dois dispositivos, tendo em vista que pacotes não criptografados podem ser lidos diretamente.

Também é preciso notar que o *Sniffing* pode ser visto apenas como uma etapa para ataques ainda mais elaborados. Ao fazer a captura de pacotes, é aberto um leque de novas vulnerabilidades que o atacante pode explorar, por exemplo, esses pacotes capturados e analisados podem ser armazenados e posteriormente usados para um *Replay Attack* ou até mesmo modificados para realização tarefas maliciosas em nome de um usuário legítimo. Ataques como esses podem ser vistos nos tópicos restantes desta seção.

5.0.2 Replay Attacks

Um Ataque de repetição (*Replay Attack*), ocorre quando o atacante tem posse de um ou mais pacotes de conteúdo significativo do seu alvo. Ele então reproduz estes pacotes na tentativa de se passar pelo dispositivo alvo, podendo realizar atividades ou utilizar recursos do dispositivo de destino de maneira indevida. De forma mais detalhada, o atacante deve primeiro analisar o tráfego gerado pelo seu alvo, capturando-o ao usar técnicas de *Sniffing*. Com os pacotes a disposição o atacante deve apenas aguardar o momento propício, dado o contexto do cenário que o mesmo se encontra, para fazer a reprodução dos pacotes capturados.

Em um exemplo hipotético e simplificado, o depósito de um aeroporto é fechado por um cadeado inteligente (*Smart Lock*), sendo aberto apenas pelos funcionários autorizados (*e.g.* um supervisor). Ao visitar o depósito, o supervisor irá dar início a liberação da trava inteligente, informando suas credenciais pelo celular. O atacante, fazendo *Sniffing* do meio, captura e analisa o tráfego emitido entre o aparelho celular da vítima e a trava inteligente, fazendo a captura desses pacotes. Em posse desses pacotes, o atacante pode posteriormente dar início a liberação da trava reproduzindo os pacotes capturados necessários e personificando o supervisor, tendo acesso ao depósito originalmente restrito a ele.

5.0.3 Relay Attacks

Relay Attack, ou ataque de retransmissão, é um ataque que tem a finalidade de retransmitir mensagens de um local para o outro a fim de fazer uma entidade parecer mais próxima do que ela está de fato.

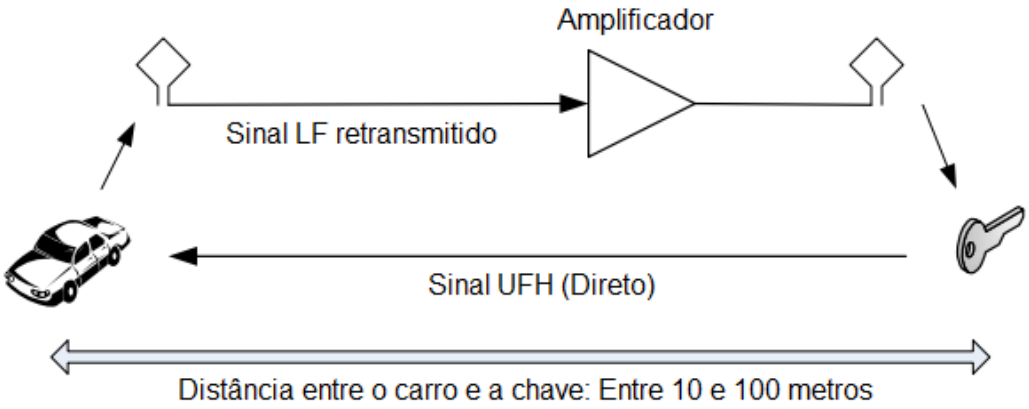


Figura 5.1 – Exemplo de Relay Attack.

Um exemplo de *Relay Attack* pode ser visto na Figura 5.1, no qual o objetivo do atacante é roubar o carro da vítima que possui travas e sistema de iniciação inteligentes. O atacante coloca um dos dispositivos de retransmissão de mensagens próximo a chave do carro e o outro dispositivo de retransmissão próximo ao carro, tendo um amplificador de sinal entre eles. O atacante então dá inicio ao protocolo de iniciação do carro, retransmitindo as mensagens entre a chave e o veículo, possibilitando que o carro seja aberto e ligado, mesmo com a chave estando distante do mesmo [12].

5.0.4 Jamming

Jamming é um ataque em que interferências de rádios são lançadas em conexões legítimas de dispositivos IoT, afim de causar uma negação de serviço nos dispositivos, ou seja, impedindo os dispositivos de receberem pacotes enviados por dispositivos legítimos.

É possível quebrar a metodologia de *jamming* em quatro modelos de ataque [13], sendo eles: o *jammer* constante, que envia continuamente sinais de rádio na conexão; o *jammer* enganoso, que envia pacotes regulares no canal sem qualquer intervalo entre as transmissões, fazendo o receptor interprete o pacote como uma mensagem legítima e o mantendo no estado de recepção; o *jammer* aleatório, que alterna entre enviar ruído no meio e hibernar; e o *jammer* reativo, que, diferentemente dos modelos anteriores, não bloqueia o canal ativamente, visto que o dispositivo bloqueador inicia o processo de *jamming* somente ao detectar a tentativa de comunicação entre dispositivos.

Santos et al. [14] aponta limitações do ataque *jamming*, pois existe a necessidade do dispositivo que realiza a interferência de estar próximo ao alvo. Em outras palavras, caso o dispositivo alvo mude de posição, o atacante terá duas alternativas: colocar um novo dispositivo de interferência próximo a nova localização do alvo, ou mudar a posição do

dispositivo de interferência original para mais próximo da nova localização do dispositivo alvo.

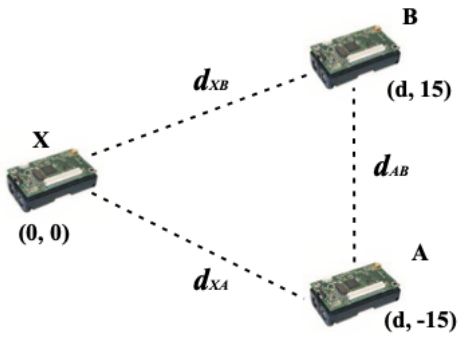


Figura 5.2 – Exemplo de ataque Jamming. Imagem de [13].

Na Figura 5.2, pode-se ver um modelo simples de ataque *jamming*, em que A é o emissor, B é o receptor e X o dispositivo que lança os sinais de rádio na rede, sendo que a potência de transmissão de A, B e X são $-4dBm$, e as distâncias são as mesmas entre A, B e X. Devido a interferência de X, A e B não serão capazes de se comunicar efetivamente, mesmo estando em raio de comunicação.

5.0.5 Man-in-the-Middle

Neste tipo de ataque, o atacante se torna o intermediário entre o dispositivo central e o periférico personificando um ao outro. A ideia focal é que ambos os dispositivos (central e periférico) não saibam da presença do atacante.

Para que o ataque possa ser executado, é necessário que o atacante saiba o endereço de Controle de Acesso à Mídia (MAC - *Media Acess Control*) tanto do dispositivo periférico atacado, como do dispositivo central atacado. Para visualizar os endereços MAC respectivos, o atacante pode fazer uso de um sniffer para capturar e analisar os pacotes, obtendo os dados necessários para a realização do ataque.

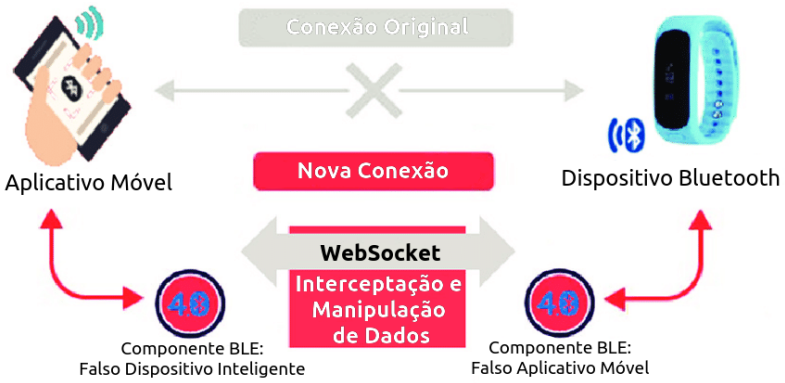


Figura 5.3 – Exemplo de ataque Man In The Middle. Imagem adaptada de [15].

