

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC (RES_020/2016 – 2024_2)		
() PRÉ-PROJETO	(X) PROJETO	ANO/SEMESTRE: 2024/2

PROTÓTIPO PARA MONITORAMENTO INTELIGENTE DE VAGAS DE ESTACIONAMENTOS UTILIZANDO REDE EM MALHA SEM FIO COM O PROTOCOLO THREAD

Leonardo de Oliveira

Prof. Miguel Alexandre Wisintainer – Orientador

1 INTRODUÇÃO

Ao longo dos últimos anos, a quantidade de veículos em circulação no Brasil está em constante crescimento. No estado de Santa Catarina, a frota de veículos cresceu de cerca de 2,3 milhões em 2010 para mais de 4 milhões de automóveis em 2023 (DETRAN-SC, 2024). Uma das consequências desse crescimento é o aumento da dificuldade e tempo para encontrar vagas livres nos estacionamentos. Uma das soluções para isso, segundo Orrie, Silva e Hancke (2015), é a construção ou expansão de estacionamentos ou a criação de vagas de estacionamentos automatizadas. Entretanto, ambas as soluções são caras. Uma alternativa é a utilização mais eficiente do espaço já existente, por exemplo, auxiliando os motoristas a encontrarem vagas mais facilmente.

Na busca de encontrar uma solução para a utilização mais eficiente dos estacionamentos, algumas pesquisas e soluções comerciais já foram desenvolvidas, como o monitoramento de ocupação por tickets ou cartões, monitoração de vagas ocupadas por sensores para exibição em painéis ou luzes indicadoras sobre as vagas. Entretanto estas soluções, em sua maioria, não são precisas o suficiente para auxiliar os motoristas ou são muito custosas e necessitam de uma instalação complexa e uma infraestrutura planejada, que muitas vezes inviabilizam esse tipo de solução para estacionamentos menores.

Mais recentemente, com o grande crescimento da indústria de semicondutores, o desenvolvimento de dispositivos cada vez menores e com capacidades de comunicação sem fio têm se tornado cada vez mais viável. Estes avanços possibilitaram o desenvolvimento de novas tecnologias para a Internet das Coisas (Internet of Things – IoT) e disponibilizaram novas possibilidades mais eficientes e baratas para aplicações de sensoramento, com dispositivos de baixo custo e consumo de energia.

A comunicação por redes sem fio é mais prática e mais flexível comparada a conexões via cabo e, junto com o crescimento de dispositivos móveis e sem fio, tem se tornado bastante acessível. Uma topologia de rede cada vez mais empregada em aplicações IoT são as redes em malha (*mesh*) sem fio, que são redes onde alguns de seus nós podem atuar como roteadores ou repetidores, fazendo com que todos os nós da rede estejam interconectados. Uma das vantagens dessa rede é a maior cobertura e confiabilidade, por reduzir pontos únicos de falha da rede.

Diante deste cenário, este trabalho se propõe a desenvolver um protótipo de baixo custo utilizando IoT e redes *mesh* sem fio com o protocolo Thread para o sensoramento e monitoramento inteligente de vagas de estacionamento. Com isso, se espera com este protótipo a redução no tempo de procura por vagas e maior conforto para os motoristas, além de fomentar a utilização de redes *mesh* para a interconexão mais eficiente em dispositivos IoT, mesmo fora do contexto residencial.

1.1 OBJETIVOS

O objetivo principal deste trabalho é desenvolver um sistema embarcado de baixo custo baseado em IoT e redes *mesh* sem fio com o protocolo Thread, integrando hardware e software para o sensoramento de ocupação de vagas de estacionamento.

Os objetivos específicos são:

- a) disponibilizar uma aplicação *back-end*, executada em um servidor, para agregar e fornecer dados confiáveis e precisos sobre as vagas;
- b) utilizar uma tecnologia de comunicação em tempo real para integrar o servidor com uma interface (*front-end*) onde clientes podem acompanhar o estado das vagas;
- c) implementar uma rede *mesh* sem fio utilizando o protocolo Thread com dispositivos ESP32-C6 para a transmissão de dados entre os dispositivos e a aplicação *back-end*;
- d) testar e avaliar o desempenho e tempo de resposta da rede para a proposta do protótipo;
- e) desenvolver um protótipo funcional capaz de mostrar a viabilidade e funcionamento do sistema.

2 TRABALHOS CORRELATOS

Nesta seção são apresentados os trabalhos correlatos com características semelhantes ao trabalho proposto e está dividida em duas subseções. Na subseção 2.1 é abordado a revisão sistemática e na subseção 2.2 é apresentado a síntese dos trabalhos correlatos, selecionados com a maior correlação e relevância ao trabalho proposto.

