UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Utilização de Algoritmo de Consenso para Comunicação Segura entre drones em FANETs com Baixo Custo Computacional.

Daniel Paiva Fernandes

Itajubá, 29 de julho de 2023

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Daniel Paiva Fernandes

Utilização de Algoritmo de Consenso para Comunicação Segura entre drones em FANETs com Baixo Custo Computacional.

Dissertação submetida ao Programa de Pós-Graduação em Ciência e Tecnologia da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciências em Ciência e Tecnologia da Computação.

Área de Concentração: Computação Aplicada

Orientador: Prof. Dr. Jeremias Barbosa Machado
Coorientador: Prof. Dr. Sidney Nascimento Givigi Junior

29 de julho de 2023
Itajubá

UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Utilização de Algoritmo de Consenso para Comunicação Segura entre drones em FANETs com Baixo Custo Computacional.

Daniel Paiva Fernandes

Dissertação aprovada por banca examinadora em XX de Junho de 2023, conferindo ao autor o título de **Mestre em Ciências em Ciência e Tecnologia da Computação.**

Banca Examinadora:
Prof. Dr. Sérgio Ronaldo Barros dos Santos
Prof. Dr. Rodrigo Maximiano Antunes de Almeida

Itajubá
2023

Daniel Paiva Fernandes

Utilização de Algoritmo de Consenso para Comunicação Segura entre drones em FANETs com Baixo Custo Computacional/ Daniel Paiva Fernandes. – Itajubá, 29 de julho de 2023-

86 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Jeremias Barbosa Machado

Dissertação (Mestrado)
Universidade Federal de Itajubá - UNIFEI
Programa de pós-graduação em ciência e tecnologia da computação, 29 de julho de 2023.

1. Palavra-chave1. 2. Palavra-chave2. I. Orientador. II. Universidade Federal de Itajubá. III. Faculdade de xxx. IV. Título

CDU 07:181:009.3

Daniel Paiva Fernandes

Utilização de Algoritmo de Consenso para Comunicação Segura entre drones em FANETs com Baixo Custo Computacional

Dissertação submetida ao Programa de Pós-Graduação em Ciência e Tecnologia da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciências em Ciência e Tecnologia da Computação.

Trabalho aprovado. Itajubá, XX de Junho de 2023:

Prof. Dr. Jeremias Barbosa Machado
Orientador

Prof. Dr. Sidney Nascimento Givigi Junior
Coorientador

Prof. Dr. Sérgio Ronaldo Barros dos Santos

Prof. Dr. Rodrigo Maximiano Antunes de Almeida

Itajubá
29 de julho de 2023

Agradecimentos

Agradeço a minha esposa, que pacientemente me deu forças, carinho, compreensão, espaço, diante dos enormes sacrifícios que somente nós dois vivenciamos para concluir o presente programa de Mestrado. Esta vitória também é sua, Daniela, meu amor.

Agradeço ainda a meus familiares, principalmente minha mãe, por compreender a necessidade de privação da minha presença durante todo este período de estudos que foi um grande desafio, ao ter que conciliar o meu trabalho com as atividades acadêmicas.

Por fim e não menos importante, agradeço a meus orientadores Sidney Givigi e Jeremias Machado, por trilhar este caminho comigo e serem o meu norte para alcançar meus objetivos acadêmicos.

"A blockchain é a tecnologia. A Bitcoin é meramente a primeira manifestação convencional do seu potencial."
(Marc Kenigsberg, fundador do Bitcoin Chaser)

Resumo

O desenvolvimento de redes veiculares encontrou um cenário mais fértil com o avanço das comunicações ultra-confiáveis de baixa latência (URLLC), implantação de redes de quinta geração (5G) em todo o mundo, capacitação da computação de borda e a adoção da "Internet das Coisas" nas soluções em cidades inteligentes. Para garantir o sucesso dessas redes, é fundamental que o processo de comunicação seja confiável e seguro contra ações maliciosas e que a solução tenha baixa complexidade computacional e consumo de energia.

Entre redes veiculares que podem obter proveito dessas novas tecnologias, estão as FANETs (Flying Ad-Hoc Networks), que podem desempenhar papel crítico em missões de salvamento e reconhecimento de áreas de risco. Esta redes necessitam de uma solução que garanta transparência, segurança e tolerância à falhas de forma descentralizada para funcionarem corretamente.

Portanto, o presente trabalho propõe uma solução de prova de conceito para garantir comunicação tolerante a falhas em Redes Aéreas *Ad-Hoc* (FANETs) heterogêneas com baixo custo computacional emuladas, usando o algoritmo de consenso Proof of Elapsed Time (PoET).

Palavras-chaves: IoV, UAV, FANET, *blockchain*, segurança, cibersegurança, *Hyperledger*, *Sawtooth*, PoET, Mininet, Containernet, Redes, Ad-hoc

Abstract

The development of vehicular networks has found a more fertile scenario with the advancement of ultra-reliable low latency communications (URLLC), deployment of fifth generation (5G) networks worldwide, empowerment of edge computing and adopting "Internet of Things" solutions in smart cities. To guarantee the success of these networks, it is essential to ensure that the communication process is reliable, safe from malicious actions, and that the solution has low computational complexity and energy consumption.

Among vehicular networks that can take advantage of these new technologies are FANETs (Flying Ad-Hoc Networks), which can play a critical role in rescue missions and reconnaissance of risk areas. These networks need a solution that guarantees transparency, security and fault tolerance in a decentralised way to function correctly.

Therefore, the present work proposes a proof-of-concept solution to ensure crash-fault tolerant communication in emulated heterogeneous Flying Ad-Hoc Networks (FANETs) using the Proof of Elapsed Time (PoET) consensus algorithm.

Key-words: IoV, UAV, FANET, blockchain, security, cybersecurity, 5G, Hyperledger, Sawtooth, PoET, Mininet, Containernet, Network, Ad-hoc

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

5G	*Fifth Generation*	18
6G	*Sixth Generation*	36
API	*Application Program Interface*	36
AVN	*Avian Network*	40
B-TSCA	*Blockchain Assisted Trustworthiness Scalable Computation for V2I Authentication*	41
B.A.T.M.A.N	*Better Approach To Mobile Ad-hoc Networking*	56
BC	*Blockchain*	19
BFT	*Byzantine Fault Tolerance*	40
BSD	*Berkeley Software Distribution*	47
CFT	*Crash Fault-Tolerante*	19
CIA	*Confidentiality, Integrity, Availability*	23
CPU	*Central Processing Unit*	12
D2D	*Device to device*	36
DFD	*Data Flow Diagram*	24
DoS	*Denial of Service*	25
DRL	*Deep reinforcement learning*	41
DSRC	*Dedicated short-range communications*	43
FANET	*Flying Ad-Hoc Network*	10
GCS	*Ground Control Station*	10
GPS	*Global Positioning System*	18
IA	Inteligência Artificial	38
IEEE	*Institute of Electrical and Electronic Engineers*	33
IoV	*Internet of Vehicles*	33
IPFS	*InterPlanetary Filesystem*	40
LPWAN	*Low Power Wide Area Network*	44
MAC	*Medium Access Control*	41
MEC	*Mobile Edge Computing*	43
MITM	*Man-in-the-middle*	25
NFV	*Network Functions Virtualization*	42
NR	*New Radio*	43
OSI	*Open System Interconnection*	27
PBFT	*Practical Byzantine Fault-Tolerance*	18
PICOC	*population, intervention, comparison, outcomes and context*	32
PoC	*Proof-of-Concept*	41

PoET	*Proof-of-Elapsed-Time*	18
PoT	*Proof-of-Traffic*	40
QoS	*Quality of Service*	42
RAM	Random Access Memory	11
REST	*Representational State Transfer*	12
RSUs	*Roadside Units*	41
SDN	*Software-Defined Networking*	40
SDWN	*Software-defined wireless network*	47
SGX	*Software Guard Extensions*	48
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege	24
UAS	*Unmaned Aerial System*	24
UAV	*Unmaned Aerial Vehicle*	18
URLLC	*Ultra-reliabe Low Latency Conection*	18
V2I	*Vehicle-to-Interfaces*	42
V2V	*Vehicle-to-Vehicle*	40
V2X	*Vehicles to Everything*	21
VANET	*Vehicular Ad-Hoc Network*	33

Sumário

1 Introdução

A operação de veículos autônomos pode se tornar mais robusta com a adoção de conexões de comunicação de baixa latência ultra-confiáveis URLLC (*Ultra-reliabe Low Latency Conection*) fornecidas por tecnologias de comunicação móvel 5G (*Fifth Generation*) e além. No entanto, abordagens tradicionais que garantem conexões criptografadas e seguras podem ter altos custos computacionais, uma característica que não é desejável em um cenário em veículos autônomos com recursos limitados, dependência de energia e exigir comunicação *ad-hoc* quase instantânea para lidar com decisões em tempo real.

Uma das aplicações das Redes Ad-Hoc Voadoras FANET está no uso de enxames para realizar missões críticas, como reconhecimento de áreas de risco, e apoio em resgates [NOOR et al., 2020]. Para que um enxame funcione corretamente, a comunicação entre os Veículos Aéreos Não Tripulados conhecidos como UAV (*Unmaned Aerial Vehicle*) participantes deve ser à prova de falhas para evitar problemas como colisões de veículos ou corrupção nos dados da missão.

Em um enxame de FANET em trânsito, também é fundamental que os veículos estejam cientes da trajetória que precisam realizar para completar uma missão. Essas informações estão sujeitas a vários tipos de ataques [KUMARI et al., 2020], como falsificação de GPS (*Global Positioning System*), interferência [GUPTA et al., 2021b] e negação de espaço aéreo [DASU; KANZA; SRIVASTAVA, 2018].

Como um sistema distribuído, uma FANET precisa ser tolerante a falhas e resiliente a ameaças, como mau funcionamento de processos, perda de comunicação e ataques de agentes mal-intencionados. Portanto, é importante que os dispositivos funcionem e tomem as decisões corretas, mesmo quando um componente apresentar mau funcionamento.

Nesses cenários, um algoritmo baseado em consenso é uma escolha confiável para garantir que sistemas distribuídos como uma FANET possam alcançar a resiliência necessária para cumprir sua missão [WANG et al., 2021a].