Na Figura 5.3 pode ser visto o funcionamento do ataque *Man-in-the-Middle*. Com o endereço MAC e características dos dispositivos alvos, o atacante então emula dispositivos

falsos com as mesmas características que os originais e solicita conexão com os alvos. Ao conectar-se com os alvos, o atacante tem total controle dos pacotes que trafegam entre ambos, podendo enviá-los do jeito que recebeu, modificar ou simplesmente descartá-los.

5.1 A Tecnologia Bluetooth

Bluetooth é uma tecnologia sem fio para transmissão de dados, criada para atender à necessidade de mercado por uma tecnologia para substituir a conexão cabeada em determinados cenários. Esta nova tecnologia precisava prover comunicação ponto-a-ponto de baixo custo e baixo consumo de energia baseada em radiofrequência.

A tecnologia foi criada em 1994 pela empresa sueca Ericsson e, em 1998, esta empresa se juntou com a Intel, IBM, Nokia e Toshiba para formar o Grupo de Interesse Especial (*Special Interest Group - SIG*) Bluetooth, na intenção de criar um padrão universal e sem direitos de uso [16]. Numa das medidas para atingir este objetivo, foi determinado em sua especificação que seria usada a banda de radiofrequência não-licenciada de 2.4 GHz (2.4000 a 2.4835 GHz) ISM (*Industrial, Scientific and Medical*). Em 2002, a tecnologia Bluetooth foi ratificada pelo IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos) como o padrão 802.15.1.

5.1.1 Protocolo

Nesta subseção, decidiu-se abordar a topologia, o pareamento e as principais diferenças entre as versões *Bluetooth Classic* e *Low Energy*. Considera-se estes pontos importantes para o entendimento da tecnologia visto que passam a ideia mais clara dos principais processos na BLE.

As comunicações entre dispositivos *Bluetooth Low Energy* se iniciam por meio dos pacotes de anúncio. Um dispositivo BLE transmite pacotes de anúncio nos canais de radiofrequência reservados para este propósito, sendo estes os canais 37, 38 e 39. Detalhes sobre estes canais de anúncio podem ser encontrados na subseção 5.1.1. Os pacotes de anúncio contém as características e serviços públicos oferecidos pelo dispositivo que os transmitiu.

Um dispositivo BLE que deseja iniciar comunicação com outro dispositivo específico, deverá realizar uma varredura nos canais de anúncio para encontrar o dispositivo ao qual deseja se conectar. Uma vez obtida a resposta da varredura, o dispositivo fará uma requisição de conexão e iniciará o processo de pareamento, que será explicado na subseção 5.1.1. O diagrama de conexão entre dispositivos BLE é exemplificado na Figura 5.4.

Topologia

Uma conexão estabelecida entre dois dispositivos Bluetooth forma uma *piconet*. Um ambiente pode conter várias *piconets* e em cada uma delas há sempre um dispositivo chamado de nó mestre, enquanto podem haver múltiplos nós escravos. Um nó pode participar de mais

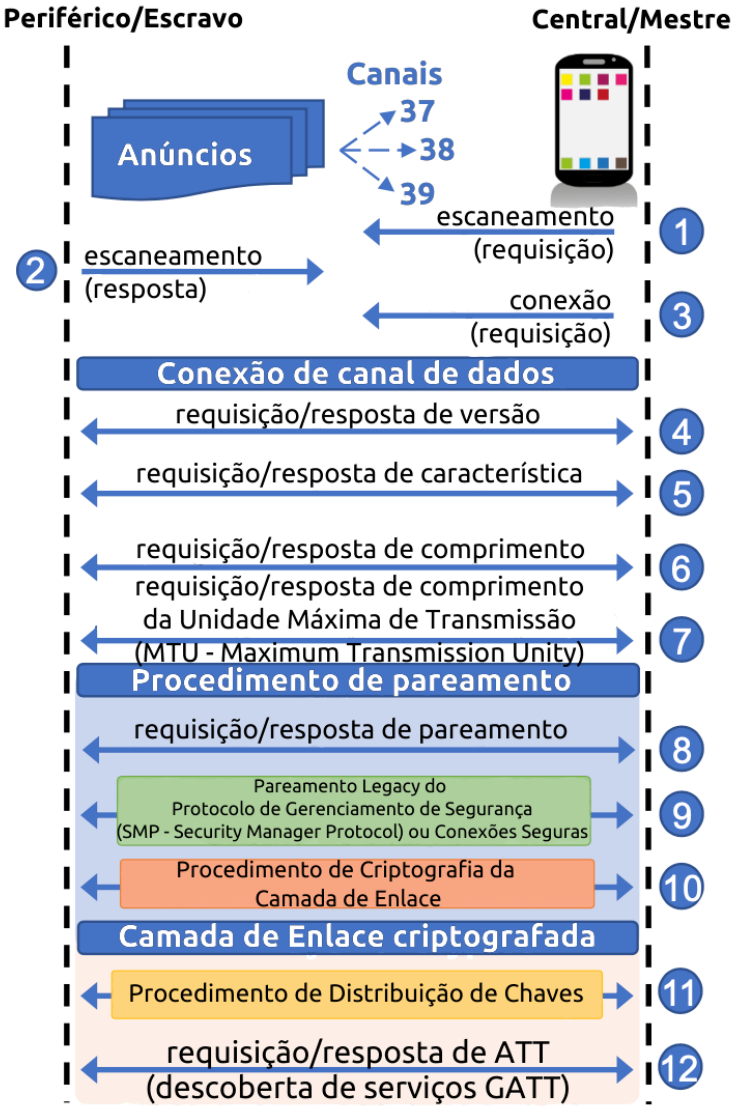


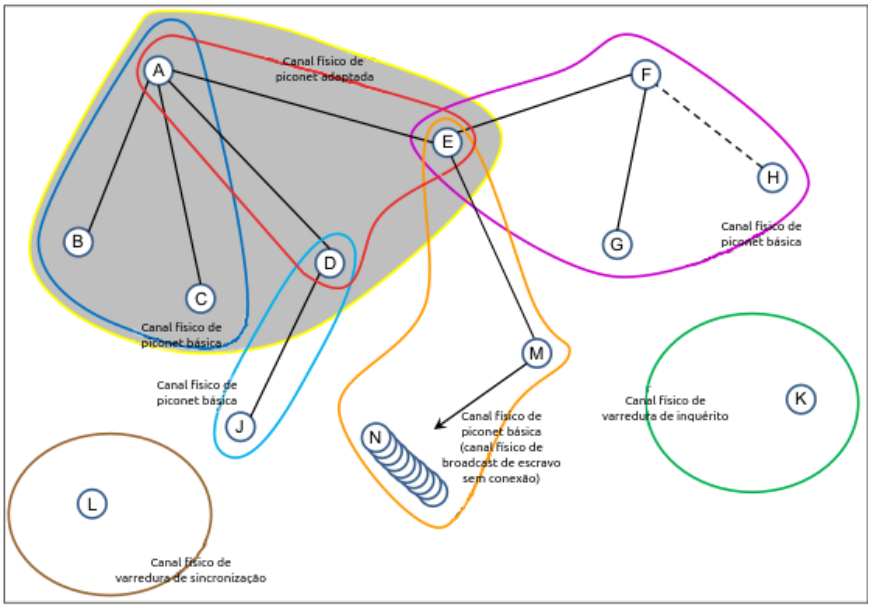
Figura 5.4 – Diagrama de conexão BLE. Imagem adaptada de [17]

de uma *piconet* de forma concorrente, mas um nó mestre de uma *piconet* não pode atuar como mestre nas demais *piconets* que participa. Se diz que um nó que participa concorrentemente em mais de uma *piconet* está em uma *scatternet* [18]. As topologias *Bluetooth Classic* e *Bluetooth Low Energy* podem ser vistas respectivamente na Figura 5.5(a) e 5.5(b).

