REVISTA CIENTÍFICA SENAI-SP

APLICAÇÃO DE REDES ESP-WIFI-MESH EM ZONA RURAL

APPLICATION OF ESP-WIFI-MESH NETWORKS IN RURAL AREA

Gleydson Henrique de Alencar Santos[1], [i]
Murilo José de Carvalho[2], [ii]
Fernando Simplicio de Sousa[3], [iii]
Leandro Poloni Dantas[4], [iv]

Data de submissão: (20/04/2022) Data de aprovação: (22/07/2022)

RESUMO

A utilização de redes de comunicação sem fio com topologia malha (*mesh*) tem crescido consideravelmente, devido à expansão do conceito de internet das coisas (*internet of things - IoT*) e à baixa capacidade de conexão dos roteadores domésticos. Existe uma vasta gama de publicações de pesquisas acadêmicas envolvendo esse tipo de tecnologia de redes de comunicação. Contudo, conforme observado nas revisões bibliográficas deste trabalho, grande parte das referidas publicações tem como principal abordagem a utilização de simulações e/ou modelos matemáticos, deixando os testes práticos em segundo plano e faltando destaque para a área de sistemas embarcados. Diante do exposto, este trabalho teve como objetivo a realização de uma pesquisa de campo da aplicação da rede ESP-WIFI-MESH em área parcialmente aberta de zona rural, onde foi analisada a viabilidade da implementação de uma rede *mesh*, utilizando o *System on Chip* (*SoC)* ESP-WROOM-32 da empresa *Espressif*, através de testes de taxa de transferência, perda de pacotes e latência, automatizados em *scripts Python*, com coleta de dados via MQTT, verificando que a rede é capaz de trafegar dados até 20 kbps, com latência média abaixo de 250 ms e praticamente sem perda de pacotes, provando-se, assim, que a aplicação da tecnologia ESP-WIFI-MESH pode ser uma ótima escolha para áreas rurais.

ABSTRACT

The use of wireless communication networks with mesh topology has increased considerably, due to the expansion of the concept of internet of things (IoT) and the low connection capability of home routers. There is a wide range of academic research papers concerning this type of communication network technology. However, as observed in the bibliographic reviews of this work, most of these publications have as their main approach the use of simulations and/or mathematical models, leaving practical tests in the background and lacking emphasis on the area of embedded systems. Given the above, this work aimed to carry out a field research on the application of the ESP-WIFI-MESH network in a partially open rural area,

[1] Engenheiro Elétrico. E-mail: ghasantos@gmail.com
[2] Engenheiro Eletrônico. E-mail: murilojcarvalho@gmail.com
[3] Mestre em Engenharia Elétrica. E-mail: fernando.simplicio@sp.senai.br
[4] Doutor em Engenharia Elétrica. E-mail: leandro.poloni@sp.senai.br

which was analyzed the feasibility of implementing a mesh network using the System on Chip (SoC) ESP-WROOM-32 from the company Espressif, through tests of transfer rate, packet loss and latency, automated in Python scripts, with data collection via MQTT, verifying that the network is capable of transferring data up to 20 kbps, with average latency less than 250 ms and practically no packet loss, thus proving that the application of ESP-WIFI-MESH technology can be a great choice for rural areas.

1 INTRODUÇÃO

As redes *wireless* são um dos meios de comunicação mais comuns utilizados atualmente. Segundo Júnior (2012), as redes sem fio surgiram com o intuito de complementar as redes cabeadas, utilizando a propagação de sinais através de ondas eletromagnéticas, e permitindo a ampliação e flexibilização da localização das estações de acesso. Características importantes nesses tipos de redes são largura de banda e alcance de transmissão. As redes de curta distância mais conhecidas são WPAN (*Wireless Personal Area Network*) e WLAN (*Wireless local Area Network*).

As redes WPAN, cujas propriedades são definidas pelo padrão IEEE 802.15, em geral, conseguem transmitir para até 10 metros e atingir velocidades superiores a 2 Gb/s. São comumente utilizadas para conexão de computadores e dispositivos localizados na mesma sala ou escritório. Um exemplo de aplicação dessa rede é o *Bluetooth* (FISHER, 2007).

As redes WLAN, padronizadas pelo IEEE 802.11, o qual inclui a tecnologia Wi-Fi, amplamente difundida nas redes de comunicação *wireless*, sendo denominada como 802.11n a versão do protocolo que predomina em aplicações domésticas pode atingir até 600 Mb/s, 70 metros em ambientes internos e 250 metros em ambientes externos (BLACK BOX, 2018).

Muitas aplicações de equipamentos eletrônicos podem exigir a comunicação através de redes *wireless*, como é o caso do conceito de IoT, onde existe a necessidade de conectar os dispositivos entre si, e/ou com serviços externos, como nuvens, sites, banco de dados, entre outros (BURGESS, 2018).

Conforme observado por Espressif Systems Inc (2016), com o crescimento de aplicações de IoT, os roteadores *wireless* se tornam o “gargalo” de conexões em uma mesma rede local, ficando, geralmente, abaixo de 32 dispositivos para os roteadores domésticos mais comuns. Nessa situação, duas soluções são mais plausíveis: aquisição de um roteador com maior capacidade de gerenciamento de dispositivos ou a utilização de redes *mesh*.

As redes *mesh* ou rede malha são formadas por módulos de rádio, que estão interconectados, dispensando dispositivos extras, como roteadores, para gerenciar as conexões. Geralmente esse tipo de rede pode ser auto organizável e autoconfigurável, permitindo uma topologia dinâmica dos dispositivos na malha (PRASINA e THANGARAJA, 2011).

O ESP-WIFI-MESH, desenvolvido pela empresa Espressif, é um protocolo de rede *mesh*, construído sobre os padrões do Wi-Fi, que permite de maneira autônoma a conexão de múltiplos dispositivos, cobrindo amplas áreas e utilizando apenas uma interface de WLAN (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2021).

Durante as revisões bibliográficas para este artigo, principalmente nas publicações do IEEE, foi verificado que existe ampla gama de estudos sobre os limites e viabilidade de implementação de redes Wi-Fi *mesh*, contudo, a grande maioria desses trabalhos utilizam somente simulações, modelos matemáticos e/ou outras revisões bibliográficas, além de

muitos não abordarem a utilização de dispositivos embarcados como solução para IoT.

Assim, diante do exposto, o objetivo deste trabalho será direcionado para a pesquisa de campo, onde será analisada a viabilidade de implementação de uma rede ESP-WIFI-MESH em uma área rural parcialmente aberta, verificando as seguintes características: taxa de transmissão, latência, perda de pacotes, tempo de construção e recuperação da rede, tabela de roteamento e *received signal strength indication* (RSSI).

2 DESENVOLVIMENTO

Durante a concepção deste trabalho foram realizadas diversas pesquisas relacionadas ao tema escolhido. A seguir, foram analisados os trabalhos que mais se aproximaram de nossa proposta.