Alguns algoritmos de consenso carregam esses recursos. Entre eles, o consenso PBFT (*Practical Byzantine Fault-Tolerance*) fornece uma solução tolerante a falhas com execução de tarefas redundantes e sistema de votação que determina se a falha pode ser tolerada em um sistema com componentes pouco sincronizados [WENSLEY et al., 1978].

Por outro lado, alguns consensos não são tolerantes a falhas bizantinas, como *Raft* [ONGARO; OUSTERHOUT, 2014], que utiliza o conceito de *cluster leader* eleito entre candidatos pelos seguidores, no caso de indisponibilidade do ex-líder.

Outra opção é o PoET (*Proof-of-Elapsed-Time*). Esse algoritmo fornece uma so-

lução tolerante a falhas CFT (*Crash Fault-Tolerante*) [LI et al., 2021] amostrando uma variável aleatória distribuída que decorre um período de tempo entre os pares, e o indivíduo com a menor amostra vence a eleição. Políticas e processos adicionais são usados para evitar trapaças.

Os algoritmos de consenso foram amplamente incorporados às estruturas de blockchain BC (*Blockchain*) e aplicado a diferentes propósitos. O *Hyperledger Sawtooth* é um framework que traz de forma modular os algoritmos de consenso mencionados acima como parte de sua solução para alcançar a confiabilidade e a comunicação tolerante a falhas em sistemas distribuídos.

Para que a análise da aplicabilidade desta solução seja bem-sucedida, faz-se necessário a implementação de um ambiente de simulação que permita ao pesquisador testar diferentes topologias, com diferentes especificações e dimensões técnicas dentro de um ambiente controlado e de forma acessível. Diante disso, a utilização de um software capaz de simular a comunicação entre diferentes tipos de nós numa rede *ad-hoc* é de grande importância não somente para alcançar o objetivo do presente trabalho, mas também para disponibilizar à comunidade científica a extensão desta simulação.

Para tanto, foi elegida uma variação da ferramenta conhecida como *Mininet* [FONTES et al., 2015], para quer suas funcionalidades fossem estendidas para simulação de casos de teste como o proposto pelo presente trabalho.

A elaboração do projeto foi planejada em duas etapas executadas em paralelo, quais sejam, a elaboração da base teórica e o desenvolvimento do experimento e da solução.

A primeira fase da elaboração teórica, como apresentado nos Capítulos 2 e 3 consistiu em:

1. condução de uma revisão sistemática da literatura à respeito do tema, para estabelecimento do problema central e potenciais soluções a serem desenvolvidas,
2. revisão teórica dos conceitos fundamentais envolvidos no trabalho.

A segunda etapa, que seria o desenvolvimento do experimento e implementação da solução, foi estruturado:

1. na escolha das tecnologias candidatas a serem aplicadas no experimento;
2. na análise dos algoritmos de consenso de acordo com o *framework* selecionado;
3. implementação de provas de conceito;
4. execução de testes e coletas do resultado.

1.1 Organização do trabalho

A presente dissertação está organizado da seguinte forma. A motivação e contribuições estão descritas no Capítulo 1. A revisão teórica sobre o tema está descrito no Capítulo 2. Trabalhos relacionados ao tópico são discutidos no Capítulo 3. A Seção 3.2 descreve o processo de planejamento da revisão sistemática de literatura. Por sua vez, os desafios atuais no esquema de comunicação das arquiteturas FANET estão descritos na Seção 3.7. O Capítulo 4 explica a arquitetura projetada como referência para a simulação e as ferramentas, frameworks e plataformas utilizadas para simular os cenários propostos por este artigo e implementar o projeto. Os cenários de teste são detalhados no Capítulo 5. A seção 5.4 apresenta os resultados e a Seção 5.5 as discussões. Por fim, o Capítulo 6 traz as conclusões e trabalhos futuros relacionados a esta pesquisa.

2 Revisão Teórica

2.1 Conceitos sobre veículos aéreos não tripuladas

Os veículos aéreos não tripulados, também conhecidos como UAV, vêm evoluindo rapidamente ao longo dos últimos anos, em diversas áreas de aplicação civil e militar, com a melhoria de peças e componentes, *hardware* e *software* responsáveis pelos controladores desta categoria de aeronave [VALAVANIS, 2007].

Primariamente desenvolvido para utilização militar na Primeira Guerra Mundial, ao longo dos anos e com investimentos governamentais massivos [VALAVANIS, 2007], a tecnologia se popularizou e o custo na produção de UAVs tornou esses dispositivos acessíveis ao consumidor final.

Outro fator que contribui para o crescimento exponencial do uso desta categoria de veículo são os avanços tecnológicos nos sistemas de comunição entre veículos entre si e com outros tipos de dispositivos, também conhecido como comunicação V2X (*Vehicles to Everything*).

Da mesma maneira que novos processos de comunicação trazem novos paradigmas, também introduzem problemas relacionados à segurança e confiabilidade, tanto a nível de *software* quanto *hardware*, além de atrair a intenção de agentes mal-intencionados [MEIJERS et al., 2022].

Tais preocupações podem ser mitigadas de várias maneiras, como aplicação de soluções arquiteturais, ou protocolos de comunicação, como será visto mais adiante.

2.1.1 Categorias de redes UAVs

Redes UAVs podem ser categorizadas de acordo com a sua arquitetura, dependendo da forma como a interação com o controlador central é estabelecida.

2.1.1.1 Centralizadas

Como o nome sugere, redes de UAVs centralizadas possuem um controlador central e, portanto, um ponto em comum de falha.

Alguns exemplos de arquitetura incluem: *UAV-GCS*, onde os UAVs se conectam diretamente com um estação base central GCS; *UAV-satélite*, quando a conexão dos UAVs é estabelecida com satélites artificiais que orbitam a superfície terrestre; *UAV-celular*, se os UAVs recorrem a tecnologia de comunicação móvel para transmitir e receber informações [TSAO; GIRDLER; VASSILAKIS, 2022].

2.1.1.2 Descentralizadas

No caso das redes veiculares aéreas descentralizadas, os UAVs podem se comunicar diretamente ou indiretamente com a GCS, servir de *gateway* ou ponto de acesso entre si para comunicarem-se dinamicamente, como no caso das redes *ad-hoc* [TSAO; GIRDLER; VASSILAKIS, 2022].

Redes heterogêneas são compostas por veículos com especificações de hardware diferentes e podem ainda assim colaborar entre si, podendo ou não juntar-se dinamicamente à rede, o que traz maior flexibilidade à rede.

2.2 FANETs

[TSAO; GIRDLER; VASSILAKIS, 2022] define FANET como uma rede que estende sua cobertura operacional usando uma topologia composta por vários nós, que permite UAVs heterogêneos e homogêneos de comunicar entre si.

2.2.1 Características

Tendo em vista a estrutura de múltiplos nós, as FANETs possuem maior escalabilidade, adaptabilidade e auto-organização, se comparada com uma rede de um único nó [TSAO; GIRDLER; VASSILAKIS, 2022].

FANETs também podem interagir dentro de seu próprio grupo ou cooperar com outros grupos, trazendo maior mobilidade e flexibilização na topologia da rede densidade de nós que se comunicam [TSAO; GIRDLER; VASSILAKIS, 2022].

2.2.2 Aplicações

Os veículos aéreos não-tripulados possuem papel publicamente reconhecido em atividades governamentais [SALAMH; KARABIYIK; ROGERS, 2021], relacionados ao monitoramento, segurança nacional e logística [LOKE, 2015].

A descentralização de uma central de controle das FANETs permite que este tipo de rede de drones tenha maior mobilidade [TSAO; GIRDLER; VASSILAKIS, 2022], e têm se popularizado em razão da crescente acessibilidade e versatilidade desses dispositivos. No contexto das FANETs, a mobilidade é uma característica crucial, pois os drones precisam ser capazes de se mover em conjunto e mudar rapidamente de posição para se adaptarem a mudanças nas condições de voo e na demanda de rede.

Outras aplicações dos UAVs incluem o monitoramento ambiental para detectar alterações em áreas naturais, como incêndios florestais, inundações e derramamentos de óleo [PASANDIDEH et al., 2022]. Também é possível utilizar drones equipados com câme-

ras para fins de segurança e vigilância, vigiando áreas remotas e protegendo infraestruturas críticas. A entrega de carga é outra aplicação possível, permitindo o transporte rápido de suprimentos médicos, comida e outros itens em áreas remotas ou de difícil acesso [LOKE, 2015]. As FANETs também podem ser utilizadas para fornecer serviços de telecomunicações em áreas com infraestrutura limitada ou inexistente, através de drones equipados com antenas que criam uma rede *ad hoc* [SAAD; BENNIS; CHEN, 2020]. Por fim, a criação de mapas detalhados de áreas remotas é outra aplicação possível das FANETs, útil para pesquisas científicas e planejamento urbano, bem como para mapear áreas de desastres [ASNAFI; DASTGHEIBIFARD, 2018]. Com o avanço da tecnologia de drones, novas aplicações para as FANETs certamente surgirão no futuro.

2.2.3 Comparação com outras redes veiculares

Se comparadas a outras redes veiculares, em particular com veículos terrestres, é possível notar que as FANETs possuem algumas vantagens. Por se tratarem de veículos aéreos, a latência e o alcance na comunicação sem fio acaba sendo maior [AL-EMADI; AL-MOHANNADI, 2020], tendo em vista a ausência de obstáculos físicos.

Por outro lado, por se deslocarem no espaço aéreo, o planejamento da trajetória deve levar em consideração um eixo adicional para controle na topologia da rede. Além disso, condições ambientais como ventos e turbulências podem adicionar complexidade na estabilidade das trajetórias. Por fim, FANETs tendem a ser mais complexas por poder possuir UAVs com especificações físicas diferentes umas das outras, tendo que levar em consideração planejamento de trajetória mais complexos possuindo poder de processamento computacional restritos e com energia limitada [WHEEB, 2022].

2.3 Ameaças em FANETs

Assim como qualquer sistema computacional, as FANETs estão sujeitas as inúmeras ameaças, dentre as quais os ataques cibernéticos. Agentes maliciosos podem perpetrar ataques visando potenciais vulnerabilidades, conforme descrito no Capítulo 1.

Diante disso, a Segurança da Informação em sistemas veiculares aéreos é uma dimensão que merece atenção especial.