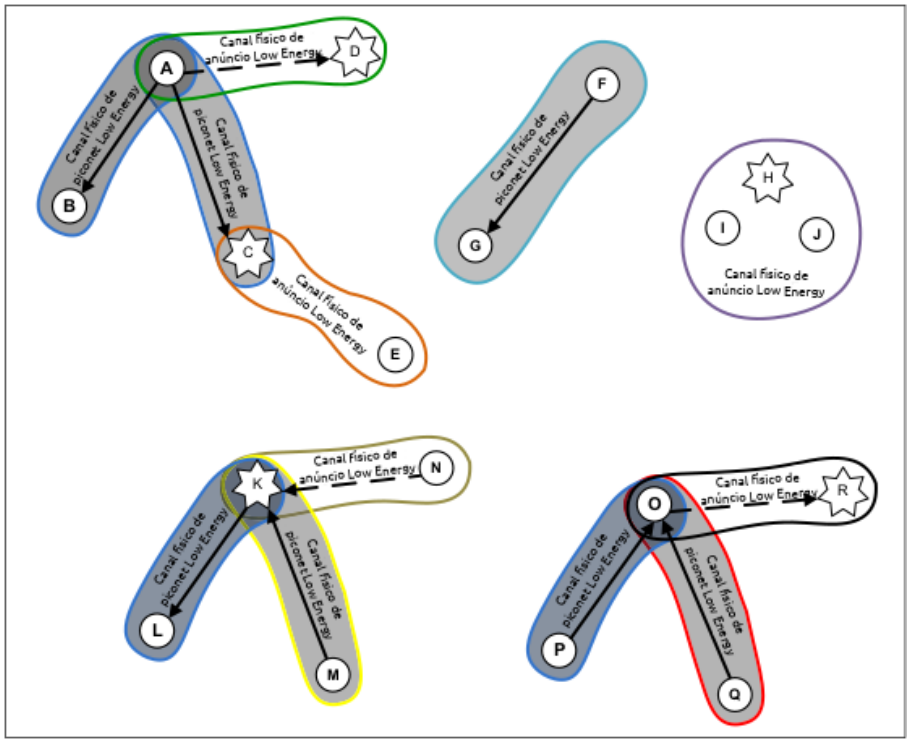
Pareamento

Após realizada a conexão do canal de dados, como mostrado na Figura 5.4, é necessário que seja realizado um processo de associação entre os dispositivos, para que estes possam se comunicar de forma mais simples e segura. Este processo é chamado de pareamento, e permite a conexão de dispositivos sem que seja necessário o processo de descoberta, uma vez que estes dispositivos já estão associados. A tecnologia Bluetooth, em sua versão atual, possui 4 modos de associação usados para pareamento entre dispositivos, são eles: *Numeric Comparison*, *Just Works*, *Out of Band* e *Passkey Entry* [18].

Numeric Comparison



(a) Topologia *Bluetooth Classic*.



(b) Topologia *Bluetooth Low Energy*.

Figura 5.5 – No *Bluetooth Classic* os nós escravos compartilham um mesmo canal físico com o nó mestre. No *Bluetooth Low Energy* cada nó escravo se comunica com um nó mestre em um canal físico separado. Imagens adaptadas de [18].

Este modo de associação foi criado para cenários em que ambos os dispositivos a serem pareados possuem formas de exibir um valor de seis dígitos numéricos para o usuário, assim como uma forma de realizar a confirmação destes números. Durante o processo, o número de seis dígitos é exibido na interface dos dispositivos e é requisitada a confirmação de que ambos os dispositivos apresentam os mesmos dígitos em suas interfaces, uma vez confirmada, os dispositivos estarão pareados. Exemplo de cenário de pareamento: computador e telefone celular.

Just Works

Este modo de associação foi criado para cenários em que ao menos um dos dispositivos não possui uma forma de exibir o número de seis dígitos nem uma forma de entrada para digitá-los. O protocolo utilizado neste modo de associação é o mesmo do *Numeric Comparison*, mas os seis dígitos usados na confirmação são definidos como zero. Exemplo de cenário de pareamento: telefone celular e fones de ouvido sem fio.

Out of Band

Este modo de associação foi criado para cenários em que o dispositivo possui um mecanismo alternativo para descoberta de dispositivos e trocas das chaves criptografadas utilizadas no processo de pareamento. A confirmação de pareamento pode ser implementada em uma via ou via dupla, dependendo do mecanismo alternativo e dos dispositivos envolvidos. Para que este modo seja efetivo, o mecanismo alternativo deve prover propriedades mais seguras que o canal de rádio utilizado pelo Bluetooth. Exemplo de cenário de pareamento: telefones celulares utilizando a tecnologia de NFC (*Near Field Communication*).

Passkey Entry

Este modo de associação foi criado para cenários em que um dos dispositivos possui apenas a capacidade de exibir o número de seis dígitos e o outro possui apenas a forma de entrada para digitá-los. Exemplo de cenário de pareamento: computador e teclado sem fio.

Bluetooth Classic x Bluetooth Low Energy

A tecnologia Bluetooth pode ser dividida em duas arquiteturas, *Bluetooth Classic* (ou convencional) e *Bluetooth Low Energy* (ou *Smart*). Essa divisão é feita pois suas arquiteturas são diferentes e não-comunicáveis entre si. No entanto, é comum encontrar chipsets modo dual, também referidos como *Bluetooth Smart Ready*, que possuem tanto a arquitetura *Classic* quanto *Low Energy*, permitindo comunicação com dispositivos de ambas as versões [19]. As três arquiteturas citadas acima estão ilustradas na Figura 5.6.

A comunicação via Bluetooth é feita por meio do espectro de propagação de salto de frequência [16] (*FHSS - Frequency Hopping Spread Spectrum*), técnica que diminui a interferência de transmissão com outros dispositivos usuários de tecnologias que utilizem a mesma banda de radiofrequência.

No *Bluetooth Classic* [20], a banda de 2.4 GHz ISM é dividida em 79 canais de 1 MHz cada, e são realizados saltos de frequência 1600 vezes por segundo. Os esquemas de modulação utilizados em conjunto com o FHSS podem ser chaveamento gaussiano de mudança de frequência (*GFSK - Gaussian Frequency Shift Keying*), chaveamento de fase de 4 símbolos (*4PSK - 4-symbol Phase Shift Keying*) e chaveamento de fase de 8 símbolos (*8PSK - 8-symbol Phase Shift Keying*)

Continuando na caracterização do bluetooth, Chang [20] explica que no *Bluetooth Low Energy*, a banda é dividida em 40 canais de rádio de 2 MHz, sendo 3 canais de anúncio e 37 canais de dados. O esquema de modulação utilizado é o GFSK. Os canais de anúncio

facilitam a descoberta e comunicação inicial entre dispositivos, já que os pacotes de anúncio informam os parí¢metros necessários para conexão.

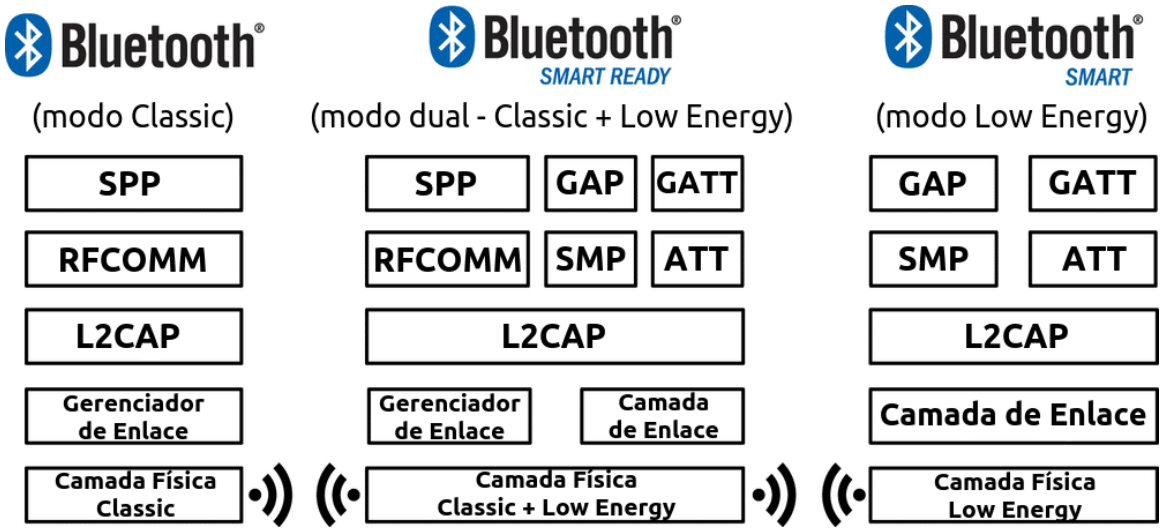


Figura 5.6 – Da esquerda para a direita: arquitetura Bluetooth Classic, arquitetura Bluetooth Smart Ready e Arquitetura Bluetooth Smart. Imagem adaptada de [21].