Naidoo e Sewsunker (2007), em sua pesquisa, estudam a implantação de uma rede Wi-Fi e WiMax *mesh* em uma cidade no sul da África, a fim de obter o máximo alcance de cobertura. Para isso, foi construída uma rede malha-estrela com e sem antenas setorizadas, além da utilização do *software* OPNET MANET para a simulação, verificando em qual distância a rede funcionaria com uma taxa de transferência adequada. Eles obtiveram o resultado mais satisfatório nas simulações de rede com topologia malha-estrela e antenas setorizadas, onde para uma capacidade de 100 kbps, a distância do primeiro salto do ponto de acesso foi de 1,302 km.

Kum et al. (2010) propõem a implementação de uma rede Wi-Fi *mesh* híbrida para melhorar o seu desempenho. Para isso, foi efetuado um teste prático sob o protocolo 802.1, com 50 nós formando em um plano retangular de 1,5 km x 0,3 km e com velocidade de movimentação de até 1 m/s. Os testes foram feitos utilizando três protocolos: OLSR (*Optimized Link State Routing*), AODV (*Ad hoc On-demand Distance Vector*) e o híbrido AODV-MHR. (*Mobility-aware Hybrid Routing*). Um nó se movia aleatoriamente no espaço e realizava pausas com tempos pré-definidos, que variavam de 0 a 600 segundos. Em todos esses intervalos entre movimentos e pausas foram realizadas as medidas de *overhead*, taxa de transferência e latência na comunicação. Os resultados de sobrecarga da rede mostram que o OLSR (*Optimized Link State Routing*) obteve um desempenho inferior aos outros protocolos, uma vez que englobava as informações de roteamento. No quesito taxa de transferência, o protocolo OLSR supera os outros protocolos quando não há movimento do nó, mas para maiores mobilidades foi superado pelos protocolos AODV-MHR (*Mobility-aware Hybrid Routing*) e AODV (*Ad hoc On-demand Distance Vector*), com uma ressalva de que o AODV-MHR praticamente empatou com OLSR quando não há mobilidade. OLSR obteve um desempenho superior aos outros dois protocolos no teste de atraso na entrega dos pacotes, com o nó em pausa e em movimento constante.

Oki et al. (2014) estudam um algoritmo para seleção de um distribuidor de chave da rede *mesh* por distância, quando o dispositivo atual falhar por algum motivo. O objetivo é selecionar o líder da rede baseado na distância em que ele se encontra da estação de base. Para a simulação, foi utilizado o *software* Network Simulator v2.35 no sistema operacional Ubuntu. Na simulação os números de nós foram variados entre 50 e 500, em uma área de 1000 m x 1000 m, com um tempo de 1000 segundos de simulação. Foram testados o *overhead* na comunicação, o atraso na seleção do líder e a taxa de consumo de energia. Nos testes com algoritmos de seleção de líder, foi observado que as estratégias baseadas em eventos obtiveram melhores resultados do que os baseados em distância, pois causam menos *overhead* na comunicação.

No quesito atraso na seleção do líder e consumo de energia, os algoritmos baseados em eventos também tiveram um resultado melhor.

Pfeifer (2018), em sua proposta, desenvolveu uma rede sob o protocolo ESP-MESH, com o intuito de ligar medidores de energia em rede. Para os testes do projeto foram estudadas as variáveis: número de nós, distância e intolerância a falhas. Nos testes práticos, para verificação do desempenho, foi implementada a rede *mesh* em diversos cenários, como os nós ao redor do gateway, um nó próximo ao gateway e os outros nós montados em linha reta e no sentido oposto do anterior. Além disso, foram criadas falhas propositais, por exemplo, afastamento e desligamento de um nó. Nos testes de desempenho da rede foram recebidas mensagens do gateway, como estado da comunicação, potência do sinal, atualização do relógio e árvore de conexões. Entretanto, o principal resultado que o autor almejava eram as medidas dos sensores de corrente, objetivo que foi alcançado.

Maksutov et al. (2019) utilizam o algoritmo de roteamento AODV, juntamente com o *hardware* ESP-8266, para formar uma rede *mesh* e demonstrar a capacidade de roteamento entre 4 dispositivos. Apesar de os autores demonstrarem resultados através de figuras, sem uma análise dos dados, conseguiram um desempenho satisfatório, pois o objetivo era provar a possibilidade de roteamento e troca de informações entre os nós.

Gergeleit (2019) desenvolveu um algoritmo denominado de "*Autotree*" em conjunto com o *hardware* ESP-8266, para configurar automaticamente as interfaces e configurações de rede. Para isso, os dispositivos são configurados para atuar como *Station* (STA) e *Access Point* (AP). A rede é criada através do nível de sinal SSID de cada nó em relação à rede Wi-Fi original, nomeando o *root* da rede e as camadas subsequentes com os seus nós. O cenário de avaliação foi composto por três ESP-8266 separados por 7 metros em linha reta, uma Raspberry Pi para envio de pacotes e um *smartphone* para recebimento das mensagens da rede. O algoritmo foi configurado para cada nó formar uma camada de rede, ou seja, três camadas. Como resultado, cada nível de malha dividiu pela metade a largura de banda e a latência aumentou cerca de 2ms por camada na ida e na volta.

Na maioria dos trabalhos pesquisados, os autores optaram em utilizar simulações para verificar o desempenho da rede *mesh*, poucos trabalhos tiveram uma construção de uma infraestrutura real para obter seus resultados e conclusões, portanto, foi decidido citar os quatro únicos trabalhos práticos pesquisados.

2.1 O Padrão IEEE 802.11

O protocolo de rede sem fio Wi-Fi (*Wireless Fidelity*) foi primeiramente padronizado em 1997 pelo IEEE com o nome 802.11, onde a capacidade de transmissão era de 1 a 2 Mbps, com uma banda de frequência de 2,4 GHz. Esse protocolo dois anos depois teria variações como o 802.11a e 802.11b. O protocolo 802.11a se consolidou melhor no mercado, pois ofereceu mais variedade na taxa de transmissão de dados, iniciando em 6 Mbps e atingindo até 54 Mbps na frequência de 5 GHz, podendo operar com 12 canais diferentes. A única melhoria implementada no 802.11b era uma técnica de transmissão de dados que possibilitava até 11Mbps. Em 2003 surgiu a terceira geração de Wi-Fi, denominada 802.11g, que combinou os padrões 802.11a e b, funcionando somente na frequência de 2,4 GHz. No ano de 2009, foi lançado o 802.11n, oferecendo melhorias em relação às versões anteriores, com velocidades podendo chegar até 300 Mbps, ou 600 Mbps com o uso de *hardwares* e antenas mais avançadas (BRIERE e HURLEY, 2011).

Em 2013, o 802.11ac foi padronizado pelo IEEE, operando na faixa de 5 GHz, com uma largura de banda de 160 MHz e podendo alcançar a taxa de transferência de até 1 Gbps.

A WLAN pode ser classificada como uma rede que não necessita de fios interligando um dispositivo ao outro, e que funciona dentro de uma área geográfica pequena, como escola, *shopping*, edifício, entre outros. Isso permite a comunicação entre múltiplos dispositivos, através de sinais de rádio sob um protocolo definido (CHEN, 2020).