2.3.1 Conceitos de segurança cibernética

Existem conceitos amplamente utilizados como base para a implementação de políticas, procedimentos e tecnologias que visam garantir a segurança da informação em uma organização. A tríade primordial da segurança da informação é conhecida pelo acrônimo CIA (*Confidentiality, Integrity, Availability*), que representam os conceitos fundamentais

de confidencialidade, integridade e disponibilidade (do inglês *availability*). Eles são descritos em diversas publicações e normas técnicas, como a *ISO 27001:2013* e o *NIST SP 800-53*.

A Confidencialidade diz respeito à proteção da informação contra acessos não autorizados, garantindo que apenas pessoas autorizadas possam acessá-la.

Já a Integridade está relacionada à garantia de que a informação não foi modificada de forma não autorizada ou acidental.

A Disponibilidade refere-se à garantia de que a informação esteja sempre acessível às pessoas autorizadas quando elas precisarem dela [HINTZBERGEN, 2018].

Além destes, outros princípios compreendem a base da segurança cibernética, como o não-repúdio. Este princípio garante que uma entidade não pode negar a autoria ou envolvimento em uma transação [JOSHI; SHARDA, 2001].

Estes conceitos são basilares no processo de modelagem de ameaças e mitigação de riscos, pois integram os controles de segurança deste processo.

2.3.2 Modelagem de ameaças

Uma forma de identificar tipos de ameaças no contexto de uma arquitetura, é por meio de técnicas já consolidadas para análise de risco, como diagramas de fluxo de dados, ou DFD (*Data Flow Diagram*), e o modelo STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege), utilizado e adaptado por [SALAMH; KARABIYIK; ROGERS, 2021] para UAS (*Unmaned Aerial System*).

O DFD tem por objetivo capturar os principais componentes de um Sistema de Informações, como os dados se movem dentro do sistema, pontos de interação do usuário e os limites de confiança. O DFD deve complementar a compreensão sobre o fluxo de informações, identificar conjuntos de dados e subconjuntos compartilhados entre sistemas, identificar aplicativos transacionando dados e destacar a classificação dos dados que estão sendo transmitidos

Por sua vez, o método STRIDE é um acrônimo para potenciais ameaças de segurança, sendo que cada um está relacionado à violação de um controle de segurança em particular:

- *Spoofing*: ou falsificação, consiste em passar por alguém; viola o controle de segurança da autenticação.
- *Tampering*: ou adulteração de dados sem autorização; viola o controle de segurança da integridade.

- *Repudiation*: evitar ser responsabilizado por uma ação; viola o controle da segurança do não-repúdio.
- *Information disclosure*: acessar dados sem permissão; viola o controle de segurança da confidencialidade.
- *Denial of service*: sobrecarregar o sistema para torná lo indisponível; viola o controle de segurança da disponibilidade.
- *Elevation of privilege*: conseguir mais privilégios do que o devido no sistema; viola o controle de segurança da autorização.

O processo de classificação de ameaças baseada em STRIDE, utiliza o DFD para mapear os riscos e seus respectivos vetores. De acordo com o resultado, é possível traçar um plano para mitigação das potenciais ameaças e assegurar a proteção dos controles de segurança.

[JACOBSEN; MARANDI, 2021] sugere um diagrama de fluxo de dados, conforme Figura 1. Com base neste tipo de modelagem, é possível identificar potenciais ameaças e vetores de ataque. A comunicação entre uma UAV e a estação base, ou entre UAVs através de uma interface REST com dados de telemetria pode, por exemplo, pode estar sujeita à potencial falta de validação de entrada por meio da adulteração (*tampering*). Isso pode levar a um ataque de DoS (*Denial of Service*), elevação de privilégio ou a uma divulgação de informações contra a estação base. Uma potencial mitigação seria checar se todas as entradas são verificadas usando uma abordagem de validação de entrada com uma lista valores aprovados.

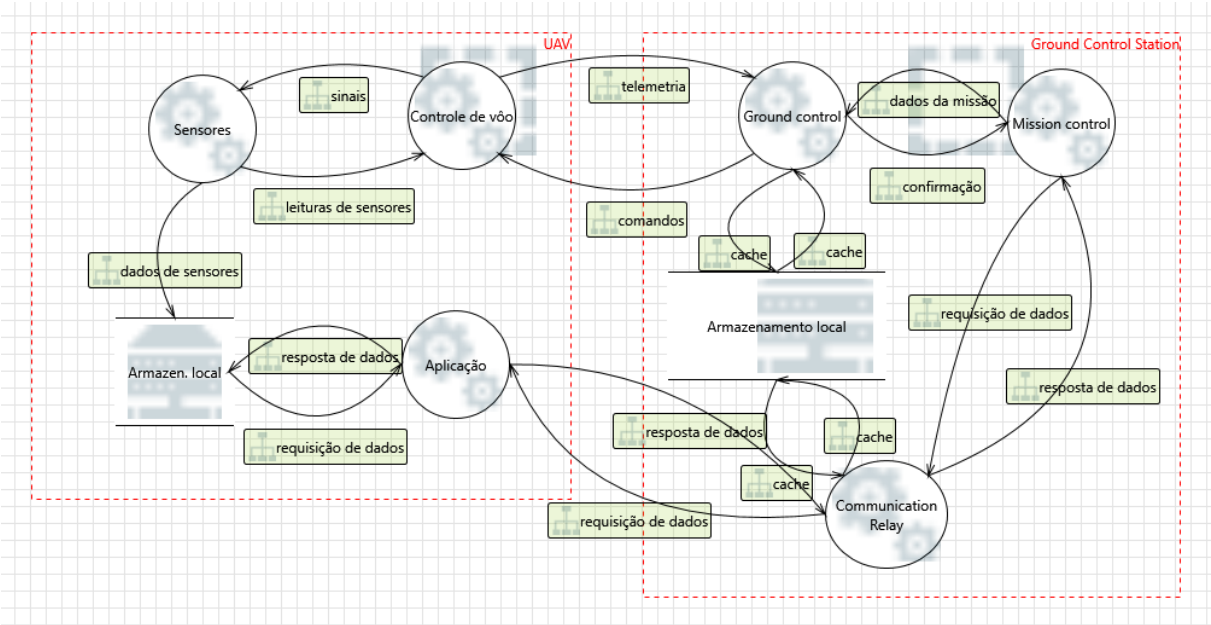
2.3.3 Ameças comuns em casos de FANET

Existem diferentes vetores de ataques cibernéticos capazes de provocar incidentes contra UAVs [SALAMH; KARABIYIK; ROGERS, 2021].

Entre as ameaças características contra FANETs, é possível citar:

- *Eavesdropping*: Ataque passivo onde o agente malicioso intercepta a comunicação na rede em busca de informações importantes sem alterar os dados ou o fluxo.
- MITM (*Man-in-the-middle*): Similarmente ao *eavesdropping*, há interceptação de dados. Contudo, o agente malicioso manipula as mensagens em trânsito.
- *Jamming*: Quando o agente malicioso interfere no sinal de comunicação da rede FANET utilizando algum mecanismo capaz de prejudicar a transmissão de rádio entre os UAVs e/ou a GCS.

Figura 1 – Diagrama de fluxo de dados sugerido por [JACOBSEN; MARANDI, 2021]



- *Replay*: Tipo de ataque onde o agente malicioso reenvia mensagens encriptadas ao GCS que é induzido a erro e acreditando ser uma conversa legítima, estabelece um canal de comunicação.
- *Insider*: Agente malicioso, normalmente um funcionário da empresa, que usa seus privilégios de acesso contra o sistema.
- *GPS Spoofing*: O atacante simula o sinal de GPS para obter controle do UAV.
- *Collision*: situação onde dois ou mais nós de uma FANET enviam dados simultaneamente em redes *half-duplex* e o fluxo de dados é interrompido.
- *Low link quality & high latency*: ocorre quando situações ambientais de mobilidade dos UAVs interferem na disponibilidade da FANET ou em sua capacidade de operação.
- *Black hole*: Quando um UAV malicioso engana os demais membros da rede informando falsamente o que possui a rota de rede mais curta, e então descarta os pacotes recebidos pelos UAVs legítimos .
- *Gray hole*: Ocorre quando um UAV malicioso altera os pacotes de uma rede, alternando periodicamente mensagens legítimas e ilegítimas para dificultar a identificação da ação maliciosa.
- *Wormhole*: ocorre quando UAVs ou *GCSs* maliciosas enviam pacotes de mensagens para enganar os nós legítimos da rede e redireciona pacotes para outra rede por túneis de comunicação com o intuito furtar dados relevantes.

- *Data tampering*: O agente malicioso modifica dados sensíveis, como chaves de sessão, tokens de acesso, leitura de sinais dos sensores do UAV.
- DoS: Ataque onde o agente malicioso sobrecarrega o alvo com uma quantidade de requisições que o sistema de destino não consegue suportar e o torna indisponível.
- *Selfishness*: ocorre quando um UAV está com seus recursos computacionais prejudicados por estar operando com baixa energia e, diante da degradação de recursos, a performance e a disponibilidade de FANET acaba sendo prejudicada.
- *Modification*: Este ataque ocorre quando um terceiro não autorizado modifica ou inunda com mensagens a rede atacada.
- *SYN Flood*: o agente malicioso envia uma grande quantidade de pacotes *SYN* falsos para um servidor, forçando que ele responda com um *SYN-ACK* para cada um deles, mantendo as conexões não finalizadas e sobrecarregando com conexões *half-open*.

A Figura 2 relaciona os diferentes tipos de ameaças e seus correspondentes vetores de ataque sob duas dimensões: pelos tipos de conexões e nós e pela camada de rede do modelo OSI (*Open System Interconnection*) [TSAO; GIRDLER; VASSILAKIS, 2022]. Já a Tabela 2 sumariza a relação entre os tipos de ameaça e seus controles de segurança, com os ataques apresentados na presente Seção.