5.1.2 Contexto de Internet das Coisas e Redes de Sensores

Devido às suas características voltadas ao baixo consumo de energia, a arquitetura *Bluetooth Low Energy* se mostra ideal para uso nos dispositivos que integram a Internet das Coisas (*IoT - Internet of Things*) e redes de sensores, considerando que estes comumente possuem fontes de energia limitadas.

Aplicações da Tecnologia

As aplicações que utilizam dispositivos que implementam a tecnologia BLE são de diversos tipos [22], tais como:

- Controle de Ambiente: monitoramento e controle de temperatura, ventilação, umidade, luz;
- Segurança: sensores de movimento, sensores de abertura de portas e janelas;
- Monitoramento de saúde: sensores de pressão sanguínea, batimentos cardíacos, pulso, oxigênio no sangue e postura;
- Esportes e atividades físicas: contagem de passos, monitoramento de atividade física;
- Detecção de proximidade: localização de dispositivos ao alcance;
- Extensões de interfaces de usuário: notificações e extensões de aplicativos em *smartwatches*;
- Conservação de energia: monitoramento de energia em dispositivos inteligentes.

Demonstração de Ataques e Vulnerabilidades

Para demonstrar algumas das vulnerabilidades presentes no BLE, foram realizados experimentos com alguns dispositivos de uso doméstico e comercial que utilizam esta tecnologia. Os experimentos foram feitos no Laboratório de Redes da UFPB e foram necessários os seguintes softwares e dispositivos:

Hardware:

- Ubertooth One: farejador e interceptador de tráfego de conexões BLE. É uma ferramenta de código aberto utilizada para desenvolver e testar aplicações BLE. O dispositivo pode ser visto na Figura 5.7;
- Cypress CY8CKIT-042-BLE-*Bluetooth Low Energy Pioneer Kit*: kit para desenvolvimento de aplicações Bluetooth Low Energy.
- VEOVA Chest Strap Sports Bluetooth 4.0: cinta medidora de batimentos cardíacos. O dispositivo pode ser visto na Figura 5.8

Software:

- Hcitool [23] e gatttool [24]: interfaces de linha de comando de código aberto ambas presentes na pilha de protocolos Bluetooth para Linux. Sendo a hcitool usada para detectar e se conectar com periféricos, enquanto a gatttool pode ser usada para se conectar com dispositivos BLE, além de ler e escrever características;
- Gattacker: é um pacote para Node.js utilizado para realizar vários tipos de ataques em dispositivos BLE. Nessa demonstração será realizado um ataque *Man-in-the-Middle*. Para utilizá-lo é necessário possuir duas interfaces Bluetooth ou usar uma máquina virtual em um computador que possua uma interface [25];
- Wireshark: farejador e analisador de pacotes de código aberto. Com ele será possível analisar o tráfego de dados capturado com o *Ubertooth One*;
- nRFConnect: é uma ferramenta genérica que permite realizar escaneamentos, criar e transmitir pacotes de anúncio, explorar dispositivos BLE e se comunicar com estes.

5.1.3 Interceptação de Tráfego

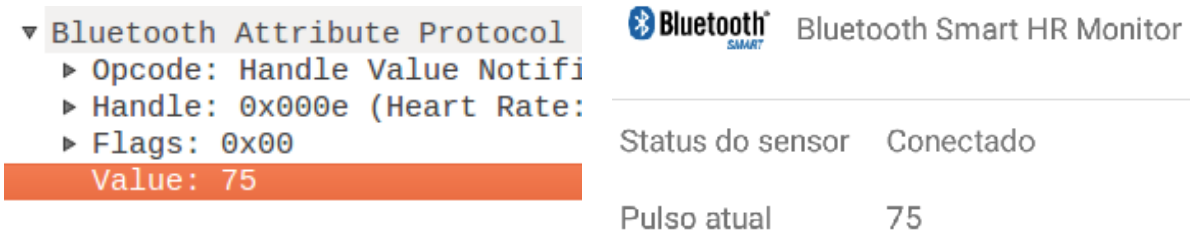
A interceptação de tráfego é um dos ataques mais simples de ser realizado, mas exige equipamentos específicos. Como já foi citado anteriormente, neste experimento será utilizado o *Ubertooth One*. O alvo do ataque será a cinta medidora de batimentos cardíacos *VEOVA Chest Strap Sports Bluetooth 4.0*, que pode ser vista na Figura 5.8. Esta cinta é utilizada em conjunto com aplicativos de monitoramento de atividades físicas, no experimento foi utilizado o *Runtastic Pro* disponível para Android e iOS.

Figura 5.7 – Ubertooth One.

Figura 5.8 – A VEOVA Chest Strap Sports Bluetooth 4.0.

Para que o ataque funcione é necessário iniciá-lo antes que a conexão entre a cinta e o celular seja estabelecida. Primeiramente, é preciso descobrir o endereço MAC da cinta, para isso foi utilizado o hcitool para varrer o ambiente e descobrir os endereços dos dispositivos BLE presentes no ambiente.

Com o endereço MAC da cinta em mãos, já é possível usar o *Ubertooth One* para interceptar o tráfego de dados entre a cinta e o celular. O *Ubertooth One* irá seguir qualquer conexão estabelecida pela cinta e armazenar os pacotes capturados em um arquivo de saída, quando este for especificado. Talvez seja necessário repetir o processo várias vezes para conseguir capturar uma quantidade satisfatória de pacotes devido í s limitações do *Ubertooth*



(a) Conteúdo de um pacote capturado.

(b) Conteúdo exibido no aplicativo.

Figura 5.9 – Resultado da captura de pacotes durante uso da cinta medidora de batimentos cardíacos. Adaptado de [26]

One.

Após o tráfego de dados ser capturado, é possível analisá-lo com o *Wireshark*. Pode-se ver na Figura 5.9 (a) o conteúdo de um dos pacotes capturados. O valor de batimentos cardíacos medido pela cinta é enviado ao celular sem nenhum tipo de encriptação. A Figura 5.9 (b) é uma captura de tela do *Runtastic Pro*, assim é possível confirmar que o valor do pacote capturado bate com o do aplicativo. Também foi descoberto que a cinta não utiliza endereços randômicos, mesmo o BLE possuindo esta funcionalidade, o que torna possível rastrear o seu usuário.

5.1.4 Injeção de pacotes

A injeção de pacotes pode acontecer em vários ataques diferentes, mas em geral, ela pode ser usada para causar interferência na comunicação, ganhar acesso à rede ou até aplicar um ataque de negação de serviço. Neste caso, o objetivo era descobrir qual o padrão das mensagens transmitidas pelo aparelho e utilizar a injeção de pacotes para controlar o dispositivo a partir de um computador com uma interface Bluetooth.

O alvo deste experimento é o termostato inteligente *EQ-3 Radiator Thermostat*, Figura 5.10, que pode ser controlado via BLE através do aplicativo calorBT que está disponível para Android e iOS. Primeiro foi realizado um ataque de interceptação de tráfego utilizando o *Ubertooth One*. Enquanto o *Ubertooth One* capturava o tráfego de dados entre a termostato e o celular, o termostato teve sua temperatura alterada diversas vezes. Depois de analisar os pacotes contendo os dados das requisições de escrita foi possível entender o padrão utilizado na comunicação.