2.2 Redes *wireless* com topologia *mesh*

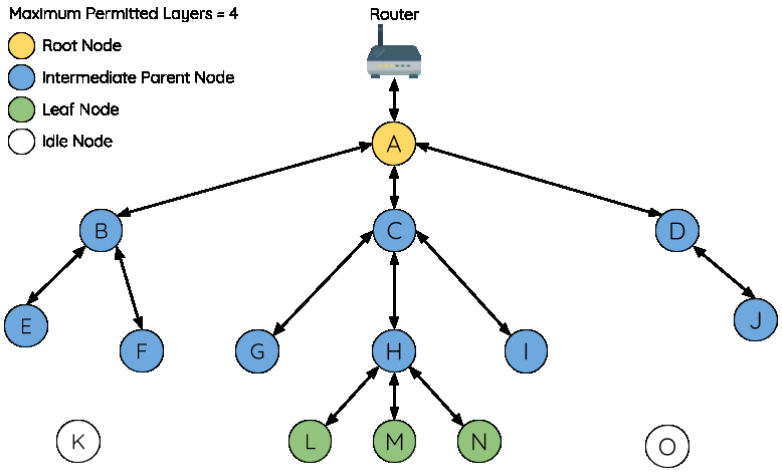
Wireless Mesh Networks (WMN) é um protocolo definido pelo padrão IEEE 802.11s, operando na topologia de "árvore", com dispositivos funcionando simultaneamente como *Station* e *Acesss Point*, rotas de tráfego de dados dinamicamente definidas pela rede e endereçamento dos dispositivos utilizando o endereço MAC (*Media Access Control)* (POKHREL e SHAKYA, 2013).

Hu e Kuo (2008) afirmam: "IEEE 802.11s é baseado em padrões IEEE 802.11 anteriores e ambas as camadas MAC são semelhantes, exceto pela adição de algumas alterações em relação ao conteúdo de *multi-hop*."

2.3 ESP-WIFI-MESH

ESP-WIFI-MESH é uma implementação de rede sem fio de topologia *mesh*, construída sobre o protocolo Wi-Fi tradicional, com intuito de conectar múltiplos dispositivos em vastas áreas territoriais, utilizando apenas uma única interface de WLAN. Foi desenvolvido pela empresa Espressif, visando a utilização nos microcontroladores da empresa, como o ESP32 e o ESP8266 (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2021). Em uma rede ESP-WIFI-MESH, exemplificada na Figura 1, os dispositivos conectados, doravante chamados de nós, ou do inglês *nodes*, podem ser classificados como sendo do tipo (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2021):

Figura 1 - Tipos de *nodes* da rede ESP-WIFI-MESH.

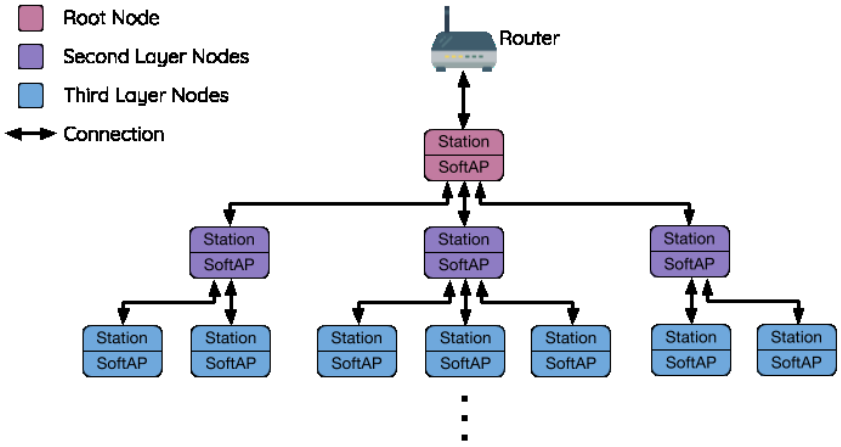


Fonte: ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD (2021).

- *Root Node*: É o nó que está no topo da rede, tendo como uma de suas funções realizar a ponte de comunicação dos restantes dos nós com a WLAN.
- *Intermediate Parent*: Nodes: São nós que possuem sub-redes.
- *Leaf Nodes*: São os dispositivos finais ou nas pontas da rede.
- *Idle Nodes*: São dispositivos que não foram conectados, seja por estarem fora do alcance do sinal, ou em razão de a rede ter atingido o número máximo de *layers* previamente configurado.

No ESP-WIFI-MESH, somente um *node* é designado como *root*, através de um algoritmo de votação implementado no protocolo, onde todos os dispositivos votam com base no RSSI em relação ao roteador. De maneira simplificada, o root é o interlocutor da ESP-WIFI-MESH com a WLAN (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2021). Todos os *nodes* da rede podem ser simultaneamente *Station* e *Acess Point*. Todos os *child nodes* estão conectados a algum *parent node*, que por sua vez também estão conectados a outros *parent nodes*, e assim sucessivamente, até que o penúltimo *layer* da rede se conecte ao *root*. Dessa forma, a rede inerentemente sempre será construída na topologia de "árvore", conforme Figura 2 (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2021).

Figura 2 - Topologia da rede ESP-WIFI-MESH.



Fonte: ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD (2021).

Nesse tipo de rede é imprescindível que todos os dispositivos possuam tabelas de roteamento, indicando quais nodes estão em suas sub-redes, assim, permitindo o adequado direcionamento dos dados trafegados. Todos os nodes *nonroot* são unicamente identificados pelo seu *Media Access Control* (MAC) *Address*. O *root* é o único dispositivo que possui um endereço de IP provido pela WLAN (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2021).

2.4 Internet das Coisas

O conceito de internet das coisas, pode ser descrito como uma rede de dispositivos, englobando desde eletrodomésticos até equipamentos industriais, com o objetivo de prover conexão e comunicação entre si e/ou com a internet. Esse conceito tem crescido notavelmente com a expansão da computação de baixo custo, serviços de nuvem, big data, tecnologias móveis, entre outros, permitindo que o mundo digital seja cada vez mais integrado ao mundo físico.

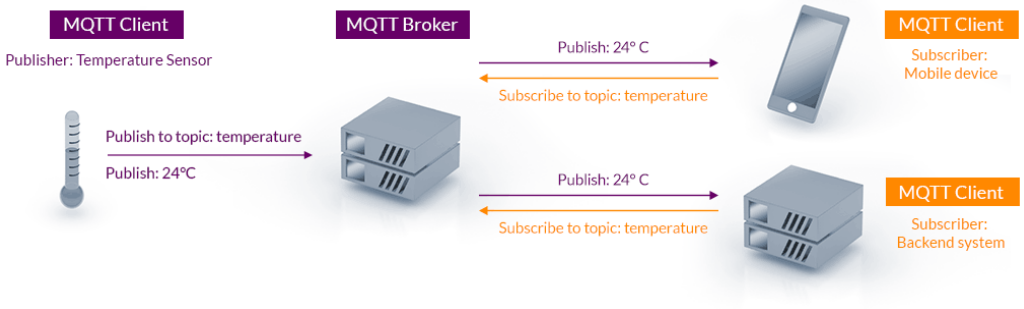
Dispositivos IoT podem ser aplicados em diversos setores da sociedade, tais como manufatura, automotivo, transporte e logística, setor público, médico, segurança do trabalho (ORACLE, 2021).