Tabela 2 – Relação entre tipos de ataque por ameaça e controle de segurança afetado

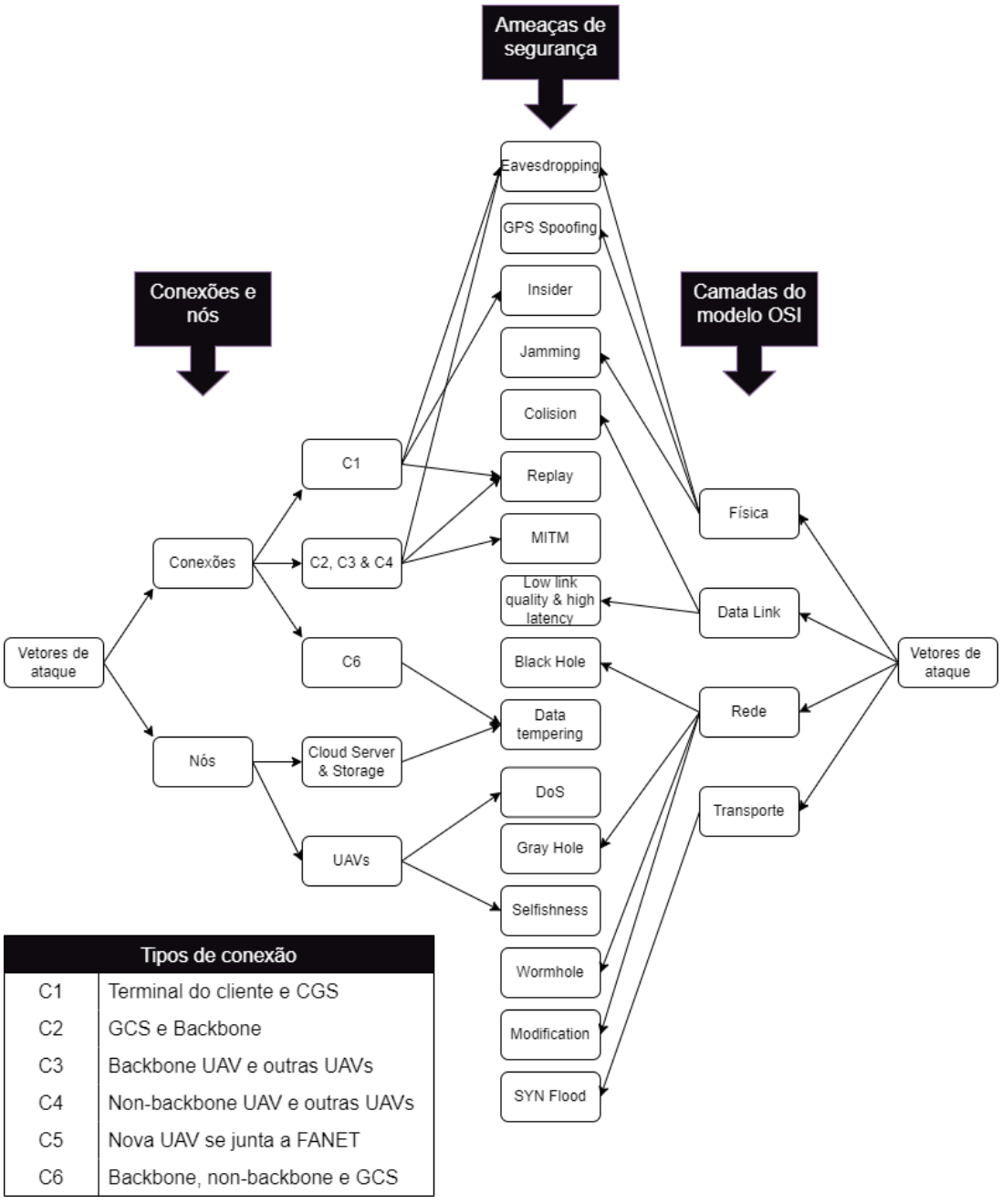
Ameaça	**Ataque**	**Controle de Segurança**
Spoofing	GPS Spoofing	Autenticação
Tampering	Data Tampering	Integridade
Repudiation	Modification	Não-repúdio
Information disclosure	Eavesdropping, MITM	Confidencialidade
Denial of Service	Jamming, Collision, Replay, LLQ&HL, Black Hole, DoS, Gray Hole, Selfishness, Wormhole, SYN Flood	Disponibilidade
Elevation of Priviledge	Insider	Autorização

2.4 Algoritmos de consenso em transações seguras e Blockchain

Algoritmos de consenso exercem um papel fundamental em *blockchain*, pois é por meio deles que os nós de uma rede concordam sobre o estado atual dos blocos, garantindo a descentralização de forma confiável.

Existem diversos algoritmos de consenso que podem adequar as *blockchains* de acordo com a sua utilização e características.

Figura 2 – Ameaças de segurança por vetor de ataque. Adaptado de [TSAO; GIRDLER; VASSILAKIS, 2022]



2.4.1 Histórico

A utilização mais proeminente do *blockchain* está nas criptomoeadas, com destaque à *Bitcoin* em 2008 [FU; WANG; SHI, 2020], que foi lançada em 2009 por uma pessoa ou grupo de pessoas que usavam o pseudônimo Satoshi Nakamoto. O objetivo do *Bitcoin* era criar uma forma de dinheiro digital descentralizada e segura que não dependesse de uma autoridade central para registrar e verificar transações.

A *blockchain* é uma estrutura de dados distribuída que foi utilizada para registrar e verificar todas as transações na rede *Bitcoin* sem a necessidade de uma autoridade

central [NAKAMOTO, 2009].

[KHAN; HARTOG; HU, 2022] ressaltam que, apesar de ter ganhado notoriedade com o advento da *Bitcoin*, os alicerces da *blockchain* vinham sendo estudados anos antes por matemáticos ao sugerirem um sistema baseado em *timestamps* em 1991, depois por criptógrafos e cientistas da computação em 1998 com a ideia de contratos inteligentes e solução de problemas baseados em processamento computacional, e por fim por um cientista alemão em 2000 com a ideia de blocos ligados entre si que poderiam ser referidos até um bloco gênese.

2.4.2 Conceito de algoritmo de consenso

Leslie Lamport primeiro descreve o conceito de algoritmos de consenso com o problema dos generais Bizantinos em 1982, garantindo que o dado não seja perdido em hipóteses de falha sistêmica [FU; WANG; SHI, 2020].

Os algoritmos de consenso permitem que um conjunto de nós pertencentes a uma rede se comportem como bancos de dados distribuídos [FERDOUS; CHOWDHURY; HOQUE, 2021]. Desta forma, esta rede pode permanecer funcional mesmo com a falha de nós, aumentando a confiança sem depender de uma entidade centralizada [NOFER et al., 2017].

Algo que se deve ser considerado ao escolher o algoritmo de consenso apropriado é se ele atenderá os requisitos mínimos de latência e *throughtput* da rede [MEIJERS et al., 2022], além de outros fatores como resiliência à falhas, particionamento de redes e comportamento dos nós [FERDOUS; CHOWDHURY; HOQUE, 2021].

2.4.3 Conceito de blockchain

[MEIJERS et al., 2022] define *blockchain* como sendo estruturas de dados especializadas pareadas numa rede distribuída que usam primitivos criptográficos com um algoritmo de consenso, que permitem a todos os membros daquela rede mantenham de forma cooperativa uma coleção ordenada de registros agrupados em blocos.

Segundo os autores, os blocos contém um código *hash* do bloco anterior a ele, formando desta forma uma corrente. Um novo bloco só será admitido com a concordância da maioria dos participantes da rede, reforçando ainda que os blocos já adicionados são imutáveis.

Para tanto, os sistemas de *blockchain* fazem uso de um algoritmo de consenso para assegurar esta concordância entre os nós da rede [FERDOUS; CHOWDHURY; HOQUE, 2021]. Por este motivo, a escolha entre os diferentes sistemas de *blockchain* devem levar

em consideração o algoritmo de consenso adotado em sua arquitetura, haja vista a relação intrínseca entre si.

Blockchains são categorizados em *permissionless*, quando a rede é aberta e qualquer nó pode se juntar à rede como um validador ou usuário, além de possuir seu conjunto de dados acessíveis e transparentes a todos, razão pelo qual também é conhecida como *blockchains* públicas [FERDOUS; CHOWDHURY; HOQUE, 2021]; enquanto que *blockchains permissioned* ou privadas [FERDOUS; CHOWDHURY; HOQUE, 2021] permitem que apenas nós autenticados possam figurar como validadores numa rede [MEIJERS et al., 2022], mantendo os dados restritos apenas a entidades confiáveis.

[FU; WANG; SHI, 2020] também classifica os algoritmos de consenso em *blockchains* de acordo com cada fase no modelo de processos. Modelos baseados em liderança focam na primeira etapa, que é a seleção do nó responsável em gerar os blocos. Quando a rede foca no sistema de votos para conseguir a aprovação da maioria dos nós, o modelo é baseado em votos. O terceiro modelo, baseado em comitê + votos, foca primariamente na escolha do líder e na adição de blocos. Finalmente, no modelo de contabilidade justa, todos os nós são líderes na fase de seleção e são responsáveis por empacotar as transações que geram.

Dadas suas características anteriormente citadas e tendo por base algoritmos de consenso em sua estrutura, a *blockchain* tem encontrado espaço para ser aplicada em vários outros campos, como contratos inteligentes, sistemas de votação eletrônica e gerenciamento de cadeia de suprimentos.

2.4.4 Aplicações em comunicações V2X

A utilização de *blockchains* em redes veiculares possuem um escopo direcionado às áreas de pagamentos e incentivos, reputação e autenticidade e autenticação de dados e marcação de tempo [MEIJERS et al., 2022].

Usuários de veículos automotivos, por exemplo, fazem recorrente uso de serviços que impliquem em transações monetárias, como abastecimento de combustível, estacionamento, pedágio, multas, impostos. Sendo assim, a aplicação mais notória da *blockchain*, as criptomoeadas, podem servir como método de pagamento automatizado destes serviços, se integrados com os veículos.

Outro uso seria a utilização de estruturas de *blockchain* para validar a reputação e a autenticidade de membros pertencentes a uma rede, ou da autenticidade do dados trafegados entre os nós, repudiando desta forma a ação de agentes maliciosos ou até mesmo de outros veículos em mal-funcionamento.

Por fim, mecanismos inerentes da tecnologia da *blockchain* podem contribuir para assegurar a integridade dos registros de dados compartilhados entre os pares que compõem

uma rede, dado a sua natureza de agir como uma forma de banco de dados distribuído. Ainda, dado a sua forma de funcionamento, os registros nos blocos armazenam o momento da ocorrência de determinada transação, garantindo a auditabilidade e rastreabilidade dos registros e ocorrências dos eventos que se procura investigar.

Diante disso, é possível exemplificar alguns casos em que a *blockchain* pode ser aplicado em redes veiculares, tais como na investigação de acidentes de trânsito, rastreamento na emissão de gases poluentes, pagamento de pedágios e estacionamentos [MEIJERS et al., 2022].