Uma das primeiras vulnerabilidades encontradas foi a possibilidade de se conectar com o termostato sem a necessidade de utilizar o aplicativo especificado. Esta vulnerabilidade permite a aplicação de ataques de injeção de pacotes. Depois de analisar o tráfego de dados, o padrão de escrita dos pacotes para alteração de temperatura foi descoberto. Com isto foi possível utilizar o gatttool para modificar a temperatura com um computador com interface Bluetooth. Além disso, o limite de temperatura especificado pelo aplicativo era de 29,5Â°, no entanto, por meio do gatttool foi possível modificá-la para 127,5 C, como é mostrado na Figura 5.10.

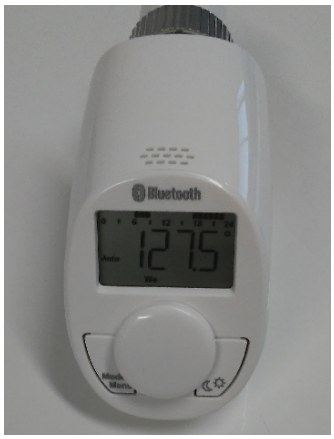


Figura 5.10 – Termostato após a realização da requisição irregular.

5.1.5 Man-in-the-Middle

Para o ataque de *Man-in-the-Middle* são necessárias duas máquinas com um adaptador BLE cada, um dispositivo BLE e o pacote *Gattacker* para no *Node.JS*. O dispositivo alvo utilizado em nossos testes foi a placa integrante do kit de desenvolvimento CY8CKIT-042-BLE, que foi programada para simular a cinta medidora de batimentos cardíacos *VEOVA Chest Strap Sports Bluetooth 4.0*. Foi utilizada a placa citada em vez da cinta pois a cinta em questão não realiza pareamento.

Para facilitar o entendimento, uma das máquinas será referida como Mestre e outra como Escravo. O pacote *gattacker* deverá estar instalado e devidamente configurado em ambas as máquinas para agir como mestre e escravo, seguindo as orientações da documentação do pacote.

O Escravo ficará em escuta da comunicação, transmitindo as informações adquiridas para o Mestre em determinados momentos. O Mestre então realizará o escaneamento por dispositivos BLE e descobrirá o endereço público da placa, que deverá estar ligado e ao alcance. Um arquivo com os dados do pacote de anúncio da placa serão armazenados no Mestre.

Após obter este endereço, o Mestre realizará o escaneamento dos serviços oferecidos pela placa, assim como atributos públicos, consultando o perfil GATT. Um arquivo com os dados dos serviços e atributos da placa serão armazenados no Mestre.

Uma vez que o Mestre possua os arquivos de anúncio e serviços da placa, poderá ser criado um arquivo de repetição de comunicação entre um outro dispositivo BLE e a placa. Para isto, o Mestre transmitirá um pacote de anúncio, utilizando as informações adquiridas da placa, se passando por ela.

O Mestre se conectará com a placa e ficará í espera da comunicação do outro dispositivo. Para este teste foi utilizado um *smartphone android*, utilizando o aplicativo *myWorkouts*, para comunicação com a placa.

Ao se conectar com o aplicativo ao Mestre (que se passa pela placa), o dispositivo passará a repassar os comandos enviados pelo aplicativo í placa, armazenando as informações da comunicação enquanto isso. Uma vez armazenadas as informações, estas poderão ser

alteradas para execução do ataque, transmitindo os comandos sem a necessidade do *gattacker* ou do aplicativo.

Para isto, ainda no *gattacker*, o arquivo deverá ser convertido para o formato XML e exportado para a ferramenta *nRFConnect*. No *nRFConnect*, deverá ser realizada a comunicação com a placa e executado o arquivo XML, realizando o ataque e atingindo o objetivo almejado.

Ataque Livre-de-Injeção

Foi visto que, utilizando técnicas de sniffing com o propósito de coletar os dados cruciais a respeito da comunicação entre dois dispositivos, pode-se abrir portas para ataques mais complexos. Um exemplo disto é visto na ferramenta Crackle, explorada no trabalho de Mike Ryan em [27], para inferência das chaves criptográficas utilizada em uma conexão. Com elas, é possível obter os dados necessários para comprometer a comunicação segura entre dois pontos e até estabelecer o controle sobre os dados trocados pelos dispositivos envolvidos. Para adicionar mais uma camada protetiva ao sigilo das mensagens trocadas na comunicação, a funcionalidade de vinculação (*bonding*) foi criada.

Quando dois dispositivos são pareados e vinculados, uma chave de longo prazo (*LTK - Long Term Key*) é gerada e armazenada em ambos os pontos da conexão. Esta chave é utilizada para criptografar toda a comunicação realizada entre os dispositivos após a primeira conexão, e é trocada somente após um período de tempo estabelecido em uma das pontas ou caso, por algum motivo específico (uma reinicialização da memória do equipamento, por exemplo).

Com esta funcionalidade protetiva, ataques como os realizados por ferramentas como a Crackle, citada anteriormente, passam a se tornar incapazes de interceptar os dados criptografados pela LTK gerada na vinculação, já que, ao iniciar uma conexão entre dois dispositivos vinculados, os dispositivos não trocam dados para a geração de uma nova chave. Sem estes dados, não é possível inferir a chave utilizada na criptografia dos dados, e, como consequência, a integridade da comunicação se mantém. Ryan ainda propõe técnicas para combater a funcionalidade de vinculação utilizando injeção de pacotes, personificando um dispositivo e enviando pacotes informando que não se possui mais a LTK. Ao receber esta informação, o outro dispositivo tem a opção de realizar uma nova vinculação, trocando os dados necessários para a inferência da chave pelo usuário malicioso.

Devido a necessidade de precisão na injeção de pacotes e, levando em consideração os cenários em que o ataque pode ser realizado, o atacante pode não obter exito em forçar os dispositivos a gerar novas chaves. Neste caso, quando a comunicação entre as partes segue normalmente, o usuário malicioso pode realizar *jamming* para derrubar a conexão e forçar uma reinicialização, criando uma nova oportunidade para injetar os pacotes necessários. Devido a dependência de condições físicas do cenário e possibilidade de detecção de comportamento anômalo causado por ataques de *jamming*, esta abordagem pode não ser a melhor indicada para burlar a proteção fornecida pela funcionalidade de vinculação.

Considerando tais variáveis, um novo ataque foi desenvolvido no Laboratório de Redes da Universidade Federal da Paraíba onde técnicas de injeção de pacotes, assim como *jamming*, não são necessárias para invalidar as chaves trocadas no processo de vinculação. Este técnica é intitulada como Ataque Livre-de-Injeção (*Injection-free*) e é detalhada em [14].

A ideia do ataque *Injection-free* é explorar as limitações de dispositivos IoT referentes a capacidade de memória, populando todas as posições livres da lista. O dispositivo alvo irá gerenciar a lista de vínculos de acordo com a implementação estabelecida, podendo prosseguir com a comunicação de novos dispositivos sem a realização de vínculos, deletar vínculos antigos para estabelecer novos com dispositivos não vinculados, ou negar serviço para dispositivos que não possuem dados na lista de vínculos.

Uma abstração da comunicação realizada no ataque pode ser vista na Figura 5.11, na qual o cenário em questão considera que o dispositivo alvo, que possui 3 vagas em sua lista de vínculos, remove o dispositivo mais antigo da lista para vincular novos dispositivos. O dispositivo legitimo é removido e, ao retornar, precisa passar por um novo processo de pareamento e vinculação, se expondo aos ataques discutidos por Mike Ryan em [27]. Nota-se para realizar este ataque, o atacante precisa de endereços diferente a cada vínculo. Abrem-se a possibilidade de conexões via múltiplas interfaces ou com uma única interface realizando personificação de endereços diferentes.