2.5 Protocolo MQTT

Segundo a descrição pelo HIVEMQ (2021), MQTT é um padrão OASIS para conectividade IoT. É um protocolo de mensagens no modelo de publicação/assinatura, extremamente simples e leve, projetado para dispositivos limitados e redes de baixa largura de banda, alta latência ou não confiáveis. Os princípios de design visam minimizar a largura de banda da rede e os requisitos de recursos do dispositivo, ao mesmo tempo que tentam garantir a confiabilidade e algum grau de garantia de entrega. Esses princípios também tornam o protocolo ideal para o mundo de dispositivos conectados IoT, e para aplicativos móveis em que a largura de banda e a energia da bateria são preciosas.

O protocolo MQTT utiliza o modelo *publishing/subscribe* em tópicos para troca de mensagens. Todas as mensagens distribuídas na rede devem obrigatoriamente ser gerenciadas por um *Broker*. Os clientes da rede podem se inscrever e publicar as mensagens nos tópicos que forem de interesse do dispositivo. Múltiplos clientes podem se inscrever e publicar em um mesmo tópico. A Figura 3 exemplifica o modelo de uma rede utilizando o MQTT.

Figura 3 - Arquitetura *Publish/Subscribe* MQTT.



Fonte: MQTT.ORG (2020).

O Eclipse Mosquitto é um *Broker* MQTT *open source*, disponibilizado pela *Eclipse Foundation* e patrocinado pela *cedalo.com*, leve e capaz de ser usado nas mais diversas plataformas, inclusive em *Single Board Computers* como a *Raspberry Pi* (ECLIPSE MOSQUITTO, 2021).

2.6 *System on Chip* (SoC) - ESP32

O ESP32 é *System on Chip* (SoC) que contempla um microcontrolador (µC) com Wi-Fi e Bluetooth integrados. Possui design robusto, podendo operar inclusive em ambientes industriais, suportando temperaturas de -40 °C a +125 °C. Inclui modos de operação de baixo consumo de energia, alto nível de integração com antenas, amplificadores e filtros de sinal, e demais interfaces comuns a microcontroladores, como IOs, UART, SPI, I2C (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2021).

2.7 Taxa de transferência, Latência e Perda de pacotes

A taxa de transferência, ou do inglês *throughput*, pode ser descrita como a quantidade de dados movida entre dispositivos em um determinado período. Para transferência em uma rede wireless, normalmente a unidade de medida é expressa em bits por segundo (bps), ou para grandes transferências de dados em kbps, Mbps e Gbps (INTEL, 2018).

A Latência (*lag*) é a quantidade de tempo, ou atraso, que um dado utiliza para trafegar de um ponto a outro em uma rede. Geralmente seu valor é apresentado em milissegundos (ms) (HOSTINGER, 2019).

A perda de pacotes, como o próprio nome já sugere, descreve o quantitativo de dados que são enviados e não chegam ao destinatário em uma rede. As principais causas de perdas de pacotes geralmente envolvem: congestionamentos na rede, *bugs* nos softwares, falhas de hardware, ameaças de cyber-segurança, configurações incorretas (PROOFPOINT, 2021).

2.8 Tempo de construção de rede (*building time*) e recuperação da rede (*healing time*)

É o tempo despendido para que a rede *mesh* seja completamente construída, englobando inicialização dos *nodes*, seleção do *root*, definição das sub-redes, e conexão à rede WLAN (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2021). *healing time* é o tempo que a rede *mesh* precisa para detectar a falha em algum *node* e definir novas conexões entre os dispositivos, inclusive nomeando um novo root, caso necessário, para que seja restabelecido o pleno funcionamento das comunicações (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2021).

3 METODOLOGIA

Os testes de campo foram realizados na área rural do munícipio de Pedra Bela, estado de São Paulo, altitude média de 1.000 m em relação ao nível do mar, segundo medições do Google Earth para coordenada 22°47’53” S e 46°30’45” W.

Todos os ESP32 foram distribuídos de forma que sempre houvesse um node vizinho sem obstruções, 2 m de altura mínima do solo e distâncias variando de 5 m a 75 m entre os dispositivos da rede. Foram distribuídos 8 *nodes* na área de teste. Vide Figura 4.

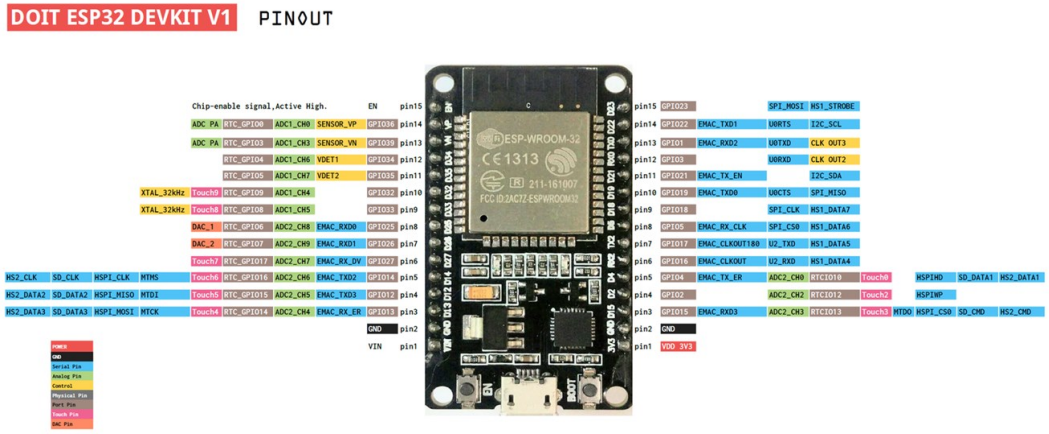
Figura 4 – Distribuição e distância máxima e mínima entre os *nodes*.

Fonte: elaborado pelo autor (2021).

Optou-se pela realização dos testes práticos em uma área parcialmente aberta, visando diminuir a subjetividade dos resultados, pois obstáculos como paredes, árvores, veículos etc., poderiam causar variações nos dados obtidos.

O *hardware* escolhido para embarcar a rede Wi-Fi *mesh* foi o ESP32, contudo, a Espressif possui grande variação de modelos deste SoC. O modelo de ESP32 utilizado foi o ESP32-WROOM-32, presente no kit de desenvolvimento DOIT ESP32 DEVKIT V1, conforme Figura 5. Esse kit de desenvolvimento facilita a utilização do ESP32, pois já possui conversor serial/USB, regulador de tensão e terminais compatíveis com protoboard.

Figura 5 – DOIT ESP32 DEVKIT V1.



Fonte: Fernando K (2021).

O *firmware* base do ESP32 utilizado nos testes possui o protocolo ESP-WIFI-MESH completamente implementado, para que os *nodes* pudessem assumir qualquer função na rede (*root*, *parente*, *child*), além de envio de mensagens entre todos os dispositivos. Também estava embutido o MQTT no código fonte, permitindo solicitar a tabela de roteamento do *root*, enviar comandos ao LED *builtin* das placas ESP32 DEVKIT V1 e coletar os dados dos testes realizados.