2.1 REVISÃO SISTEMÁTICA

Nesta subseção são apresentados os trabalhos correlatos que contém características semelhantes ao trabalho proposto no Quadro 1. Todos os trabalhos selecionados possuem alguma relação com IoT, sensores de vagas de estacionamento ou redes *mesh*.

O processo de pesquisa foi realizado nos seguintes portais de busca: Google Acadêmico e IEEE Xplore, e Scientific Electronic Library On-Line (SciELO). Durante a busca, a preferência foi de trabalhos de conclusão de curso, monografias e dissertações e os critério principal foi a semelhança ao trabalho atual com base no período de 2015 e 2024. Dos portais citados, apenas o Google Acadêmico e IEEE Xplore apresentaram resultados relevantes.

As palavras chaves utilizadas para a pesquisa foram: “esp32 mesh protocol”, “mesh” AND (“iot” OR “internet das coisas”), “mesh” AND (“iot” OR “internet das coisas”) AND (“estacionamento” OR “parking”) e “iot” AND “parking spot”. Optou-se pelos filtros de pesquisa com termos em inglês e português para facilitar a busca de trabalhos relacionados.

Quadro 1 - Síntese dos trabalhos correlatos selecionados

Local	Assunto	Filtro	Referência
Google Acadêmico	Evaluation Method of Mesh Protocol over ESP32 and ESP8266	esp32 mesh protocol	Afife <i>et al.</i> (2021)
	Comparativo de Protocolos de IoT para Automação Residencial: Potenciais Vulnerabilidades e Sugestões de Melhorias		Correa (2018)
	Um protótipo de sistema inteligente para monitoramento de vagas de estacionamentos – Uma aplicação baseada em Internet das Coisas	"mesh" AND ("iot" OR "internet das coisas")	Ronsani (2018)
	Sistema Multiagente baseado numa Rede Mesh		Silva (2022)
	Protótipo de sistema para monitoramento de vagas de estacionamento utilizando Internet das Coisas e comunicação por LoRa	"mesh" AND ("iot" OR "internet das coisas") AND ("estacionamento" OR "parking")	Goldner (2022)
	Rede em malha sem fio utilizando o dispositivo ESP-8266: Aplicação e avaliação de desempenho em cenário de detecção de presença de veículos em estacionamentos		Souza (2023)
	Sistema inteligente de gerenciamento e controle de vagas de estacionamento		Ferreira (2021)
	Secure Outdoor Smart Parking using Dual Mode Bluetooth Mesh Networks		Seymer, Wijesekera e Kan (2019)
IEEE Xplore	IoT-Based Smart Parking Management System Using ESP32 Microcontroller	"iot" AND "parking spot"	Simatupang, Lubis e Vincent (2022)
	Implementation of a smart parking system using LoRa, STM32 Board & IoT		Malik <i>et al.</i> (2023)
	A Wireless Smart Parking System	“iot” AND “parking system”	Orrie, Silva e Hancke (2015)

Fonte: elaborado pelo autor (2024).

Durante a pesquisa, além da semelhança com o objetivo proposto, a data de publicação e variedade de diferentes abordagens sobre o problema, com diferentes tecnologias, metodologias e resultados em diversos cenários, também foram pontos importantes para a escolha e para disponibilizar uma visão mais ampla sobre o

tema escolhido. Diante disso, Correa (2018) se destaca por fazer um comparativo de diversos protocolos conhecidos de IoT enquanto Goldner (2022) se destaca pela utilização da rede LoRa de uma forma pouco utilizada em outros trabalhos e pela criação de um *gateway* com o ESP32. Por fim, os trabalhos de Ronsani (2018), Ferreira (2021) e Souza (2023) foram selecionados por conta da similaridade do objetivo do protótipo proposto e suas variadas formas de implementação e resultados.

2.2 SÍNTESE DOS TRABALHOS CORRELATOS

A seguir, conforme os trabalhos selecionados no Quadro 1, são expostos os trabalhos mais próximos dos objetivos centrais deste estudo. O primeiro, desenvolvido por Ronsani (2018), exibido no Quadro 2 é um sistema completo de detecção de veículos em vagas de estacionamento, incluindo uma aplicação de usuário visual com atualizações em tempo real e uma aplicação servidor. O segundo trabalho apresentado no Quadro 3 desenvolvido por Ferreira (2021), tem como objetivo principal a detecção de veículos e a criação de um sistema mais robusto e *dashboards* de gerenciamento e visualização do estado do estacionamento. Por último, o Quadro 4 apresenta o estudo de Souza (2023) que aborda, principalmente, o funcionamento e comportamento de redes *mesh* em diversos cenários.