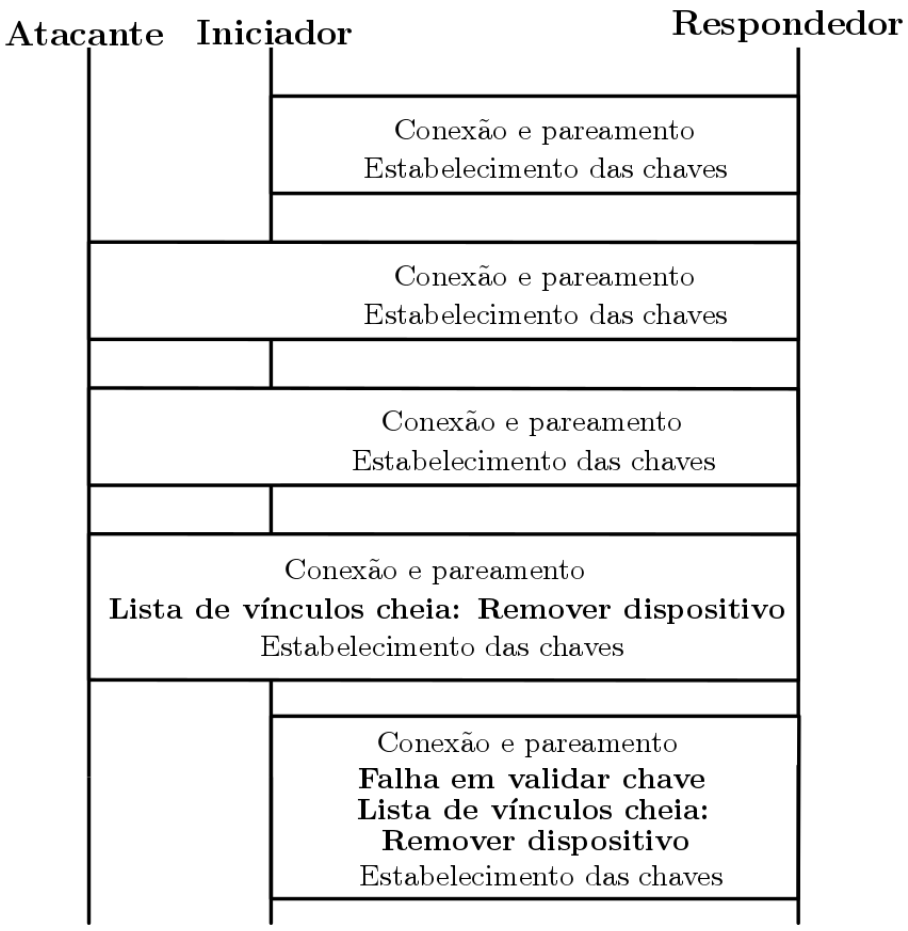


Figura 5.11 – Diagrama do ataque Livre-de-Injeção. Adaptado de [14].

O caso demonstrado na Figura 5.11 é somente um dos possíveis cenários do ataque

Livre-de-Injeção. Como dito anteriormente, o fluxo a ser seguido pelo atacante irá variar de acordo com a implementação do gerenciamento da lista de vinculação do dispositivo. O desenvolvedor possui três meios de gerenciar conexões uma vez que a lista de vínculos está cheia:

- **Deletando dados**: Ao chegar uma nova conexão, um dispositivo será removido da lista de vínculos para inserção do endereço e chave do dispositivo iniciador da conexão;
- **Pareando sem vínculo**: Novas conexões são realizadas, mas dispositivos não fazem vinculação. Ao invés disto, novas chaves são geradas a cada conexão, e todos os dispositivos que não encontram-se presente na lista de vínculos precisarão gerar chaves a cada conexão;
- **Negação de Serviço**: Novas conexões não são realizadas. Somente dispositivos que possuem registro na lista de vínculos podem se comunicar com o dispositivo alvo.

Porém, deve-se ter em mente que não existe um método definido para melhor combater o ataque Livre-de-Injeção. Caso a metodologia de deleção de dados seja utilizada, um atacante pode tentar popular a lista de vínculos até que todos os dispositivos legítimos sejam removidos da lista. De maneira análoga, caso novas conexões sejam criadas sem vinculação após o preenchimento completo da lista, novos dispositivos estarão completamente desprotegidos contra ataques de quebra de chaves de curto prazo.

Finalmente, caso serviço seja negado para novos dispositivos, ao preencher a lista de vínculos, o atacante efetivamente inutiliza o dispositivo alvo para novas conexões. Num cenário em que o atacante tem acesso aos dispositivos alvos antes de serem adquiridos pelas vítimas (por exemplo, numa loja), o dispositivo poderá ser inutilizado antes mesmo de qualquer conexão legítima ser realizada.

Cabe ao desenvolvedor dos periféricos escolher o melhor método para proteção de seus dispositivos de acordo com os níveis de criticidade da aplicação. Na Seção 5.1.5 abordamos contramedidas que podem ser utilizadas para mitigar a progressão do ataque Livre-de-Injeção.

A demonstração do ataque, exibida em detalhes em [14], consiste da utilização de um dispositivo alvo com uma lista de vínculos capaz de armazenar 4 dispositivos. Um dispositivo legítimo se vincula ao dispositivo alvo e, em seguida, um usuário malicioso com quatro interfaces BLE tenta vincular todas com o alvo do ataque. Um modelo teórico do ataque pode ser visto na Figura 5.12.

As três estratégias de gerenciamento da lista de vínculos citadas anteriormente foram implementadas, testadas e amplamente dissertadas em [14]. A critério de exemplo, pode-se visualizar na Figura 5.13. Na Figura 5.13 (a), é possível ver a lista de vínculos após a conexão do dispositivo legítimo, populando a lista com seu endereço. Na Figura 5.13 (b), vê-se a lista de vínculos após a conexão de três interfaces do usuário malicioso, preenchendo por completo as vagas na lista de vínculos. Na Figura 5.13 (c), vê-se o que ocorre quando um novo dispositivo tenta se conectar com o dispositivo alvo, resultando impossibilidade de finalizar o processo de autenticação por falha no processo de vinculação.

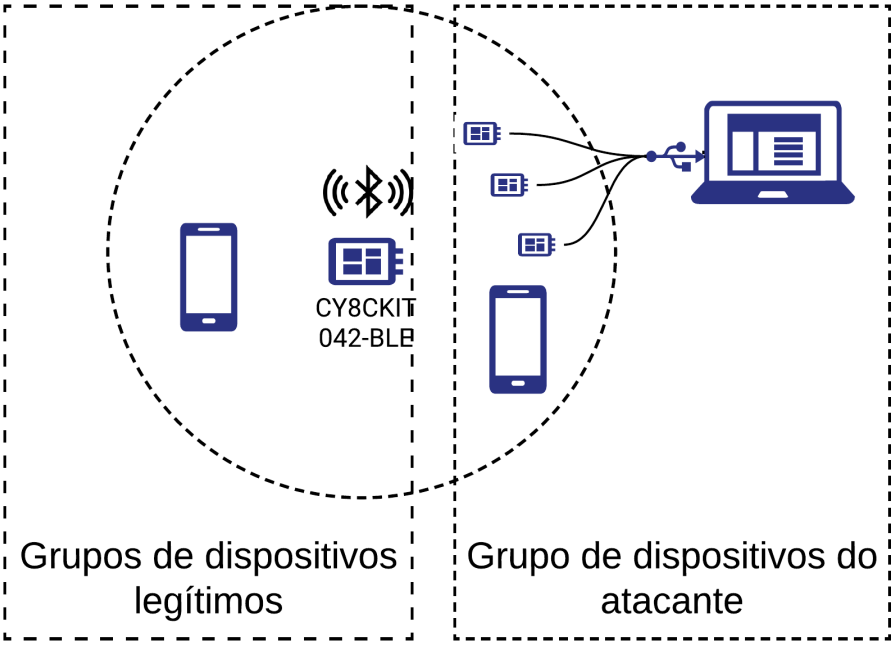
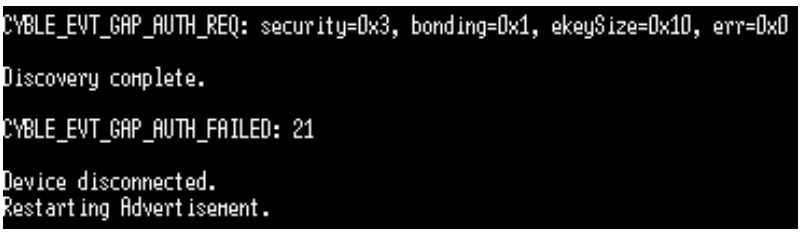


Figura 5.12 – Layout dos dispositivos utilizados no ataque Livre-de-Injeção. Adaptado de [14].