3 Revisão Sistemática da Literatura

Nesse capítulo são apresentados os trabalhos relacionados necessários para uma boa compreensão do desenvolvimento deste trabalho, elaborado a partir de uma revisão sistemática de literatura no processo de definição do tema da presente dissertação.

Inicia-se com uma breve explicação da metodologia e ferramentas empregadas. À seguir, são apresentadas as etapas de planejamento, além do processo de condução e extração dos resultados.

O capítulo é finalizado com a descrição do escopo do trabalho e do problema que se pretende solucionar.

3.1 Metodologia de pesquisa

O reforço da segurança nas redes de UAVs tem uma infinidade de características e abordagens dependentes de cada caso. Subsequentemente, existem vários trabalhos relacionados com este tema. Diante disso, a primeira etapa do projeto consistiu em estabelecer uma revisão sistemática de literatura para coletar a base teórica do presente trabalho e estabelecer qual seria o escopo de atuação pesquisa, antes de se apresentar uma simulação baseada em um cenário real.

Numa primeira etapa, o trabalho utilizou mapeamento sistemático para encontrar e selecionar o estado da arte da literatura relacionada ao uso de blockchain para garantir comunicação segura e econômica em redes veiculares.

Utilizou-se a ferramenta chamada *Parsifal* para gerenciar esta pesquisa [GARCíA-PEñALVO, 2017].

3.2 Planejamento

3.2.1 PICOC

Antes de estabelecer as questões de pesquisa, alguns aspectos importantes foram identificados utilizando o *framework* PICOC (*population, intervention, comparison, outcomes and context*) [WOHLIN et al., 2012], conforme Tabela 3.

3.2.2 Definição das questões de pesquisa

Após a definição dos aspectos a serem considerados, foram definidas as questões de pesquisa, como Tabela 4.

Tabela 3 – Definição da PICOC

Aspecto	Valor
Population	IoV (*Internet of Vehicles*), UAV, VANET (*Vehicular Ad-Hoc Network*)
Intervention	blockchain
Comparison	-
Outcome	security
Context	vehicular networks

Tabela 4 – Questões de Pesquisa

Questão	Descrição
QP-01	É possível garantir transações de dados seguras entre redes de veículos autônomos com modelos *blockchain*?
QP-02	Quais *frameworks* de *blockchain* são comumente usados em redes veiculares?
QP-03	É possível garantir que as redes 5G podem garantir comunicação ultra-confiável em redes veiculares?
QP-04	As soluções *blockchain* podem afetar o desempenho da comunicação em redes veiculares?
QP-05	Quais são os desafios futuros que as técnicas de *blockchain* podem superar para reforçar a segurança e a eficiência nas redes veiculares?
QP-06	É possível conceder soluções de baixo custo computacional para superar UAVs de recursos limitados?

3.2.3 Pesquisa de *String* e Fontes

O próximo passo consistiu em formular uma string de busca genérica para ser utilizada em bases de dados científicas [WOHLIN et al., 2012]:

("vehicles"OR "vehicle"OR "vehicular"OR "unmanned"OR "iov"OR "uav"OR "uavs") AND ("blockchain") AND ("cyber security"OR "security") AND 5G

As bases de dados digitais utilizadas para processar a busca neste estudo foram IEEE (*Institute of Electrical and Electronic Engineers*) Digital Library e *Scopus*. A segunda fonte inclui várias outras editoras.

3.2.4 Critérios de seleção

Definiram-se os critérios de inclusão e exclusão como linha de base para a seleção de estudos primários [WOHLIN et al., 2012].

Os critérios de inclusão listados na Tabela 5 foram considerados relevantes para a realização deste estudo. Por outro lado, os critérios de exclusão da Tabela 6 foram selecionados para definir o estudo como inelegível para a presente pesquisa.

3.2.5 Lista de verificação de avaliação de qualidade

Para avaliar a qualidade de cada estudo selecionado nas etapas anteriores, foram formuladas algumas questões de avaliação, conforme a Tabela 7. As possíveis respostas para cada questão eram "*sim*", "*parcialmente*" e "*não*", com peso de 1, 0,5 e 0, respectivamente. A pontuação máxima foi baseada no número de questões e no peso maior, ou seja, 7,0 pontos. A nota de corte é 5,0.

Tabela 5 – Critérios de inclusão

Critério	Descrição
CI-01	Publicado entre 2018 e 2021 - Figura 4
CI-02	Questões de pesquisa relevantes
CI-03	Estudo aborda dois ou mais tópicos relevantes para a pesquisa
CI-04	Tópico é a aplicação de *blockchain* para proteger redes IoV

Tabela 6 – Critérios de exclusão

Critério	Descrição
CE-01	Artigos que não estão totalmente disponíveis
CE-02	Documentos duplicados ou já incluídos
CE-03	Editoriais e artigos não revisados por pares
CE-04	Conteúdo não inglês-português
CE-05	Estudo diz respeito a poucos tópicos relevantes para o estudo
CE-06	O estudo está classificado incorretamente, incompleto e/ou pouco claro em termos das propostas de pesquisa

Tabela 7 – Questões de avaliação de qualidade

Pergunta	Descrição
AQ-01	O estudo é focado principalmente em VANETs?
AQ-02	O artigo aborda preocupações com uma solução de custo-benefício?
AQ-03	Os objetivos da pesquisa estão claramente especificados?
AQ-04	O estudo foi projetado para atingir esses objetivos?
AQ-05	As técnicas utilizadas estão claramente descritas e sua escolha justificada?
AQ-06	As variáveis consideradas pelo estudo são mensuradas adequadamente?
AQ-07	Os métodos de coleta de dados estão adequadamente detalhados?
AQ-08	Todos os tópicos (segurança, IoV ou UAVs, blockchain) são cobertos pela pesquisa?

3.3 Formulário de extração de dados

As seguintes perguntas foram definidas para responder à etapa de extração de dados:

- Os critérios de inclusão/exclusão foram informados?
- Os dados foram sintetizados?
- Foram relatados detalhes suficientes do estudo?
- A qualidade do estudo foi avaliada?

3.4 Condução

3.4.1 Strings de pesquisa de bibliotecas digitais

Considerando que cada biblioteca utiliza estruturas de strings de busca diferentes, a string de busca base mencionada na Subseção 3.2.3 foi adaptada para as seguintes fontes:

IEEE Digital Library: ("All Metadata": "vehicles"OR "All Metadata": "vehicle"OR "All Metadata": "vehicular"OR "All Metadata": "unmanned"OR "All Metadata": "iov"OR "All Metadata": "uav"OR "All Metadata": "uavs") AND ("All Metadata": "blockchain")

AND ("All Metadata": "cyber security"OR "All Metadata": "security") AND "All Metadata": 5G.

Scopus: TITLE-ABS-KEY (("vehicles"OR "vehicle"OR "vehicular"OR "unmanned"OR "iov"OR "uav"OR "uavs") AND ("blockchain") AND ("cyber security"OR "security") AND "5G")

3.4.2 Estudos importados

A busca, realizada em 15 de abril de 2021, resultou em:

Biblioteca Digital IEEE: 21 artigos.

Scopus: 48 artigos.

3.4.3 Seleção de estudos primários

Primeiramente, identificaram-se dois artigos duplicados a partir dos resultados e, em seguida, realizou-se a verificação manual de cada resumo de artigo para avaliar a pertinência do tema com as questões de pesquisa, rejeitando 26 artigos.

Dos resultados restantes reunidos nas etapas acima, aplicou-se uma API (*Application Program Interface*) de *cluster* de texto da ferramenta *MeaningCloud* sobre o "*Abstract*" de cada estudo, para extrair e processar o texto com inteligência computacional, aplicar recursos linguísticos e identificar informações significativas do conteúdo [IBRAHIM; SUPERVISOR, 2018].

Os resultados do agrupamento de texto recuperaram os termos por grupos e a classificação do termo dentro de cada resumo.

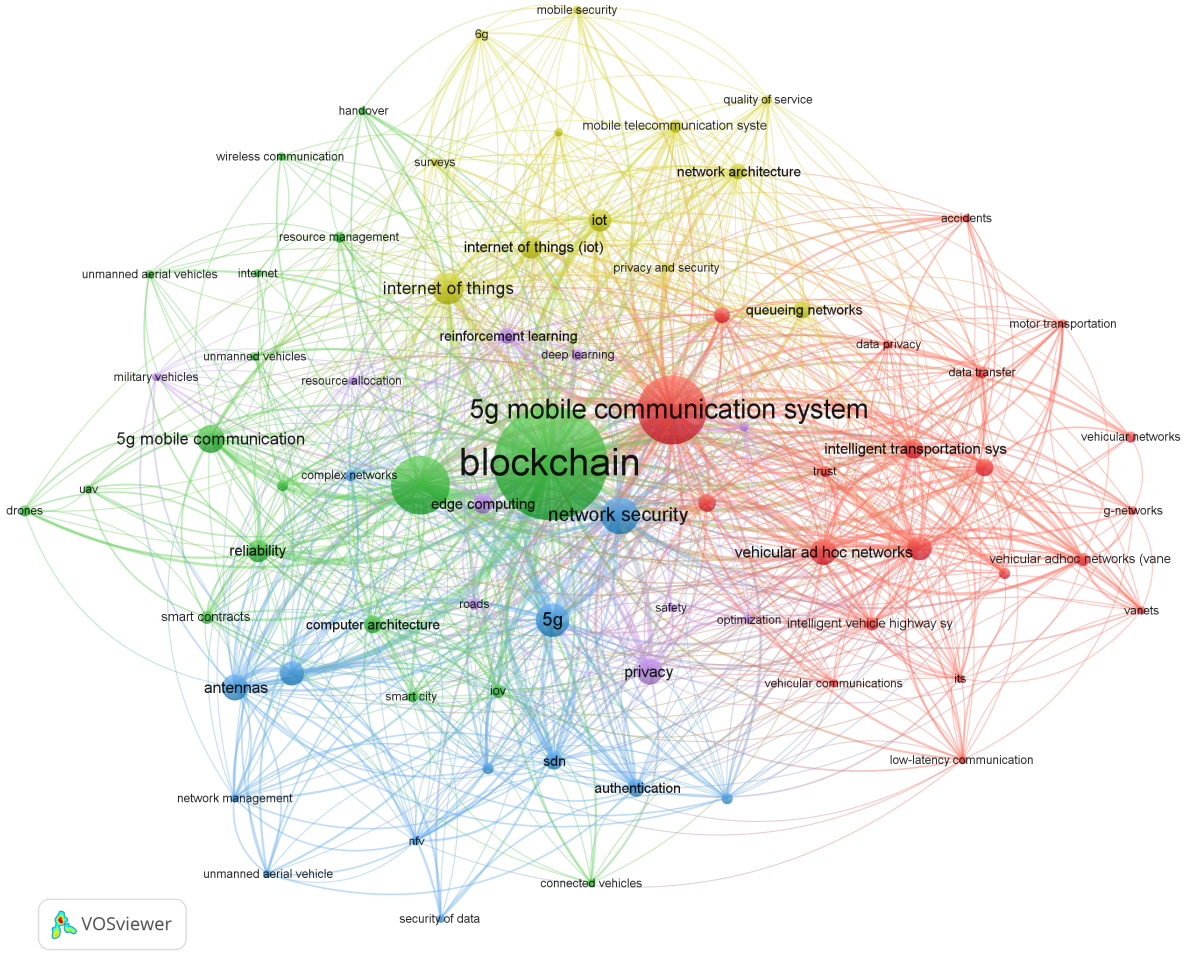
Em seguida, destacou-se os termos relevantes para esta pesquisa (5G, Segurança e Privacidade, Comunicação Móvel, Sistema Autônomo, *Deep Reinforcement Learning*, *Edge Nodes*, *Edge Computing*, Tempo de Resposta, Sistema Autônomo, Redes Veiculares, Comunicações V2X, Mineração, D2D (*Device to device*), *Blockchain* , UAV, *Computation Offloading*, *Resource Allocation*, *Quality of Service QoS*, *Drones*, *Open Issues*, VANETs, 6G (*Sixth Generation*), *V2X Communications*, conforme mapa bibliométrico na Figura 3 foi fornecido por [VOSVIEWER, 2010]) e usou-se um pontuação ponderada do termo w_n com base na equação (3.1):