Quadro 2 – Trabalho Correlato 1

Referência	Ronsani (2018)
Objetivos	Desenvolver um sistema de gerenciamento e detecção de veículos em vagas de estacionamento com uma rede <i>mesh</i> sem fio, uma aplicação de servidor para integrar as informações dos sensores e uma aplicação com interface visual amigável.
Principais funcionalidades	Deteção de carros nas vagas, página web para exibir os estados das vagas em tempo real e conexão sem fios por longas distâncias.
Ferramentas de desenvolvimento	Arduino IDE, C++, microcontroladores ESP8266 e ESP32, bibliotecas <i>painlessMesh</i> , <i>ESPAsyncWebServer</i> e <i>WebSocketsClient</i> , sensor HC-SR04P, Node.js, JavaScript, Express e MongoDB.
Resultados e conclusões	O sistema desenvolvido se mostrou eficiente para os objetivos propostos, com algumas vantagens sobre os produtos comerciais, como a aplicação web. Além disso a escolha por uma topologia <i>mesh</i> se mostrou eficiente e robusta para aplicações IoT com vários nós conectados espalhados por uma área que, sem essa topologia, estariam fora do alcance da rede Wi-Fi.

Fonte: elaborado pelo autor (2024).

Ronsani (2018) tem um objetivo mais geral e prático sobre seu trabalho. Tem como foco principal o desenvolvimento de um sistema completo simples, funcional, escalável e em tempo real, porém mostra poucos detalhes, principalmente do software desenvolvido e da rede implementada. Apesar disso, mostra uma solução prática de um sistema de gerenciamento e detecção dos veículos, uma aplicação de servidor para processar os dados, uma interface visual e a utilização de uma rede *mesh* em um cenário de envio de mensagens por um nó sem conexão direta com a rede Wi-Fi.

Quadro 3 – Trabalho Correlato 2

Referência	Ferreira (2021)
Objetivos	Desenvolver um sistema de gerenciamento de vagas de estacionamento em ambientes fechados utilizando redes Wi-Fi, um banco de dados online, uma aplicação visual para exibir informações do estacionamento em tempo real, ser compacto e de fácil instalação.
Principais funcionalidades	Indicador visual com luzes acima do veículo, aplicação de gerenciamento do sistema de vagas, informações relevantes do sistema em tempo real e uma aplicação para visualização voltada para o usuário final.
Ferramentas de desenvolvimento	Arduino IDE, C++, Microcontrolador ESP32, sensor HC-SR04, Node-RED para programação visual, MySQL, PhpMyAdmin, Fritzing, LEDs RGB, bibliotecas Wi-Fi Multi, WebServer, 8266WebServer e WiFiManager.
Resultados e conclusões	No quesito interface gráfica, foi desenvolvido uma plataforma de simples de gerenciamento, visualização de informações relevantes sobre o estacionamento e das vagas. A plataforma Node-RED se mostrou útil para facilitar a implementação da aplicação do servidor e das <i>dashboards</i> . O autor também comenta que o projeto se mostra escalável, compacto e com potencial comercial. Além disso, sugere como melhoria a implementação de redes <i>mesh</i> para maior alcance e escalabilidade.

Fonte: elaborado pelo autor (2024).

O trabalho de Ferreira (2021) foca principalmente na aplicação em si, ou seja, uma aplicação *back-end* e *front-end* amigável com mais funcionalidades, como *dashboards*, estatísticas e gráficos sobre o estado do estacionamento, visualização do estado das vagas em tempo real e a construção de um protótipo mais robusto, que também conta com um indicador visual com LEDs que ficam posicionados acima do veículo. Entretanto, assim como o autor relata em suas conclusões, o projeto desenvolvido tem uma certa limitação por utilizar uma rede Wi-Fi comum, que limita o alcance do protótipo.

Quadro 4 – Trabalho Correlato 3

Referência	Souza (2023)
Objetivos	Implementar uma rede <i>mesh</i> utilizando o microcontrolador ESP8266, testar e avaliar o funcionamento da rede em longa distância e suas particularidades e prover dados precisos e confiáveis na aplicação de detecção de ocupação de vagas em estacionamentos.
Principais funcionalidades	Deteção de presença de veículos, comunicação sem fio por longas distâncias, LEDs indicadores e uma página web simples para verificar a disponibilidade da vaga.
Ferramentas de desenvolvimento	Arduino, Microcontrolador ESP8266, sensor HC-SR50, bibliotecas <i>painlessMesh</i> , <i>Esp8266Wifi</i> , <i>WiFiClient</i> , <i>ESP8266WebServer</i> e <i>ESP8266mDNS</i> , Visual Studio Code, PlatformIO.
Resultados e conclusões	O autor conclui que as WMNs são uma ferramenta importante para aplicações IoT por permitirem a conexão entre dispositivos de forma descentralizada e flexível. Os testes realizados mostraram que é possível implementar uma WMN com dispositivos ESP8266 eficientes para os objetivos definidos. Por fim, o autor sugere a comparação de eficiência energética e a possível integração com outras tecnologias com a finalidade de melhorar a comunicação e coleta de dados.