(a) Conexão do primeiro dispositivo

(b) Lista de vínculos cheia



(c) Tentativa de conexão após preenchimento da lista

Figura 5.13 – Ataque Livre-de-Injeção em um dispositivo com 4 vagas na lisa de vínculos. Imagem de [14].

Para a simulação do caso da imagem, nenhuma forma de gerenciamento da lista de vínculos foi implementada. A consequência disso é a negação de serviço para todos os dispositivos que já não possuem endereços armazenados na lista vínculos, potencialmente inutilizando o dispositivo para novos usuários.

	Vulnerabilidades	Contramedidas
Anúncios	*Jamming*; Personificação; Ataques de replicação; Ataques de retransmissão.	Isolamento do ambiente; Uso de *nonces*; Delimitação de distância.
Conexão	Interceptação de pacotes; Ataques de força bruta; Injeção de pacotes.	Troca de chave segura; Módulos de comunicação atualizados.
Comunicação	*Jamming*; Interceptação de pacotes; Personificação de dispositivos.	Criptografia em nível de aplicação; Utilização de certificação.
Descoberta de Serviços	Escaneamento de serviços expostos.	Autenticação para descoberta; Desenvolvimento seguro.
Uso de; Serviços	Personificação de dispositivos.	Autenticação para uso; Desenvolvimento seguro.
Detecção de; dispositivos	Ataques de força bruta; Detecção indesejada.	Desativação do módulo de comunicação; Anúncio de presença e conexão usando endereços aleatórios.

Tabela 5.1 – Principais vulnerabilidades e suas contramedidas nas funcionalidades compartilhadas de dispositivos IoT.

Contramedidas

Com base nas observações a respeito das vulnerabilidades e ataques discutidos nas seções anteriores, foi realizado um levantamento de defesas genéricas que podem ser utilizadas para mitigar tais falhas de segurança. O propósito da genericidade destas defesas encontra-se na sua aplicabilidade em diferentes cenários nos quais o paradigma IoT é utilizado. Estas defesas variam desde boas práticas de desenvolvimento até técnicas consolidadas na literatura. Devido as restrições enfrentadas pelos dispositivos para realizar suas atividades, seguir as regras dos protocolos nestes casos significa reduzir drasticamente a lista de possíveis ações de defesas que podem ser tomadas. Já a correção de algumas vulnerabilidades aqui discutidas poderiam inferir em mudanças nos protocolos utilizados, o que, a curto prazo, é uma medida inviável.

Inicialmente, um processo de Engenharia de Software bem executado é capaz de solucionar grande parte dos problemas relacionados í ataques relativos a privacidade, como a identificação de funcionalidades e até mesmo a presença do dispositivo. Isto se deve ao fato de que parte destes problemas são derivados de decisões infelizes do time de desenvolvedores, sejam com o propósito de economizar recursos ou na tentativa de otimizar operações. Decidir quando oferecer uma certa funcionalidade a um determinado usuário, assim como o nível de autenticação necessária para se acessar esta, é fundamental para a segurança dos dispositivos.

No caso de ataques sobre o processo de anúncio da presença de equipamentos e serviços,

ataques de replicação podem ser combatidos com a utilização de *nonces* para garantir a *freshness*, ou frescura, das mensagens, como dito por P. Syverson em [28], enquanto ataques de retransmissão podem utilizar técnicas de estimação de tempo de ida e volta de uma mensagem para detectar a proximidade real do usuário, como dito por Saar Drimer et al em [29]. O combate aos *jammers*, porém, é mais complexo, uma vez que, com uma quantidade relativamente pequena de recursos, um atacante pode bloquear os canais de anúncio com facilidade fazendo uso de *jammers* reativos. Recomenda-se que o usuário busque e neutralize a fonte do ataque ou tente operar seu dispositivo em outro ambiente. Tem-se em mente também que *jammings* são consideravelmente mais fáceis de se detectar, visto que uma interrupção de comunicação em um ou mais canais poderá vir a causar uma disrupção no funcionamento de todos os dispositivos usuários da tecnologia afetada na região. Nota-se, também, que em casos onde o causador da disrupção procura agir de maneira mais sutil, *jammers* irão agir em canais de comunicação isolados, abrindo espaço para defesas envolvendo saltos de canais como abordado por Popper et al. [30].

No processo de conexão, deve-se escolher um método de geração e troca de chaves que não possa ser vítima de ataques para evitar o comprometimento da conexão. Para isso, pode-se delegar a funcionalidade de pareamento para outra tecnologia, como ocorre ao utilizar o método de pareamento OOB no caso do Bluetooth. Outra opção consiste no uso de versões atualizadas e com metodologias de troca de chave seguras. Contextualizando nos protocolos aqui debatidos, o Bluetooth em sua versão 4.2 introduz um novo método de pareamento em conjunto com a funcionalidade LE *Secure Connections*, como descrito por Kai Ren em [31]. Esta nova funcionalidade faz uso do algoritmo de Curva Elíptica de Diffie-Hellman para trocar as chaves utilizadas, o que substitui os métodos de geração e trocas de chaves usadas nas versões anteriores, agora chamadas de LE *Legacy pairing*. Um desafio é identificar módulos *Bluetooth* no mercado que utilizam destas versões atualizadas, uma vez que a abundí¢ncia de dispositivos com versões anteriores é grande.

Quanto a proteção contra ataques de interceptação passiva e ativa, deve-se ter em mente que as vulnerabilidades derivadas da capacidade de interceptação das mensagens por terceiros surgem do processo de geração de chaves comprometido. Desta forma, são reforçadas as recomendações de segurança feitas a respeito do processo de conexão/pareamento. Contra ataques do tipo *Man-in-the-Middle*, porém recomenda-se técnicas mais avançadas como, por exemplo, o uso de certificação.

O uso de certificação digital garantirá que o usuário malicioso não conseguirá simplesmente fazer uso das mensagens obtidas de um dos pontos da conexão para iludir o outro. Boneh e Franklin descrevem como isto se deve ao fato de que o certificado de um dispositivo pode ser utilizado como parte do processo de encriptação das mensagens enviadas para este, fazendo que somente o dispositivo seja capaz de decriptá-las e assegurando que, desde o princípio, a comunicação será segura [32].

Com base nesta análise de ataques sobre a tecnologia e defesas encontradas durante esta pesquisa, foi possível desenvolver uma relação de vulnerabilidades e opções de contramedidas

que podem ajudar a estabelecer a segurança contra explorações nos dispositivos. Esta relação pode ser vista na Tabela 5.1.

Para um documento mais detalhado a respeito de mitigação de vulnerabilidades e proteções especificamente na tecnologia Bluetooth, recomendamos o trabalho do Instituto Nacional de Padrões de Tecnologia dos Estados Unidos (NIST - *National Institute of Standards and Technology*) [10].

Com relação a possíveis mitigações do ataque Livre-de-Injeção, temos consciência de que o desenvolvedor do software utilizado pelo dispositivo BLE tem controle sobre como a lista de vínculos é gerenciada quando cheia.

A princípio, negar a conexão de novos dispositivos é uma forma de garantir o serviço para dispositivos já conectados, porém deve-se ter em mente a possibilidade de inutilização do dispositivo alvo para novos dispositivos legítimos após a realização do ataque. De forma análoga, caso o usuário malicioso consiga realizar o ataque em um dispositivo antes da conexão do qualquer usuário legítimo, nenhum dispositivo conseguira se conectar com o alvo do ataque antes da realização de uma reinicialização de sua memória.

A simples opção de deleção de dados se torna uma opção para evitar a criação do cenário de negação de serviços, porém abre espaço para os ataques discutidos em [27]. Para evitar que o usuário malicioso tenha a posse dos dados necessários para descriptografar as mensagens trocadas pelos dispositivos, um usuário legítimo precisaria realizar o processo de pareamento e vínculo em um ambiente completamente controlado, evitando a presença de dispositivos de terceiros próximos o suficiente para capturar os dados das mensagens trocadas no processo de pareamento.