$$w = \sum_{r=1}^{N} 1/r \tag{3.1}$$

onde r é a classificação do termo no resumo do artigo.

A pontuação de peso W do artigo é expressa pela seguinte equação:

Figura 3 – Rede bibliométrica com 5 *clusters*



Fonte: VosViewer

$$W = \frac{w}{\underset{w=w_1,w_2,...w_n}{\text{argmax}} \; w}$$

.

Em seguida, cada artigo recebeu uma pontuação manual ($0.0 \leq m \leq 1.0$) com base na análise do resumo para evitar falsos negativos e obter a pontuação final f da média das duas pontuações.

$$f = \frac{W + m}{2}$$

.

Dos 69 artigos apresentados pela busca, 20 foram aprovados (19 aprovados sem intervenção manual e 1 falso negativo) e 20 foram selecionados (5) após a avaliação de qualidade.

Figura 4 – Artigos por ano

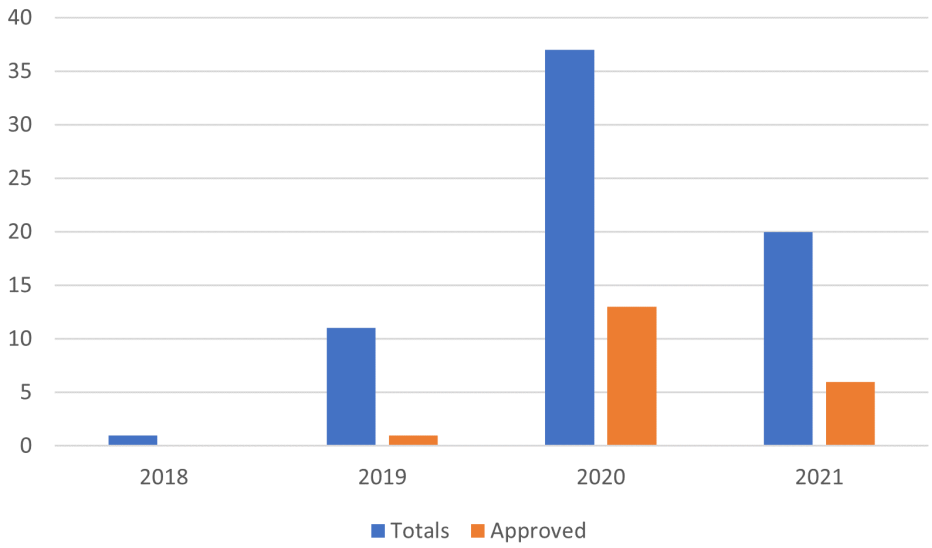
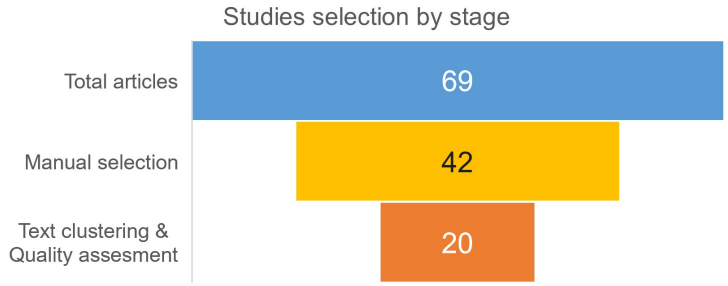


Figura 5 – Seleção de artigos por estágio



3.4.4 Análise de avaliação de qualidade

Todos os 20 artigos aprovados foram totalmente revisados e avaliados seguindo as questões de avaliação de qualidade (Tabela 7) conforme detalhado na Tabela 8.

3.5 Extração de dados e resultados

Como resultado das etapas anteriores, os seguintes artigos foram extraídos e resumidos.

O estudo elaborado por [MOHAMMED et al., 2020] explora aplicações BC na arquitetura *Multi-access Edge Computing* assistida por UAV para proteger e otimizar problemas de descarregamento (QP-04). Aborda o uso da computação de borda para superar os altos custos computacionais (QP-06), bem como o uso do aprendizado por reforço profundo na arquitetura proposta (QP-05), estruturada em três camadas.

[REEBADIYA et al., 2021] sugere uma solução baseada em BC para detecção inteligente e arquitetura de rastreamento para veículos autônomos usando redes URLLC 5G (QP-03) implantando algoritmos de IA (Inteligência Artificial) nos servidores de borda e

Tabela 8 – Seleção de estudos depois da extração de textos

Autor	Título	AQ-01	AQ-02	AQ-03	AQ-04	AQ-05	AQ-06	AQ-07	AQ-08
[REEBADIYA et al., 2021]	Blockchain-based Secure and Intelligent Sensing Scheme for Autonomous Vehicles Activity Tracking Beyond 5G Networks	0.5	0.5	1.0	1.0	1.0	0.5	1.0	1.0
[PRAVEEN et al., 2020]	Blockchain for 5G: A Prelude to Future Telecommunication			1.0	1.0	0.5	0.5	0.5	1.0
[AKHTER et al., 2021]	A secured privacy-preserving multi-level blockchain framework for cluster based VANET	1.0	0.5	1.0	1.0	1.0	1.0	1.0	1.0
[Jameel et al., 2020]	Efficient Mining Cluster Selection for Blockchain-Based Cellular V2X Communications	1.0	0.5	1.0	1.0	1.0	1.0	0.5	0.5
[NOOR et al., 2020]	A review on communications perspective of flying AD-HOC networks: Key enabling wireless technologies, applications, challenges and open research topics	0.5	1.0	1.0	1.0	1.0	1.0	1.0	
[GUPTA et al., 2020]	VAHAK: A blockchain-based outdoor delivery scheme using UAV for healthcare 4.0 services	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
[SHRESTHA et al., 2020]	Evolution of v2x communication and integration of blockchain for security enhancements	0.5	0.5	1.0	1.0	1.0	1.0	1.0	1.0
[Aggarwal; Kumar; Tanwar, 2021]	Blockchain-Envisioned UAV Communication Using 6G Networks: Open Issues, Use Cases, and Future Directions	0.5	1.0	1.0	1.0	1.0	0.5	0.5	1.0
[Li et al., 2020]	Blockchain-Based Data Security for Artificial Intelligence Applications in 6G Networks		0.5	1.0	1.0	1.0	1.0	1.0	1.0
[RAHMAN et al., 2020]	Lightweight blockchain assisted secure routing of swarm UAS networking	1.0	1.0	1.0	1.0	1.0	1.0	1.0	
[GUPTA et al., 2021a]	Blockchain-Envisioned Softwarized Multi-Swarming UAVs to Tackle COVID-19 Situations	0.5		1.0	1.0	1.0	1.0	0.5	1.0
[MOHAMMED et al., 2020]	Deep Reinforcement Learning for Computation Offloading and Resource Allocation in Blockchain-Based Multi-UAV-Enabled Mobile Edge Computing	0.5	1.0	1.0	1.0	1.0			1.0
[WANG et al., 2020]	Secure Crowdsensing in 5G Internet of Vehicles: When Deep Reinforcement Learning Meets Blockchain	0.5	0.5	1.0	1.0	1.0	1.0	1.0	1.0
[KUMARI et al., 2020]	A taxonomy of blockchain-enabled softwarization for secure UAV network	0.5		1.0	1.0	1.0	1.0	1.0	1.0
[RAHMADIKA; LEE; RHEE, 2019]	Blockchain-Enabled 5G Autonomous Vehicular Networks	1.0	0.5	1.0	0.5	1.0	0.5	0.5	1.0
[ARIF et al., 2020]	Integration of 5G, VANETs and blockchain technology	1.0	1.0	1.0	1.0	1.0	1.0	0.5	1.0
[WANG et al., 2020]	B-TSCA: Blockchain assisted Trustworthiness Scalable Computation for V2I Authentication in VANETs	1.0	0.5	1.0	1.0	1.0	1.0	1.0	1.0
[RIDHAWI et al., 2021]	Enabling Intelligent IoCV Services at the Edge for 5G Networks and Beyond	0.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
[MEHTA; GUPTA; TANWAR, 2020]	Blockchain envisioned UAV networks: Challenges, solutions, and comparisons	0.5	0.5	1.0	1.0	1.0	1.0	1.0	1.0

exemplifica possíveis ataques que poderiam ser evitados com os recursos de BC (QP-01), como evitar adulterações na imutabilidade das informações, rastreabilidade e transparência para evitar fraudes, e confiabilidade e tolerância a falhas para garantir a disponibilidade. No entanto, este trabalho não demonstra uma comunicação V2V (*Vehicle-to-Vehicle*) segura para lidar com congestionamentos de tráfego de forma autônoma e inteligente (QP-05); também não se preocupa com o custo energético da arquitetura já que a solução proposta utiliza alto poder computacional (QP-06).