A rede ESP-WIFI-MESH foi configurada com *self-building*, *self-healing*, definição automática do *root* pela votação dos *nodes* e criação da topologia pelos próprios dispositivos. Essa foi a abordagem adotada, pois o usuário final provavelmente implementará a rede da maneira mais autônoma possível, visto que uma pessoa mais leiga não teria conhecimento e/ou interesse em definir topologias de rede, *root*, entre outros. Os testes de desempenho foram implementados no *firmware* base e automatizados em *Scripts* Python. Entre Python e ESP32, e/ou vice-versa, todos os dados trafegam via MQTT no formato JSON, visando facilitar a captação e posterior análise das informações. As versões dos softwares utilizados foram: Python 3.8.10 (PYTHON SOFTWARE FOUNDATION, 2021) e ESP-IDF v3.3.2 (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2018).

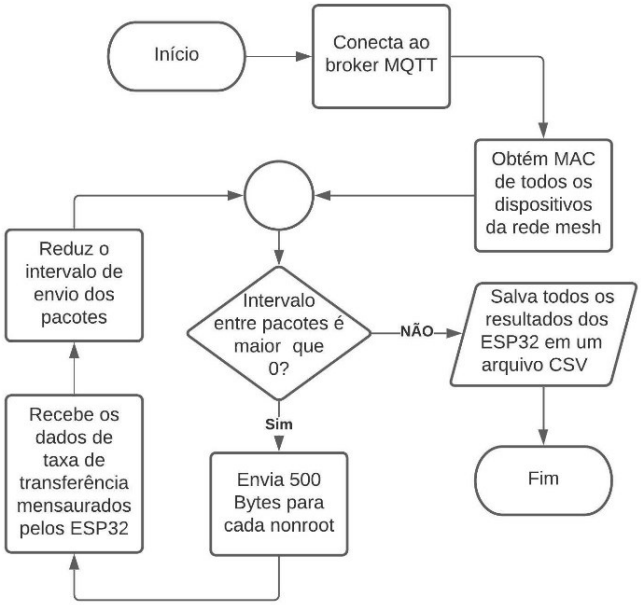
3.1 Teste de taxa de transferência

O teste de taxa de transferência foi implementado no ESP32 utilizando como referência o algoritmo do arquivo *example_spp_acceptor_demo.c* (ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, 2020), fornecido na pasta de exemplos do ESP-IDF.

Inicialmente o *Script* Python obtém os endereços de todos os *nonroots* da rede. Após a listagem dos dispositivos, são enviados pacotes de 500 bytes, variando de maneira

automatizada os intervalos de tempo entre os envios. O ESP32 devolve para o Python, via MQTT, os valores de *throughput* mensurados. No final dos testes, os resultados são salvos em um arquivo do tipo *comma-separated values* (CSV). Vide fluxograma na Figura 6.

Figura 6 - Fluxograma do teste da taxa de transferência.

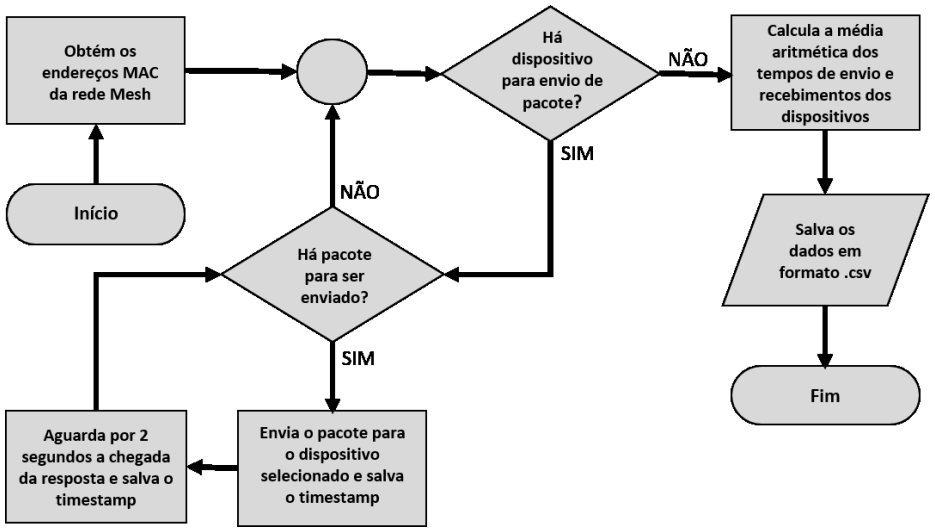


Fonte: elaborado pelo autor (2021).

3.2 Teste de latência

O *script* Python, cujo fluxograma é exibido na Figura 7, envia um pacote para um determinado *node* e salva o *timestamp* neste momento em uma lista. O *node* recebe esse pacote e o encaminha de volta. Assim que o cliente MQTT, através do *script*, recebe a mensagem, o segundo *timestamp* desta mensagem é salvo em uma lista.

Figura 7 – Fluxograma do teste de latência.



Fonte: elaborado pelo autor (2021).

Esse processo é repetido 50 vezes por *node*, e no final dos envios o *timestamp* atual é subtraído pelo *timestamp* anterior, onde ainda é calculada a média aritmética destes tempos. A cada teste concluído a resposta é salva em formato CSV para cada endereço MAC da rede.

3.3 Teste de perda de pacotes

Neste teste, o *script* Python envia pacotes numerados para cada *node*, com variação automatizada dos intervalos de tempo entre envios. Após o envio, aguardam-se 10 segundos para que o *node* retorne os pacotes, verificando quais e quantos retornaram. No final dos testes é gerado um arquivo CSV com os resultados.

3.4 Teste de tempo de construção de rede

Na verificação do tempo de construção de rede, o Python requisita que todos os *nodes*, incluindo o *root*, acionem a função implementada no ESP32 que provoca o *reboot* do dispositivo. Após o reinício, o Python monitora quanto tempo leva para todos os nodes se reconectarem à rede.

3.5 Teste de *healing time*

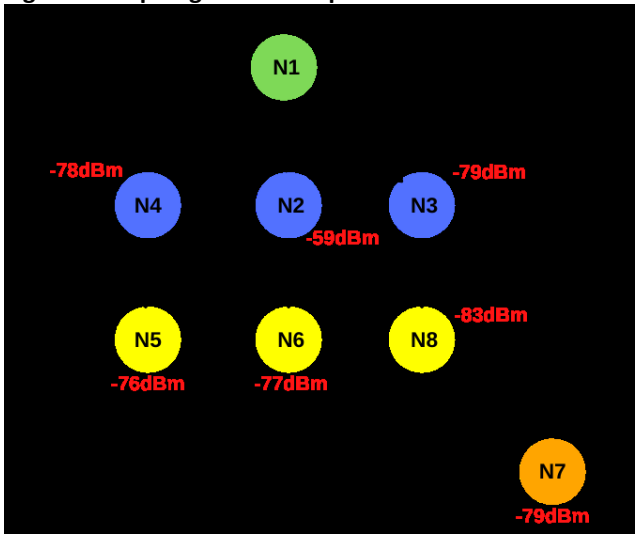
No teste de recuperação da rede, o *script* Python provoca o *reboot* do *root*, após isso, manualmente, esse dispositivo é permanentemente desligado. O *script* monitora quanto tempo leva para a nomeação de novo *root* e reconexão de todos os *nodes*.