Fonte: elaborado pelo autor (2024).

O estudo de Souza (2023) tem como objetivo central a experimentação, implementação e avaliação de uma rede *mesh* utilizando dispositivos ESP8266. O trabalho realiza testes em vários cenários, com a finalidade de avaliar a viabilidade e a qualidade do funcionamento de um sistema utilizando essa topologia de rede, além de entrar em detalhes sobre o funcionamento e comportamento de redes *mesh* e sua implementação utilizada durante o trabalho. Entretanto, por conta deste objetivo central, foi desenvolvida uma aplicação mais simples, com apenas uma página web em cada nó para verificar o estado da vaga, além de utilizar um sensor que é indicado para detecção de movimentos ao invés de objetos.

3 PROPOSTA DO PROTÓTIPO

Nesta seção é apresentada proposta do protótipo, a motivação e justificativa para sua elaboração e uma comparação do que foi apresentado nos trabalhos correlatos. Também é apresentada a metodologia adotada para o desenvolvimento do protótipo.

3.1 JUSTIFICATIVA

No Quadro 5 é apresentado um comparativo das principais características de cada trabalho correlato. Nas linhas são apresentadas as características destacadas e nas colunas os trabalhos.

Quadro 5 - Comparativo dos trabalhos correlatos

Trabalhos Correlatos Características	Ronsani (2018)	Ferreira (2021)	Souza (2023)
Comunicação entre vários sensores fora de alcance direto a rede	Sim	Não	Sim
Método de conexão	Wi-Fi Mesh (<i>painlessMesh</i>)	Wi-Fi	Wi-Fi Mesh (<i>painlessMesh</i>)
Atualizações em tempo real	Sim	Sim	Não
Indicação visual com LEDs	Não	Sim	Não
Funcionalidades de gerenciamento	Não	Sim	Não
Microcontrolador	ESP32/ESP8266	ESP32	ESP8266
Aplicação centralizada para acompanhamento das vagas	Sim	Sim	Não
Plataforma do cliente	Web	Web	Web

Fonte: elaborado pelo autor (2024).

Conforme apresentado no Quadro 5, pode-se observar que apenas os trabalhos de Ronsani (2018) e Souza (2023) são eficientes no contexto de grandes estacionamentos com necessidade de comunicação de vários sensores fora do alcance direto a uma rede Wi-Fi, por utilizarem uma rede com topologia *mesh*. Também se observa que ambos utilizaram a biblioteca *painlessMesh* sob Wi-Fi e técnicas similares para sua implementação. Ferreira (2021), por sua vez, utiliza uma rede Wi-Fi comum em seu protótipo.

Em relação a plataforma do cliente e dispositivo embarcado, todos os trabalhos desenvolveram uma página web para a visualização e utilizaram os microcontroladores ESP32 ou ESP8266, já conhecidos e compatíveis com as ferramentas de desenvolvimento e bibliotecas mais utilizadas. Com a observação de que Souza (2023) desenvolveu uma página simples como prova de conceito, enquanto os outros projetos contaram com uma aplicação final e em tempo real, já voltada para o uso por clientes.

Ronsani (2018) tem como resultado o protótipo mais completo considerando o contexto geral, Souza (2023), em comparação, apesar de ter um protótipo com menos funcionalidades, tem o estudo e testes sobre redes

com topologia *mesh* mais completo. Enquanto isso, o trabalho de Ferreira (2023) tem o resultado mais completo ao se considerar apenas as funcionalidades do protótipo.

Diante dos resultados apresentados nestes trabalhos, pode-se afirmar que o uso de redes sem fio em malha são uma boa solução para o problema de comunicação de baixo custo entre muitos dispositivos em uma grande área onde uma conexão Wi-Fi ou com fio se torna muito cara ou inviável. Entretanto, as soluções de redes em malha apresentadas contam com algumas limitações, como quantidade de dispositivos, compatibilidade, confiabilidade da rede e perda de mensagens. Além disso, muitas das soluções comerciais para essa aplicação ainda são caras e exigem uma instalação complexa, muitas vezes utilizando redes Wi-Fi comuns ou com fio. Sendo assim, o presente trabalho compartilha mais com os objetivos de Ronsani (2018) e Souza (2023), tendo como objetivo o estudo, validação e a criação de um protótipo para apresentar uma solução mais eficiente e confiável para a comunicação sem fio entre dispositivos IoT utilizando o protocolo Thread.