A solução proposta do trabalho elaborado por [Jameel et al., 2020] leva em consideração o custo que o descarregamento de tarefas de mineração poderia ser em redes celulares V2X (QP-06). Por esse motivo, eles apresentaram uma abordagem para equilibrar a carga computacional no *cluster* de mineração com base em preço e recursos com uma estratégia de teoria dos jogos. O preço do *cluster* de mineração é determinado e atribuído e, em seguida, o veículo de descarga se vincula à manutenção do *cluster*, garantindo também a equidade entre os veículos concorrentes. Aspectos de segurança não foram profundamente levados em consideração, assim como a escalabilidade da rede BC. Além disso, esta solução não leva em consideração cenários onde os recursos computacionais são distribuídos de forma não uniforme (QP-05).

[GUPTA et al., 2020] apresenta um esquema detalhado para implementar uma rede UAV segura baseada em Ethereum (QP-02) para transporte de suprimentos médicos de saúde. Este trabalho sugere o uso do protocolo IPFS (*InterPlanetary Filesystem*) para mitigar os problemas de custos de armazenamento relacionados ao *framework* BC (QP-06), e apresenta uma análise abrangente do desempenho da solução proposta (QP-04).

[RAHMADIKA; LEE; RHEE, 2019] aborda informações essenciais do AVN (*Avian Network*) descentralizado habilitado para 5G e como o BC pode conceder segurança por sua natureza (QP-01). Este artigo também apresenta um protocolo para métodos de autenticação secundária em redes 5G (QP-05). Ele também enumera desafios futuros, como a escolha de um mecanismo de consenso adequado para evitar atrasos e garantir BFT (*Byzantine Fault Tolerance*).

[WANG et al., 2021b] apresenta um algoritmo de roteamento baseado em BC leve para rede celular UAS de enxame (QP-01) para mitigar UASs maliciosos, usando o mecanismo de consenso PoT (*Proof-of-Traffic*) (QP-02) para reduzir o consumo de recursos (QP-06) e comunicação rápida (QP-03, QP-04), demonstrando em detalhes a avaliação dos resultados, comparando as diferentes estratégias de consenso.

A pesquisa apresentada por [GUPTA et al., 2021a] foca em uma arquitetura de cinco camadas baseada em um SDN (*Software-Defined Networking*) em uma URLLC 6G (QP-03, QP-05) para gerenciar UAVs multi-swarm, usando BC para garantir a segurança (QP-01) como contrapartida de ameaças usuais inerentes à natureza aberta dos canais de comunicação propostos (QP-04). Apresenta conceitos importantes de cada tecnolo-

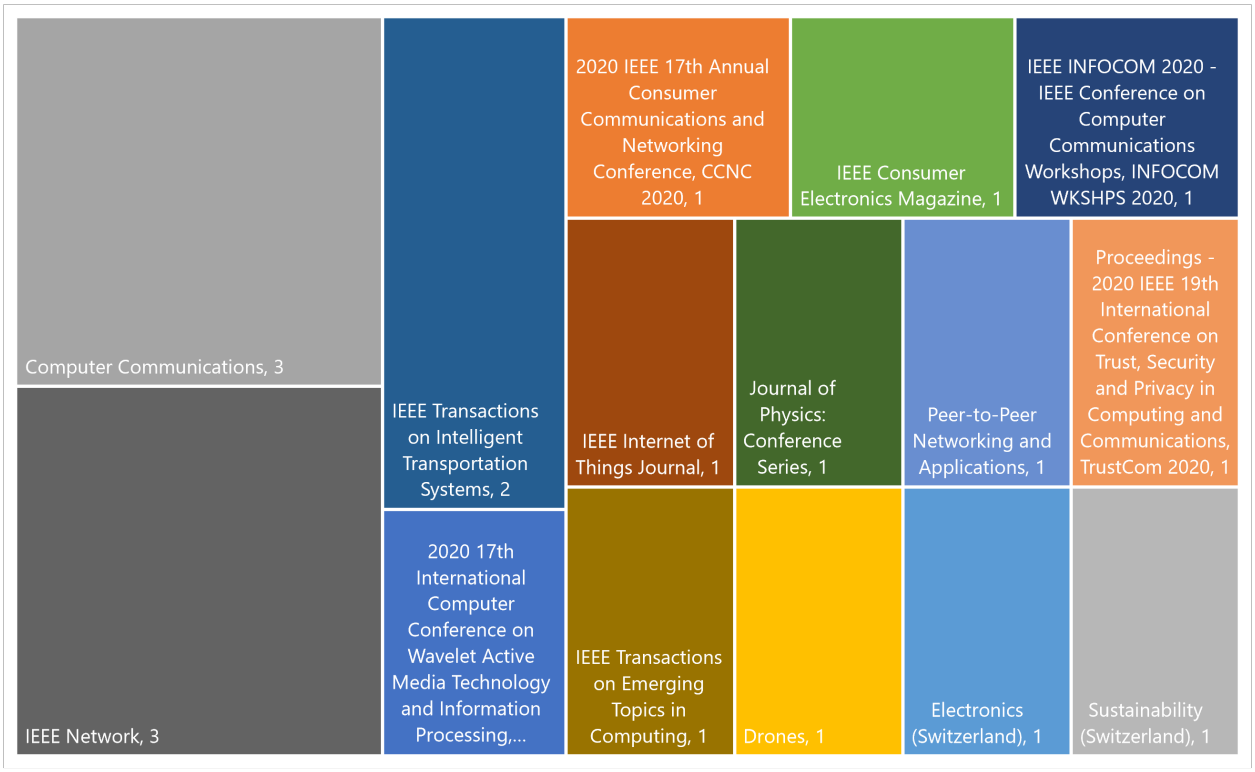


Figura 6 – Artigos do Jornal

gia envolvida, descreve uma lista detalhada de falhas de segurança e como o BC pode beneficiar a comunicação segura nesta arquitetura. Esses trabalhos deixam em aberto questões relacionadas à gestão de energia (QP-06), lacunas de segurança e privacidade, entre outras.

O artigo escrito por [WANG et al., 2020] concentra-se em uma solução segura (QP-01) e de baixa latência (QP-04) com *clusters* BCs baseados em localização para detecção de multidões para redes veiculares 5G (QP-03). O DRL (*Deep reinforcement learning*) é aplicado para lidar com a seleção e transações de mineração (QP-05), com detalhes de avaliação e simulação de desempenho (QP-06).

O trabalho do [AKHTER et al., 2021] propõe um esquema de autenticação habilitado para BC seguro multinível para *clusters* VANETs (QP-01), com uma estrutura de centros de autenticação global e centros de autenticação locais. Também sugere uma modificação dos protocolos MAC (*Medium Access Control*) tradicionais estabelecidos no formato de pacote de controle do IEEE 802.11. Uma PoC (*Proof-of-Concept*) detalhada é descrita por [AKHTER et al., 2021] e também se preocupa com a criptografia de mensagens (QP-05), estratégia de eleição de chefe de *cluster*, mensagem crítica e priorização de veículos de emergência (QP-04).

RSUs (*Roadside Units*) desempenham um papel importante como terminais confiáveis no esquema B-TSCA (*Blockchain Assisted Trustworthiness Scalable Computation for V2I Authentication*) apresentado pelo [WANG et al., 2020] (QP-03), como minera-

dores no processo de autenticação V2I (*Vehicle-to-Interfaces*) usando o mecanismo de consenso BC (QP-01). De acordo com o estudo, a proposta supera o problema com atraso de entrega V2I (QP-04) e garante segurança e confiabilidade na VANET contra agentes de ameaças. Além disso, os autores explicam os modelos de segurança e de sistema que compõem o esquema, fornecendo dados detalhados para sustentar suas pesquisas.

Para as lacunas de segurança derivadas das técnicas de *softwarização* trazidas pelo 5G em redes UAV (QP-03), [KUMARI et al., 2020] contribuiu com seu levantamento sistemático baseado em BC para conceder segurança (QP-01) nestes tipos de redes. Eles trouxeram pesquisas anteriores para proteger SDN e NFV (*Network Functions Virtualization*) e respectivas lacunas, enumerar vários tipos de ataques e trazer conceitos centrais de *softwarização* de rede. Em seguida, este trabalho explora a integração do BC, para finalmente apresentar uma arquitetura de software UAV baseada em blockchain com o *Ethereum* (QP-02), e deixa pontos em aberto e desafios futuros (QP-05).

O trabalho realizado por [RIDHAWI et al., 2021] traz conceitos importantes de *clusters* baseados em serviços, e propõe uma arquitetura descentralizada multicamada baseada em *cluster*, com agentes BC colaborativos (QP-01) e aprendizado por reforço (QP-05) para melhorar a QoS (*Quality of Service*) (QP-04). Abrange também trabalhos anteriores, implementação, configuração e avaliação de desempenho resultante da simulação.

O escopo da pesquisa [MEHTA; GUPTA; TANWAR, 2020] é fornecer uma pesquisa abrangente com base em extensa literatura, demonstrando os problemas de segurança mais comuns com redes 5G e UAV (QP-05) e como o BC pode ser integrado aos UAVs para garantir confiabilidade, disponibilidade e imutabilidade , e transparência (QP-01) em uma solução descentralizada de baixa latência (QP-04). O trabalho ilustra a aplicação com um estudo de caso.

[Aggarwal; Kumar; Tanwar, 2021] estende o trabalho anterior feito em [MEHTA; GUPTA; TANWAR, 2020], trazendo os rumos futuros da comunicação baseada em BC em UAVs (QP-01) e novidades da tecnologia 6G (QP-05), comparando as limitações do 5G (QP-03). Os autores demonstram o papel fundamental do BC em redes 6G em diversas aplicações em sua arquitetura multicamadas e os pontos em aberto que ele traz.

Como um facilitador chave para aplicações de inteligência artificial distribuída (QP-05), as redes 6G poderiam alavancar seu esquema de segurança com a integração do BC (QP-01), de acordo com as premissas trazidas pelo [Li et al., 2020]. Este trabalho apresenta uma arquitetura potencial de uma rede 6G (QP-03), e como as aplicações de IA podem ser ameaçadas sem a adoção de um esquema de segurança habilitado para BC. Os trabalhadores do BC estão sujeitos a um sistema de reputação para diferenciar os atacantes. O trabalho deixa em aberto questões relacionadas à privacidade no final.