4 RESULTADOS E DISCUSSÕES

4.1 Taxa de transferência

Os testes de taxa de transferência foram realizados com a topologia de rede mostrada na Figura 8, sendo o N1 o *root* da rede.

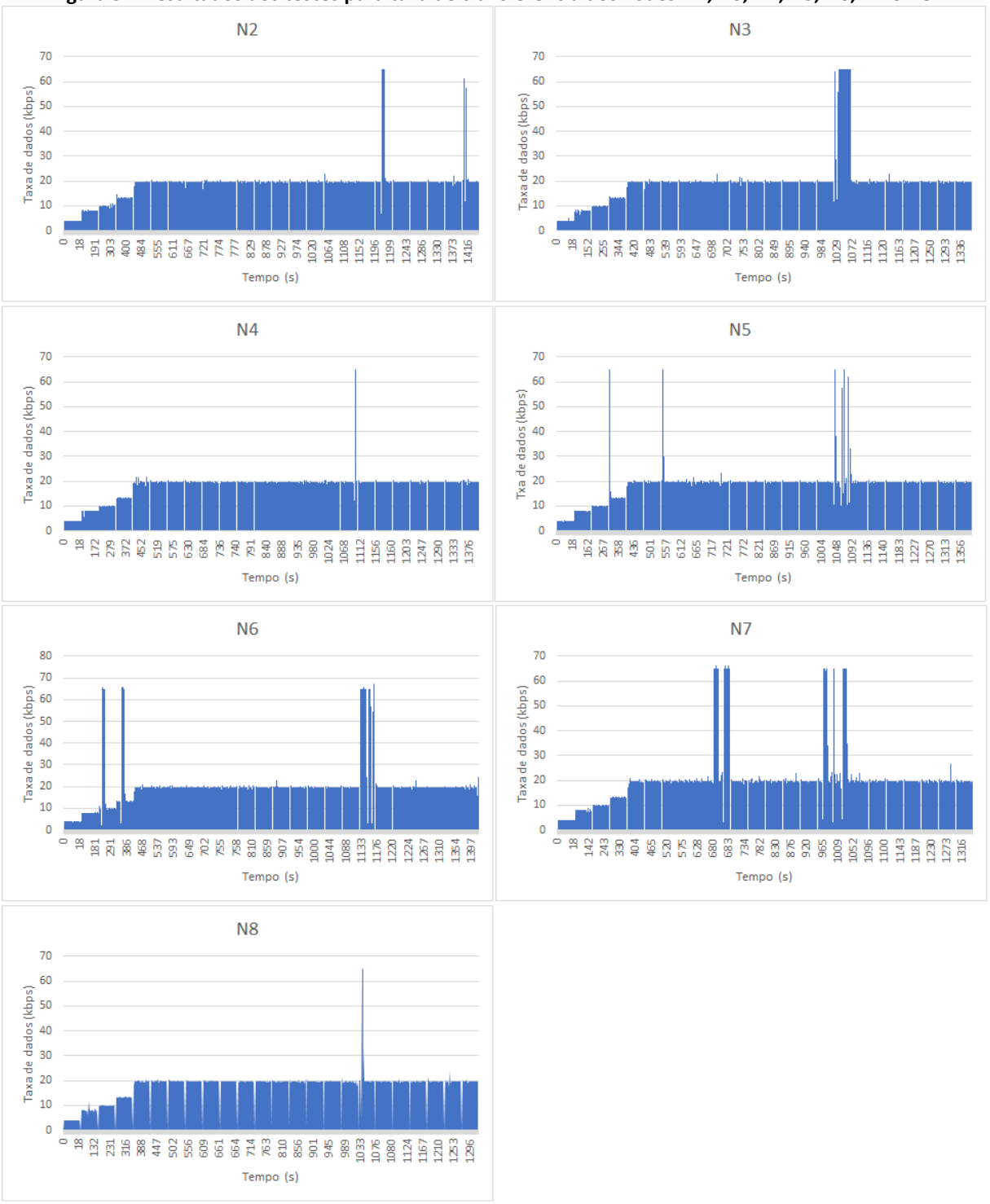
Figura 8 - Topologia formada para o teste de taxa de transferência.



Fonte: elaborado pelo autor (2021).

Conforme os gráficos da Figura 9, a seguir, a velocidade de transferência não ultrapassou 20 kbps, independentemente da distância ou intensidade de sinal, sendo vistos alguns *spikes* provenientes de pacotes atrasados na rede.

Figura 9 – Resultados dos testes para taxa de transferência dos *nodes* N2, N3, N4, N5, N6, N7 e N8.

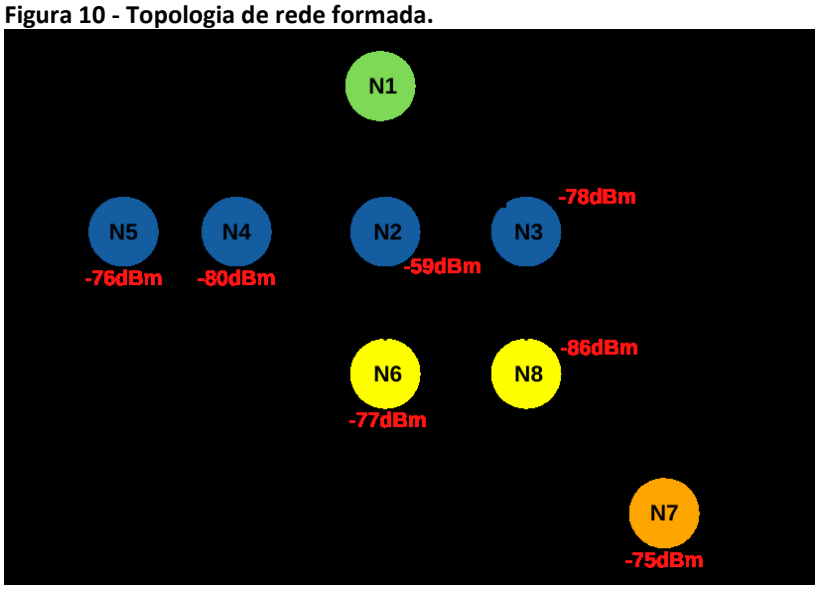


Fonte: elaborado pelo autor (2021).

4.2 Perda de pacotes

Nos resultados obtidos não houve mais que 1 pacote perdido, independente de distância, RSSI e intervalo de envio dos pacotes. A Figura 10 apresenta a topologia de rede formada, sendo o dispositivo N1 o *root* da rede.

Figura 10 - Topologia de rede formada.



Fonte: elaborado pelo autor (2021).

4.3 Latência

A topologia formada neste teste foi a mesma do teste de perda de pacotes (Figura 10), com a mesma intensidade de sinal RSSI. Após uma sequência de 5 testes consecutivos, foi obtida a Tabela 1, a seguir:

Tabela 1 – Teste de Latência.

Device	RSSI (dBm)	Latência (ms)	Nº de saltos
N2	-59	94	1
N3	-78	95	1
N4	-80	95	1
N5	-76	97	1
N6	-77	97	2
N7	-75	125	3
N8	-86	101	2

Fonte: elaborado pelo autor (2021).