Acredita-se que a utilização desse protótipo irá possibilitar uma solução de menor custo e fácil implementação para a aplicação de sensoriamento ocupação de vagas de estacionamento para auxiliar motoristas a encontrar vagas disponíveis, e com isso, reduzir o tempo de procura e proporcionar um maior conforto aos motoristas.

Para contribuir com o aprendizado de IoT e redes *mesh*, este protótipo pretende utilizar o protocolo Thread que é ainda relativamente novo e pouco explorado, principalmente fora do ambiente residencial. Como contribuição tecnológica de forma teórica e prática, se espera disponibilizar uma aplicação integrando este protocolo em um cenário não tradicional e pouco explorado, além de servir como uma documentação do processo de implementação e desenvolvimento, desde a criação da rede Thread até a integração com a aplicação IoT.

3.2 METODOLOGIA

O trabalho será desenvolvido de acordo com as seguintes etapas:

- a) revisão bibliográfica: realizar um levantamento sobre sensores de distância, ESP32-C6, redes *mesh* e o protocolo Thread;
- b) desenvolvimento inicial da aplicação do sensor: desenvolver a aplicação embarcada inicial para realizar a detecção de presença de um objeto e indicação do estado através de um LED colorido;
- c) testes de detecção: validar a precisão do sensor na detecção de veículos em vagas de estacionamento;
- d) implementação da rede: criar uma rede Thread com a implementação de código aberto OpenThread, utilizando o nRF52840 e o Raspberry Pi 3 B como roteador de borda;
- e) conexão do ESP32-C6 com a rede Thread: desenvolver a aplicação base para se conectar com a rede criada com a capacidade dos nós serem promovidos para roteadores;
- f) análise de performance e testes da rede: realizar testes sobre a performance e confiabilidade da rede, tempo de resposta, alcance e reconstrução da rede em caso de perda de um nó;
- g) desenvolvimento da aplicação *back-end*: criar um serviço para agrupar os dados dos sensores e distribuir para os clientes consultarem o estado das vagas em tempo real;
- h) desenvolvimento da aplicação *front-end*: criar uma aplicação para smartphones integrada com a aplicação *back-end* para fazer a consulta da ocupação das vagas em tempo real;
- i) integração da aplicação dos sensores com a rede Thread e *back-end*: integrar o envio da informação de ocupação da vaga com o serviço *back-end* através da rede Thread;
- j) desenvolvimento do protótipo final: elaboração de uma caixa para conter o sensor, o ESP32-C6 e o LED indicador;
- k) validação do protótipo final: efetuar testes de funcionalidade com o protótipo final em um estacionamento real.

4 REVISÃO BIBLIOGRÁFICA

Esta seção descreve brevemente os principais conceitos que fundamentarão o protótipo a ser desenvolvido neste trabalho: a internet das coisas (subseção 4.1), o sensor HC-SR04 (subseção 4.2), ESP32-C6 (subseção 4.3) e a rede Thread (subseção 4.4).

4.1 INTERNET DAS COISAS

Segundo Ronsani (2018), o termo Internet das Coisas (Internet of Things – IoT), surgiu inicialmente em 1999 por Kevin Ashton. O termo descreve um sistema em que objetos do mundo físico estejam estar conectados à internet por meio de sensores, estendendo à objetos do dia a dia, antes não considerados computadores, a conectividade à rede e a capacidade computacional. Isto permite que tais dispositivos possam gerar, trocar e consumir dados sem intervenção humana. Esse tipo de conexão também é chamado de máquina-à-máquina (machine-to-machine – M2M).

Na época, o conceito era principalmente utilizando em projetos de pesquisa ou militares, entretanto, junto com os avanços na tecnologia atual e o crescimento da internet, a IoT se expandiu para diversas áreas em nosso cotidiano. Segundo Sinha (2024), no final de 2023 havia cerca de 16.6 bilhões de dispositivos IoT conectados à internet, com uma estimativa de atingir 40 bilhões de dispositivos até 2030.