Uma metodologia de deleção pseudoaleatória de entidades na lista de vínculos é uma opção mais sofisticada e já utilizada em outros casos como por Yuri Gil Dantas et al em [33] e por Marcilio Lemos em [34]. Seguindo esta estratégia, considerando que a quantidade de dispositivos maliciosos presentes na lista de vínculos é maior que a quantidade de legítimos, a probabilidade de remoção de um dispositivo malicioso também é maior, caso um seja escolhido aleatoriamente. A estratégia de defesa também pode ser aprimorada fazendo uso de variáveis de peso, atribuindo, por exemplo, uma maior probabilidade de remoção para um dispositivo que está na lista a menos tempo.

Conclusão

Neste capítulo foram apresentadas definições para melhor entendimento das tecnologias seguidoras do paradigma IoT, bem como um breve panorama das vulnerabilidades que existem nas mesmas, com enfase no funcionamento do padrão de comunicação *Bluetooth Low Energy* (BLE). Como o nome do capítulo sugere, o trabalho teve como foco a segurança em aplicações de Internet da Coisas (*IoT - Internet of Things*).

Após a fundamentação teórica do cenário IoT atual e suas principais vulnerabilidades, foi realizada uma apresentação superficial do protocolo *Bluetooth*, explorando sua gama

de dispositivos e os usos dessa tecnologia. Com todos esses conceitos apresentados, foram então demonstrados ataques a dispositivos reais com vulnerabilidades exploráveis. Por fim, este trabalho apresenta as principais contramedidas í s vulnerabilidades apresentadas.

Por este ser um trabalho introdutório e expositivo, recomendamos a leitura de [3] por Jayasree Sengupta et al, e [8] por Kolias Kambourakis et al, para um melhor entendimento do cenário IoT. Para o foco no estudo do padrão BLE e suas vulnerabilidades, recomendamos o documento [10] do NIST (National Institute of Standards and Technology), e [35] por A. M. Lonzetta et al.

Referências Bibliográficas

[1] Mass Framingham. Worldwide spending on the internet of things. https://www.idc.com/getdoc.jsp?containerId=prUS46609320. Acessed: September 21, 2020.

[2] Number of internet of things (iot) connected devices worldwide in 2018, 2025 and 2030. https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/. Acessed: August 19, 2020.

[3] Jayasree Sengupta, Sushmita Ruj, and Sipra Das Bit. A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot. *Journal of Network and Computer Applications*, 149:102481, 2020.

[4] A Ray, V Raj, M Oriol, A Monot, and S Obermeier. Bluetooth low energy devices security testing framework. *2018 Ieee 11th International Conference on Software Testing, Verification and Validation (Icst)*, pages 384–393, 2018.

[5] M. Langone, R. Setola, and J. Lopez. Cybersecurity of wearable devices: An experimental analysis and a vulnerability assessment method. 2:304–309, 2017.

[6] K. Saravanan, L. Vijayanand, and R. Negesh. A novel bluetooth man-in-the-middle attack based on ssp using oob association model. 03 2012.

[7] Chunling Sun. Application of rfid technology for logistics on internet of things. *AASRI Procedia*, 1:106 – 111, 2012. AASRI Conference on Computational Intelligence and Bioinformatics.

[8] Geneiatakis Karopoulos Makrakis Kambourakis, Kolias and T Kounelis. A state-of-the-art review on the security of mainstream iot wireless pan protocol stacks. *Symmetry*, 2020.

[9] Attack landscape report h1 2019. https://blog.f-secure.com/attack-landscape-h1-2019-iot-smb-traffic-abound/. Acessed: August 22, 2020.

[10] John Padgette, John Bahr, Mayank Batra, Marcel Holtmann, Rhonda Smithbey, Lidong Chen, and Karen Scarfone. Guide to bluetooth security. *NIST Special Publication*, 800:121, 2017.

[11] Wireshark home page. https://www.wireshark.org. Acessed: October 13, 2020.

[12] Francillon A. Capkun S., Danev B. Relay attacks on passive keyless entry and start systems in modern cars. 2010.

[13] Pradip M. Jawandhiya, Mangesh Ghonge, and Ali. A survey of mobile ad hoc network attacks. *International Journal of Engineering Science and Technology*, pages 4063–4071, 2010.

[14] Aellison C. T. Santos, José L Soares Filho, Ávilla Ítalo S Silva, Vivek Nigam, and Iguatemi E. Fonseca. BLE Injection-free Attack: a Novel Attack on Bluetooth Low Energy Devices. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–11, 2019.

[15] Tal Melamed. An active man-in-the-middle attack on bluetooth smart devices. *International Journal of Safety and Security Engineering*, 8:200–211, 02 2018.

[16] Jaap Haartsen, Mahmoud Naghshineh, Jon Inouye, Olaf J. Joeressen, and Warren Allen. Bluetooth: Vision, goals, and architecture. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2(4):38–45, October 1998.

[17] Matheus E. Garbelini, Chundong Wang, Sudipta Chattopadhyay, Sun Sumei, and Ernest Kurniawan. Sweyntooth: Unleashing mayhem over bluetooth low energy. pages 911–925, July 2020.

[18] Bluetooth SIG. *Bluetooth Core Specification*. Editora Encanamentos, 2019.

[19] Shaikh Shahriar Hassan, Soumik Das Bibon, Md Shohrab Hossain, and Mohammed Atiquzzaman. Security threats in bluetooth technology. *Comput. Secur.*, 74(C):308 – 322, May 2018.

[20] K. Chang. Bluetooth: a viable solution for iot? [industry perspectives]. *IEEE Wireless Communications*, 21(6):6–7, 2014.

[21] Leandro Pessoa. Introdução ao bluetooth smart (ble). https://www.embarcados.com.br/bluetooth-smart-ble/. Acessed: August 30, 2020.

[22] R. Want, B. Schilit, and D. Laskowski. Bluetooth le finds its niche. *IEEE Pervasive Computing*, 12(4):12–16, 2013.

[23] Ubuntu manpage repository - hcitool. http://manpages.ubuntu.com/manpages/focal/en/man1/hcitool.1.html. Acessed: October 12, 2020.

[24] Ubuntu manpage repository - gatttool. http://manpages.ubuntu.com/manpages/focal/en/man1/gatttool.1.html. Acessed: October 12, 2020.

[25] Sławomir Jasek. Gattacking bluetooth smart devices. In *Black Hat USA Conference*, July 2016.

[26] Aellison C. T. Santos, Ávilla Ítalo S Soares Filho, Iguatemi E. Fonseca, and Vivek Nigam. Sobre Os Impactos das Vulnerabilidades Presentes na Tecnologia Bluetooth Low Energy. 2019.

[27] Mike Ryan. Bluetooth: With low energy comes low security. In *Presented as part of the 7th USENIX Workshop on Offensive Technologies*, 2013.

[28] Paul Syverson. A taxonomy of replay attacks [cryptographic protocols]. In *Computer Security Foundations Workshop VII, 1994. CSFW 7. Proceedings*, pages 187–191. IEEE, 1994.

[29] Saar Drimer and Steven J Murdoch. Keep your enemies close: Distance bounding against smartcard relay attacks. In *USENIX security symposium*, volume 312, 2007.

[30] Christina Popper, Mario Strasser, and Srdjan Capkun. Anti-jamming broadcast communication using uncoordinated spread spectrum techniques. *ieee journal on selected areas in communications*, 28(5):703–715, 2010.

[31] Kai Ren. Bluetooth pairing part 4: Le secure connections - numeric comparison. https://blog.bluetooth.com/bluetooth-pairing-part-4?_ga=2.99221656.724766273.1510512070-1121671378.1510512070. Acessed: September 21, 2020.

[32] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO 2001*, pages 213–229. Springer, 2001.

[33] Yuri Gil Dantas, Vivek Nigam, and Iguatemi E. Fonseca. A Selective Defense for Application Layer DDoS Attacks. In *JISIC 2014*, pages 75–82, 2014.

[34] Marcilio Lemos, Yuri Gil Dantas, Iguatemi Fonseca, Vivek Nigam, and Gustavo Sampaio. A selective defense for mitigating coordinated call attacks. In *34th Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*. sn, 2016.

[35] AM Lonzetta, P Cope, J Campbell, BJ Mohd, and T Hayajneh. Assessing vulnerabilities in bluetooth low energy (ble) wireless network based iot systems. *IEEE 2nd International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, IEEE International Conference on Intelligent Data and Security*, 2016.