A tabela mostra as variáveis que influenciaram no tempo de envio e chegada de pacotes, como o nó N6 que, apesar de estar em uma camada inferior, obteve o mesmo desempenho do nó N5 da camada superior, devido ao RSSI em relação ao seu nó pai ser de -77 dBm e deste último com o *root* ser de -59 dBm (ver figura 10 das camadas de rede). No resultado obtido é possível verificar que o fator salto teve pouca influência na latência, quando comparado ao fator distância e RSSI.

4.4 Tempo de construção de rede

Os tempos de construção de rede variaram de 22,11 a 94,52 segundos, conforme Tabela 2, a seguir. Foi observado que, geralmente, havia um aumento no tempo de construção da rede, quando **não** era selecionado o ESP32 denominado N1 como *root*, pois ocorriam mais "*rounds*" de votação durante a definição da rede *mesh*, visto que o referido ESP32 era o mais bem posicionado (Figura 4), em relação ao roteador, na topologia de rede aplicada.

Tabela 2 – Resultados do teste de *Building Time*.

Root nomeado	Tempo de construção (s)
N1	25,89
N1	25,63
N1	25,63
N1	25,64
N1	24,14
N1	25,9
N2	27,14
N5	50,77
N3	79,44
N1	24,88
N1	26,89
N1	24,38
N1	47,26
N6	53,8
N1	25,12
N1	26,13
N1	26,88
N1	22,11
N1	53,07
N2	94,52
N1	26,13
N1	30,66
N1	25,4
N1	26,64

Fonte: elaborado pelo autor (2021).

4.5 Tempo de recuperação de rede

Nos resultados do teste de recuperação de rede, os valores variaram de 14,82 a 78,68 segundos, de acordo com a Tabela 3. Como também aconteceu no teste de construção de rede, os melhores desempenhos foram obtidos quando o *node* N1 foi nomeado como *root*.

Tabela 3 – Resultados do teste de *Healing Time.*

Root nomeado	Tempo de recuperação (s)
N1	22,37
N2	48,51
N1	48,00
N6	78,68
N2	55,29
N1	47,76
N2	47,28
N5	34,43
N1	37,97
N2	14,82
N6	60,82
N1	18,59

Fonte: elaborado pelo autor (2021).

5 CONCLUSÃO

Na maior parte dos testes de desempenho, a rede *mesh* permaneceu estável, com raros momentos em que dispositivos apresentaram falha de conexão, e mesmo quando isso ocorreu, a rede se autorrecuperava, criando rotas alternativas. Um contraponto seria a taxa de transferência máxima de 20 kbps, no entanto, aplicações IoT normalmente não necessitam de grande volume de tráfego de dados. Dessa maneira, com a entrega de praticamente todas as mensagens entre os *nodes*, tempo de recuperação e construção de rede sempre abaixo de 2 minutos, latência e taxa de transferência que permitem a transmissão de milhares de *bytes* por segundo, distância entre nodes de até 75 metros, ainda considerando o tamanho dos pacotes que normalmente trafegam entre dispositivos IoT, este trabalho demonstrou a eficácia na construção de uma rede ESP-WIFI-MESH, provando que essa tecnologia pode ser uma escolha viável para aplicações em áreas rurais, onde em determinadas situações é praticamente impossível a criação de infraestrutura de rede ethernet convencional (cabeada ou Wi-Fi), como é o caso de estábulos, áreas de plantio e pastagem, granjas, fazendas, entre outros.

Como proposta de trabalhos futuros, seria possível a realização dos testes de desempenho variando a posição dos *nodes* e a quantidade de dispositivos na rede *mesh*. Essas verificações não foram executadas nesta pesquisa devido à limitação de tempo e força de trabalho, pois cada "*round*" de testes consumia várias horas, somadas ao *setup* inicial para montar e posicionar todos os equipamentos em campo.

REFERÊNCIAS

BLACK BOX. **Faster. Farther. Better. The evolution of 802.11**. 2018. Disponivel em: https://www.bboxservices.com/resources/blog/bbns/2018/04/30/802.11-wireless-standards-explained. Acesso em: 19 set. 2021.

BRIERE, D.; HURLEY, P. **Wireless home networking for dummies**. Indiana: Wiley, 2011. 364 p.

BURGESS, M. **What is the Internet of Things? WIRED explains**. WIRED, 2018. Disponivel em: https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot. Acesso em: 19 set. 2021.

CHEN, M. **Compare difference: WLAN vs. Wi-Fi**. 2021. Disponivel em: www.router-switch.com/faq/compare-difference-wlan-vs-wi-fi.html. Acesso em: 21 ago. 2021.

ECLIPSE MOSQUITTO. **An open source MQTT broker.** Disponivel em: https://mosquitto.org/. Acesso em: 21 ago. 2021.

ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD. **ESP32.** 2021. Disponivel em: https://www.espressif.com/en/products/socs/esp32. Acesso em: 21 ago. 2021.

ESPRESSIF SYSTEMS INC. **ESP8266 Mesh User Guide**. 2016. Disponivel em: https://www.signal.com.tr/pdf/cat/30a-esp8266_mesh_user_guide_en_v1.2.pdf. Acesso em: 2021.

FERNANDO K TECNOLOGIA. **ESP32 com Touch Button Capacitivo**. Fevereiro 2021. Disponivel em: https://www.fernandok.com/2018/02/esp32-com-touch-button-capacitivo.html. Acesso em: 21 ago. 2021.

FISHER, R. 60 GHz WPAN Standardization within IEEE 802.15.3c. *In*: INTERNATIONAL SYMPOSIUM ON SIGNALS, SYSTEMS AND ELECTRONICS, 2007. Montreal: IEEE, p. 103-105. DOI: 10.1109/ISSSE.2007.4294424. Disponível em: https://ieeexplore.ieee.org/document/4294424. Acesso em: 21 ago. 2021.

GERGELEIT, M. Autotree: Connecting Cheap IoT Nodes with an Auto-Configuring WiFi Tree Network. *In:* FOURTH INTERNATIONAL CONFERENCE ON FOG AND MOBILE EDGE COMPUTING (FMEC), 2019. IEEE, p.199-203. DOI: 10.1109/FMEC.2019.8795311. Disponível em: https://ieeexplore.ieee.org/document/8795311. Acesso em: 21 ago. 2021.

HIVEMQ. **MQTT Essentials: The Ultimate Guide to MQTT for Beginners and Experts**, 2021. Disponivel em: https://www.hivemq.com/mqtt-essentials/. Acesso em: 21 ago. 2021.

HOSTINGER. **O Que é Latência?** Hostinger Tutoriais, 2019. Disponivel em: https://www.hostinger.com.br/tutoriais/o-que-e-latencia. Acesso em: 21 ago. 2021.

HU, M.-X.; KUO, G.-S. Delay and throughput Analysis of IEEE 802.11 s Networks. IN: ICC WORKSHOPS-2008 IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS WORKSHOPS, 2008. IEEE, p. 73-78. DOI: 10.1109/ICCW.2008.19. Disponível em: https://ieeexplore.ieee.org/document/4531867. Acesso em: 21 ago. 2021.

INTEL. **Largura de banda vs throughput vs velocidade vs taxa de conexão**. Suporte, 2018. Disponivel em: https://www.intel.com.br/content/www/br/pt/support /articles/000026190/wireless.html. Acesso em: 21 ago. 2021.