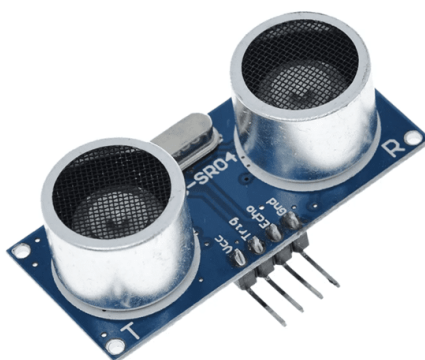
Existem seis principais elementos para o funcionamento de aplicações IoT. De acordo com Ronsani (2018), eles são: identificação, sensoriamento ou atuadores, comunicação, processamento, serviços e semântica. Um avanço importante vem ocorrendo nos últimos anos, principalmente nas áreas de comunicação e processamento, que tornaram possível a interconexão eficiente entre dispositivos heterogêneos, com baixo custo e baixo consumo de energia, ao mesmo tempo que possuem o poder de processamento para a coleta de dados de sensores e execução de ações por meio de atuadores.

4.2 SENSOR ULTRASSÔNICO HC-SR04

O sensor ultrassônico HC-SR04 (Figura 1) é um sensor de baixo custo e comumente utilizado em dispositivos embarcados para medição de distância de um objeto e com isso, podendo também ser utilizado para detecção de presença. Ele permite que sejam feitas leituras de distâncias entre 2cm e 4m com uma precisão de 3mm e um ângulo de efeito de 15°. Com uma interface simples de quatro pinos (VCC, *Trigger*, *Echo* e GND), pode ser utilizado com diversos microcontroladores.

O sensor funciona enviando ondas ultrassônicas e medindo o tempo que as ondas levam para refletirem de volta ao sensor após atingir um objeto. Durante a emissão e recebimento das ondas, o pino *Echo* ficará em nível alto. Com isso, a distância entre o sensor e o objeto pode ser calculada multiplicando o tempo que *Echo* está em nível alto pela velocidade do som e dividindo o resultado por 2 (emissão e reflexão).

Figura 1 – Sensor ultrassônico HC-SR04



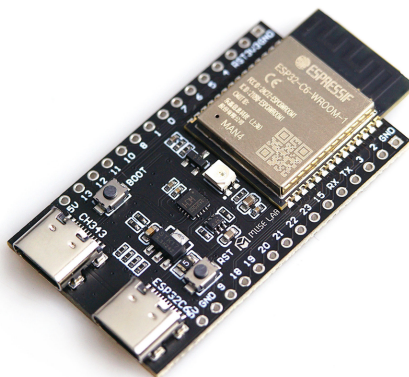
Fonte: MakerHero (2021).

4.3 MICROCONTROLADOR ESP32-C6

O ESP32 é um microcontrolador com um sistema em um chip (System-on-Chip – SoC) desenvolvido pela Espressif e muito utilizado em projetos de IoT, pelo seu baixo custo, baixo consumo de energia e boas opções de conectividade. Os SoCs da família ESP32 tem suporte ao ESP-IDF, o framework de desenvolvimento oficial da Espressif, baseado no kernel FreeRTOS e permite o desenvolvimento nas linguagens C e C++. Além disso, os SoCs também têm suporte ao ambiente Arduino, que facilita o desenvolvimento e permite o uso de bibliotecas já conhecidas e compatíveis com outros microcontroladores.

O modelo escolhido para o protótipo deste trabalho é o ESP32-C6, com a respectiva placa de desenvolvimento exibida na Figura 2. Este modelo possui um processador RISC-V com clock máximo de 160 MHz e um segundo processador de baixa potência de pode chegar a até 20 Mhz, 512KB de SRAM, 320KB de ROM e 16MB de memória SPI Flash. O grande diferencial do ESP32-C6 para o ESP-32 é que além do suporte a Wi-Fi 6 e Bluetooth 5, conta com suporte ao padrão de rádio IEEE 802.15.4, essencial para a compatibilidade com os protocolos de rede Thread e Zigbee, por exemplo, sem necessidade de algum equipamento extra (Espressif Systems, 2024).

Figura 2 – Placa de desenvolvimento nanoESP32-C6 v1.0



Fonte: MuseLab (2023).

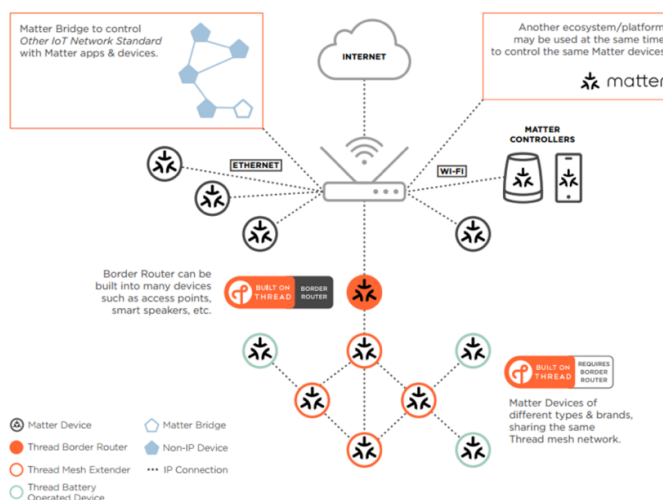
4.4 REDE THREAD

Thread é um protocolo de rede *mesh* de baixo consumo de energia para comunicação de dispositivos de internet das coisas, desenvolvido pela Thread Group Inc., publicado em 2015 e baseado no protocolo IPv6. As redes *mesh* são compostas por roteadores e clientes, onde cada nó pode operar tanto como cliente quanto como roteador, encaminhando pacotes para nós que não possuem caminho direto ao destinatário (Ronsani, 2018, p. 42).