KUM, Dong-Won *et al*. Mobility-aware hybrid routing approach for wireless mesh networks. *In*: THIRD INTERNATIONAL CONFERENCE ON ADVANCES IN MESH NETWORKS, 2010, IEEE, 2010. p. 59-62. DOI: 10.1109/MESH.2010.18. Disponível em: https://ieeexplore.ieee.org/document/5632079. Acesso em: 21 ago. 2021.

MAKSUTOV, Artem A. *et al*. Development of mesh networks based on the 802.11 family of standards. *In:* CONFERENCE OF RUSSIAN YOUNG RESEARCHERS IN ELECTRICAL AND ELECTRONIC ENGINEERING (EICONRUS), 2019. IEEE, 2019. p. 23-25. DOI: 10.1109/EIConRus.2019.8657121. Disponível em: https://ieeexplore.ieee.org/document/8657121. Acesso em: 21 ago. 2021.

NAIDOO, K.; SEWSUNKER, R. 802.11 Mesh mode provides rural coverage at low cost. **IEEE. AFRICON**. Windhoek, South Africa., 2007. 1-7. DOI: 10.1109/AFRCON.2007.4401647. Disponível em: https://ieeexplore.ieee.org/document/4401647. Acesso em: 21 ago. 2021.

OKI, O. A. *et al*. Using node position to improve the robustness of the IEEE 802.11s authentication mechanism for wireless mesh networks. *In*: INTERNATIONAL CONFERENCE ON ADAPTIVE SCIENCE & TECHNOLOGY (ICAST),6, 2014., Ota, Nigeria: IEEE, 2014, pp. 1-6. DOI: 10.1109/ICASTECH.2014.7068065. Disponível em: https://ieeexplore.ieee.org/abstract/document/7068065. Acesso em: 21 ago. 2021.

ORACLE. **O que é IoT?** 2021. Disponivel em: https://www.oracle.com/br/internet-of-things/what-is-iot/. Acesso em: 21 ago. 2021.

PFEIFER, E. A. **Uma rede de sensores sem fio em malha para monitoramento online do consumo de energia nos edifícios da Universidade Federal de Santa Catarina**: um estudo de caso. Araranguá, 2018. Disponível em: https://core.ac.uk/download/pdf/188205259.pdf. Acesso em: 21 ago. 2021.

POKHREL, S. R.; SHAKYA, S. Enhanced optimization of field based routing for macro mobility in IEEE 802.11s mesh. *In:* TENTH INTERNATIONAL CONFERENCE ON WIRELESS AND OPTICAL COMMUNICATIONS NETWORKS (WOCN), 2013. pp. 1-5. DOI: 10.1109/WOCN.2013.6616202. Disponível em: https://ieeexplore.ieee.org/document/6616202. Acesso em: 21 ago. 2021.

PRASINA, A.; THANGARAJA, M. Interoperability of Wireless Mesh and Wi-Fi network using FPGA for 4G solutions. *In*: INTERNATIONAL CONFERENCE ON RECENT TRENDS IN INFORMATION TECHNOLOGY (ICRTIT), 2011. pp. 491-496. DOI: 10.1109/ICRTIT.2011.5972364. Disponível em: https://ieeexplore.ieee.org/document/5972364. Acesso em: 21 ago. 2021.

PROOFPOINT. **What is Packet Loss?,** 2021. Disponivel em: https://www.proofpoint.com/us/threat-reference/packet-loss. Acesso em: 21 ago. 2021.

PYTHON SOFTWARE FOUNDATION. **Python language site:** documentation, 2021. Disponível em: https://www.python.org/doc/. Acesso em: 21 de ago. 2021.

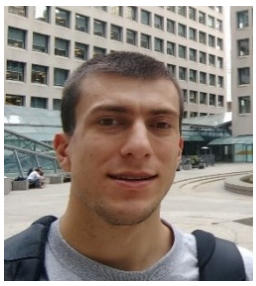
SOBRE OS AUTORES

_____

[i] **Gleydson Henrique de Alencar Santos**

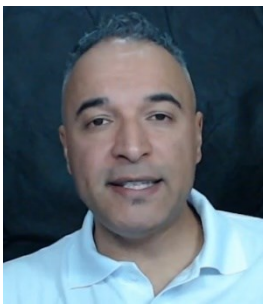
Possui graduação em Engenharia Elétrica pela Universidade Nove de Julho (2017) e formação de Técnico Eletrônico pela ETEC (2008). Atualmente é Engenheiro Desenvolvedor de Firmware na SatCompany, empresa do segmento de rastreamento de frotas. Tem experiência em projetos de dispositivos embarcados e IoT com ênfase em LoRa e LoRaWan. Atuou na área médica por 10 anos como Técnico em Equipamentos Biomédicos.

[ii] **Murilo José de Carvalho**



Possui graduação de Tecnólogo em Eletrônica Industrial pelo Instituto Federal de São Paulo (2013). Atualmente é Tecnólogo em Eletrônica Industrial no Instituto Federal de São Paulo. Tem experiência na área de Circuitos Elétricos/Eletrônicos com ênfase na área de Microcontroladores. Atou em empresa da área de energia fotovoltaica e iluminação LED.

[iii] **Fernando Simplicio de Sousa**



Professor da Faculdade SENAI no curso de Pós-Graduação em Sistemas Embarcados. Formado pela Universidade Federal do ABC (UFABC) no curso de Pós-Graduação (Mestrado) em Engenharia Elétrica (2017) e Pós-graduação (Lato Sensu) pela Instituição de Ensino Mackenzie (2010). Graduado em Tecnologia em Gestão de Pequenas e Médias Empresas pela Universidade Paulista (UNIP -2007) e em Tecnologia em Projetos Mecânicos pela Faculdade de Tecnologia de São Paulo (UNESP/FATEC-SP- 2004). Atualmente é aluno do curso de Pós-Graduação (Doutorado) em Energia pela UFABC. Autor do livro Programação BASIC para Microcontroladores 8051 (2006) pela Editora Erica. Currículo Lattes: http://lattes.cnpq.br/4579382987984065

[iv] **Leandro Poloni Dantas**

Doutor em Engenharia Elétrica pelo Centro Universitário FEI, cuja tese apresenta proposta de arquiteturas alternativas para projetos de processadores e microcontroladores com foco em sistemas de tempo real. Possui graduação em Engenharia Elétrica com ênfase em Eletrônica (2004) e mestrado em Dispositivos Eletrônicos Integradas (2008) ambos pelo Centro Universitário FEI. Dissertou sobre estudo de distorção harmônica em transistores MOSFET de porta circular. Atuou por 15 anos na indústria eletrônica no desenvolvimento de novos produtos para diferentes segmentos, com foco principal no projeto de equipamentos eletrônicos para o mercado de segurança eletrônica e automação predial. Desde 2009, vem lecionando em cursos de pós-graduação, graduação e de nível técnico em diferentes instituições paulistanas. Publicou livros sobre eletrônica digital e microcontroladores. (Texto extraído na íntegra da plataforma Lattes). Currículo Lattes: http://lattes.cnpq.br/6255986062207024