O protocolo Thread se diferencia de outras redes similares como a Zigbee, Z-Wave e Bluetooth Mesh por implementar uma comunicação IPv6 entre os nós, além de ser uma tecnologia com a intenção de ser um padrão aberto, com uma implementação open-source (OpenThread) mantida pela Google. Além disso, o protocolo tem como alguns de seus pilares principais, a simplicidade, segurança, confiabilidade (auto-reparável e sem ponto único de falha), eficiência e escalabilidade (Silicon Labs, 2024).

Um dos diferenciais do protocolo Thread é a utilização do IPv6, que além de facilitar o desenvolvimento das aplicações, por já utilizar um protocolo de rede padrão, permite a comunicação direta dos dispositivos da rede Thread com a Internet utilizando o endereço IPv6, com o único requisito sendo existir um roteador de borda de estabelecer a conexão entre as duas redes. A Figura 3 mostra os componentes que podem compor uma rede Thread. Os nós finais são compostos por dispositivos operados com baterias, que não tem capacidade de tomar o papel de roteador, dispositivos conectados a rede elétrica, que podem ser promovidos para roteadores conforme a necessidade da rede e um roteador de borda, que serve como uma ponte entre a rede Thread e uma rede Wi-Fi, por exemplo.

Figura 3 – Diagrama de componentes de uma rede Thread

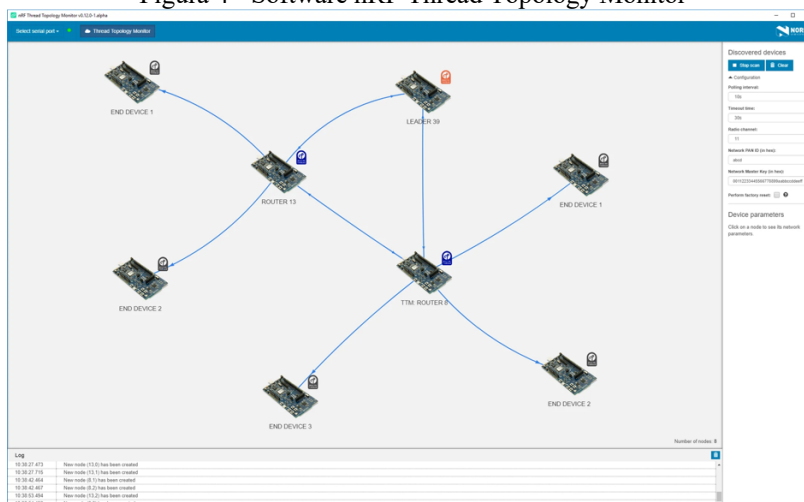


Fonte: Thread Group (2024).

O protocolo Thread é auto-reparável, o que significa que se um nó que atuava como roteador for desconectado, a rede será reorganizada e um novo nó será promovido como roteador por meio de uma votação dos outros nós da rede, permitindo escolher o melhor nó disponível. Isso é especialmente importante em aplicações onde é necessário uma alta confiabilidade da rede.

Atualmente, a OpenThread tem suporte oficial completo para o ESP32-C6, porém apenas no framework ESP-IDF. Além disso, o ESP32-C6 não tem a capacidade de atuar como um roteador de borda, por conta de o rádio ser compartilhado com o Wi-Fi. Por conta disso, neste trabalho será utilizado um Raspberry Pi 3 B junto com um dongle nRF52840, desenvolvido pela Nordic Semiconductor, para atuar como o roteador de borda da rede Thread. A Nordic também disponibiliza um software para a visualização da topologia (Figura 4) que é compatível com o dongle nRF52840 e que também será utilizado para verificar e validar a topologia da rede implementada.

Figura 4 - Software nRF Thread Topology Monitor



Fonte: Nordic Semiconductor (2021).

REFERÊNCIAS

- AFIFE, Nur A. et al. Evaluation method of mesh protocol over ESP32 and ESP8266. **Baghdad Science Journal**, v. 18, n. 4 (Suppl.), p. 1397-1405, dec. 2021.
- CORREA, Naiara. **Comparativo de protocolos de IoT para automação residencial**: potenciais vulnerabilidades e sugestões de melhorias. 2018. 19 f. Artigo de conclusão de curso (Bacharel em Ciência da Computação) - Instituto de Ciências Exatas e Geociências, Universidade de Passo Fundo, Passo Fundo.
- DETRAN-SC. **Histórico de Frota**. [S.l.], [2024]. Disponível em: <https://www.detran.sc.gov.br/estatisticas-historico-de-frota-transparencia>. Acesso em: 17 set. 2024.
- ESPRESSIF SYSTEMS. **ESP32-C6 Series Datasheet Version 1.2**. [S.l.], 2024. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf. Acesso em: 14 set. 2024.
- FERREIRA, Fernando. **Sistema inteligente de gerenciamento e controle de vagas de estacionamento**. 2021. 62 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Controle e Automação) - Curso de Engenharia de Controle e Automação, Universidade Federal de Santa Maria, Santa Maria.
- GOLDNER, Rafael K. **Protótipo de sistema para monitoramento de vagas de estacionamento utilizando Internet das Coisas e comunicação por LoRa**. 2022. 75 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia Elétrica) - Curso de Graduação em Engenharia Elétrica, Instituto Federal do Espírito Santo, Vitória.
- MAKERHERO. **Sensor de Distância Ultrassônico HC-SR04**. [S.l.], [2021?]. Disponível em: <https://www.makerhero.com/produto/sensor-de-distancia-ultrassonico-hc-sr04>. Acesso em: 20 set. 2024.
- MALIK, Praveen K. et al. Implementation of a smart parking system using LoRa, STM32 Board & IoT. **2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE)**, Ghaziabad, p. 135-139, nov. 2023.
- MUSELAB. **nanoESP32-C6**. [S.l.], [2023]. Disponível em: <https://github.com/wuxx/nanoESP32-C6>. Acesso em: 20 set. 2024.
- NORDIC SEMICONDUCTOR. **nRF Thread Topology Monitor**. [S.l.], [2021?]. Disponível em: <https://www.nordicsemi.com/Products/Development-tools/nRF-Thread-topology-monitor>. Acesso em: 25 set. 2024.
- ORRIE, O.; SILVA, B.; HANCKE, G. P. A Wireless Smart Parking System. **IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society**, p. 4110-4114, nov. 2015.

RONSANI, Luiz A. K. **UM PROTÓTIPO DE SISTEMA INTELIGENTE PARA MONITORAMENTO DE VAGAS DE ESTACIONAMENTOS: UMA APLICAÇÃO BASEADA EM INTERNET DAS COISAS**. 2018. 89 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Computação) – Centro de Ciências, Tecnologia e Saúde, Universidade Federal de Santa Catarina, Araranguá.

SEYMER, Paul; WIJESEKERA, Duminda; KAN, Cing-Dao. Secure Outdoor Smart Parking using Dual Mode Bluetooth Mesh Networks. **2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)**, p. 1-7, apr./may. 2019.

SILICON LABS. **UG103.11: Thread Fundamentals**. [S.l], [2024]. Disponível em: <https://www.silabs.com/documents/public/user-guides/ug103-11-fundamentals-thread.pdf>. Acesso em: 14 set. 2024.

SILVA, David J. B. **Sistema Multiagente baseado numa Rede Mesh**. 2022. 116 f. Tese de Doutorado (Mestrado em Engenharia Electrotécnica e de Computadores) – Departamento de Engenharia Eletrotécnica, Instituto Superior de Engenharia do Porto, Porto.

SIMATUPANG, Joni W.; LUBIS, Aida M.; Vincent. IoT-Based Smart Parking Management System Using ESP32 Microcontroller. **2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)**, p. 305-310, oct. 2022.

SINHA, Satyajit. **State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally**. [S.l], 2024. Disponível em: <https://iot-analytics.com/number-connected-iot-devices/>. Acesso em: 29 set. 2024.

SOUZA, David M. **Rede em malha sem fio utilizando o dispositivo ESP-8266**: aplicação e avaliação de desempenho em cenário de detecção de presença de veículos em estacionamentos. 2023. 71 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia Elétrica - Eletrônica) - Curso de Graduação em Engenharia Elétrica - Eletrônica, Universidade Federal do Amazonas, Manaus.

THREAD GROUP. **SMART HOME NETWORK TOPOLOGY**. [S.l], [2024?]. Disponível em: <https://www.threadgroup.org/BUILT-FOR-IOT/Smart-Home>. Acesso em: 25 set. 2024.