[SHRESTHA et al., 2020] explica o impacto das tecnologias 5G e NR (*New Radio*), MEC (*Mobile Edge Computing*) na evolução V2X (QP-03) baseada em celular e DSRC (*Dedicated short-range communications*), e os resultados dessa integração e como ela abre a segurança e questões de privacidade que podem ser derrubadas pela adoção do BC nesta arquitetura (QP-01). A pesquisa fornece uma pesquisa detalhada e uma revisão sobre os tópicos mencionados, como o BC pode potencialmente conceder segurança e privacidade e enumera os futuros desafios e oportunidades neste campo (QP-05).

3.6 Definição de escopo

Com base nos resultados obtidos da revisão sistemática de literatura, foi identificada a carência de trabalhos que direcionassem os esforços em analisar a aplicabilidade de algoritmos baseados em consenso em redes distribuídas que possuíssem baixa capacidade computacional e recursos limitados.

Diante disso, foi definido o escopo para contribuição com um projeto de simulação de uma rede computacional com baixo poder computacional, que pudesse fazer uso de técnicas derivadas de algoritmos de consenso, para se comunicarem de forma segura e descentralizada.

A partir deste ponto, uma nova pesquisa foi conduzida para localizar trabalhos relacionados mais especificamente voltados a veículos aéreos não tripulados, por se tratarem de dispositivos que podem possuir escassez de recursos computacionais e que precisam economizar no processamento de dados tendo em vista a limitação na sua fonte de energia.

No contexto de UAVs e a aplicação de algoritmos de consenso no seu processo de comunicação, foram encontradas as seguintes contribuições bibliográficas:

Para as lacunas de segurança derivadas das técnicas de softwarização trazidas pelo 5G em redes UAV, [KUMARI et al., 2020] contribuíram com seu levantamento sistemático baseado em BC para garantir a segurança nestes tipos de redes. Eles resumiram pesquisas anteriores sobre Rede Definida por Software (SDN) e Virtualização de Funções de Rede (NFV), bem como os diferentes tipos de ataques no contexto de redes UAV. Em seguida, eles exploram a integração do BC para finalmente apresentar uma arquitetura de software de UAV baseada em *blockchain* com o *Ethereum*.

Vários algoritmos de consenso podem ser usados para evitar violações de segurança em sistemas de aeronaves não tripuladas (UAS). Nesse contexto, os algoritmos também precisam estar cientes do consumo de energia dos UAS. Um desses algoritmos é o *Proof-of-Traffic* [WANG et al., 2021b], que se concentra na prevenção de violações e na redução do consumo de energia no processo de roteamento aprimorado da rede.

As FANETs são construídas com diferentes tecnologias, como comunicações sem fio

(LPWAN (*Low Power Wide Area Network*), *Bluetooth*, *ZigBee*, etc.) para aplicações específicas (agricultura, transporte, emergência, etc.). Dependendo da arquitetura, há desafios (como segurança e custos de energia) a serem considerados e tópicos em aberto, como o uso de BC, aprendizado de reforço profundo e virtualização de recursos de UAV [NOOR et al., 2020]. Como ilustração, a pesquisa apresentada por [MEHTA; GUPTA; TANWAR, 2020] foca em uma arquitetura de cinco camadas baseada em um sistema SDN em uma rede URLLC 6G para gerenciar UAVs *multi-swarm*, usando BC para garantir a segurança como contrapartida das habituais ameaças à natureza aberta dos canais de comunicação propostos. Apresenta conceitos importantes de cada tecnologia envolvida, descreve uma lista detalhada de falhas de segurança e como o BC pode se beneficiar da comunicação segura nesta arquitetura. Este trabalho deixa em aberto questões relacionadas à gestão de energia, segurança e lacunas de privacidade, entre outras.

3.7 Descrição do problema

Dito isso, embora alguns dos trabalhos mencionados acima tenham medido o desempenho da rede, não há dados explícitos sobre o consumo de CPU ou memória dos UAVs individuais.

As FANETs são redes de drones autônomos que trabalham em conjunto para cumprir missões específicas. A duração da operação dos drones é limitada pela autonomia energética, que é determinada pelo tipo de missão a ser executada e pelos recursos computacionais dos drones.

A capacidade energética das FANETs pode variar de acordo com a tarefa a ser cumprida. Em alguns casos, a rede de drones deve ter autonomia suficiente para completar a missão, enquanto em outras, o tempo de operação não é tão crítico.

O consumo de energia dos drones é diretamente influenciado pelos recursos computacionais necessários para executar a tarefa. A utilização de algoritmos de segurança na comunicação em sistemas computacionais pode afetar o consumo de energia desses sistemas. Isso se deve à necessidade de realizar operações intensivas de computação, como cálculos matemáticos e transformações, exigidas pelos algoritmos de segurança. Essas operações consomem poder de processamento e, consequentemente, energia. Além disso, o uso de chaves de criptografia mais longas, para aumentar a segurança, também pode aumentar o consumo de energia. Isso porque tarefas mais complexas exigem mais processamento de dados, o que acarreta maior consumo de energia.

É importante que se leve em consideração a autonomia energética das FANETs antes de implantá-las em uma missão, a fim de garantir que a rede de drones seja capaz de cumprir sua tarefa sem falhar devido a esgotamento de energia.

Tal condição conduz ao seguinte enunciado do problema: É possível usar algoritmos de consenso em redes com baixo poder computacional e recursos limitados para garantir uma arquitetura de comunicação resiliente e segura?

Para tanto, o presente trabalho propôs implementar a simulação de uma FANET utilizando blockchain e composta por drones de baixo recurso computacional para completar uma trajetória. Durante o curso da trajetória proposta, alguns agentes maliciosos tentarão alterar as coordenadas do waypoint do enxame. Para mostrar os benefícios da abordagem, os UAVs usando uma interface REST desprotegida irão realizar a mesma trajetória e enfrentar a mesma interferência de agentes maliciosos.

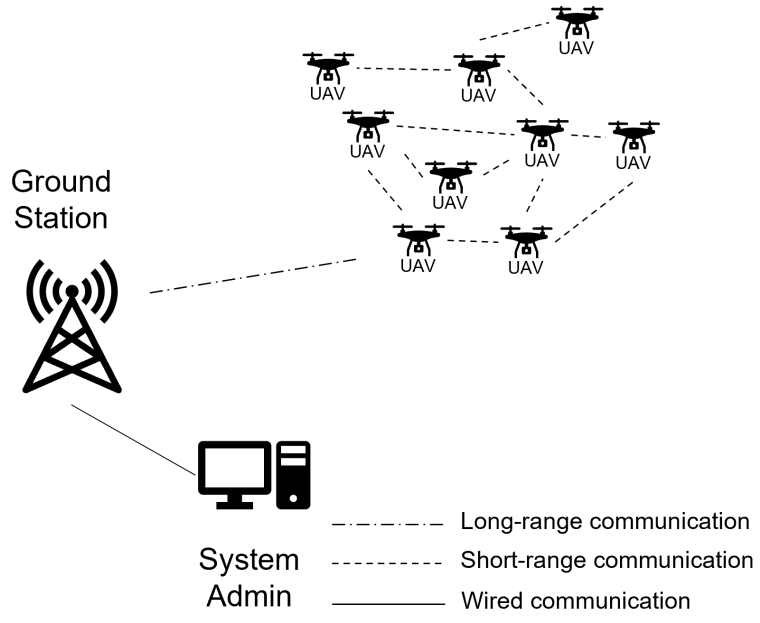
Após a conclusão da simulação, os resultados de uso da CPU e memória serão comparados, bem como se ambas as FANETs conseguiram completar sua missão com sucesso.

4 Definição da arquitetura e implementação do Framework

A arquitetura das FANETs proposta para ser explorada no presente trabalho é mostrada na Figura 7. Essa arquitetura é comum na literatura [WANG et al., 2021a; HU et al., 2021; KUMARI et al., 2020] e é composta pelos seguintes elementos:

- **UAVs:** os veículos podem ser operados manualmente por pilotos ou controlados de forma autônoma [MEHTA; GUPTA; TANWAR, 2020]. Eles são compostos por propulsores, sensores, um módulo de energia, um sistema de controle, um sistema de comunicação sem fio e um sistema de geoposicionamento. Na arquitetura proposta, os UAVs se comunicam com as estações de controle terrestres por meio de um sistema de comunicação sem fio. Eles também podem compartilhar dados de posicionamento e localização com outros UAVs usando os mesmos sistemas de comunicação sem fio.
- **Estação de controle terrestre:** a estação terrestre, estação base ou estação de controle terrestre (GCS), é a estação responsável pela operação dos UAVs. O controle de missão é definido por um operador que usa um sistema de administração para determinar as diretrizes de comando dos FANETs [WANG et al., 2021a]. O foco neste trabalho está limitado aos comandos de posicionamento e localização. No entanto, outros tipos de comandos podem ser facilmente implementados com base na arquitetura proposta.

Figura 7 – Arquitetura proposta de simulação



- **Sistemas de administração:** estes sistemas são responsáveis por gerar os comandos de controle para os UAVs. No presente trabalho, o sistema de administração é a entidade responsável por enviar quaisquer alterações de navegação coordenada de FANETs para os UAVs.
- **Comunicação sem fio:** nesta arquitetura, a comunicação sem fio permite a comunicação entre UAVs. Considerando que a plataforma utilizada para simular os cenários propostos se limita a Rede Sem Fio Definida por Software, ou SDWN (*Software-defined wireless network*), e que o padrão IEEE 802.11 é amplamente adotado [FONTES; ROTHENBERG, 2019; NOOR et al., 2020] em adaptadores sem fio, a tecnologia adotada será *WiFi* (802.11 Series).
- **Comunicação cabeada:** na arquitetura proposta, a comunicação entre o sistema da administração e a estação base será realizada através de uma rede cabeada.

Além da simulação para investigar a aplicabilidade de algoritmos de consenso de baixo custo em FANETs de recursos limitados, este trabalho pretende contribuir permitindo um projeto que pode ser extensível à comunidade científica para realizar experimentos relacionados em um *testbed*, baseado em soluções de código aberto, disponível em **https://github.com/danielpfernandes/containernet**.

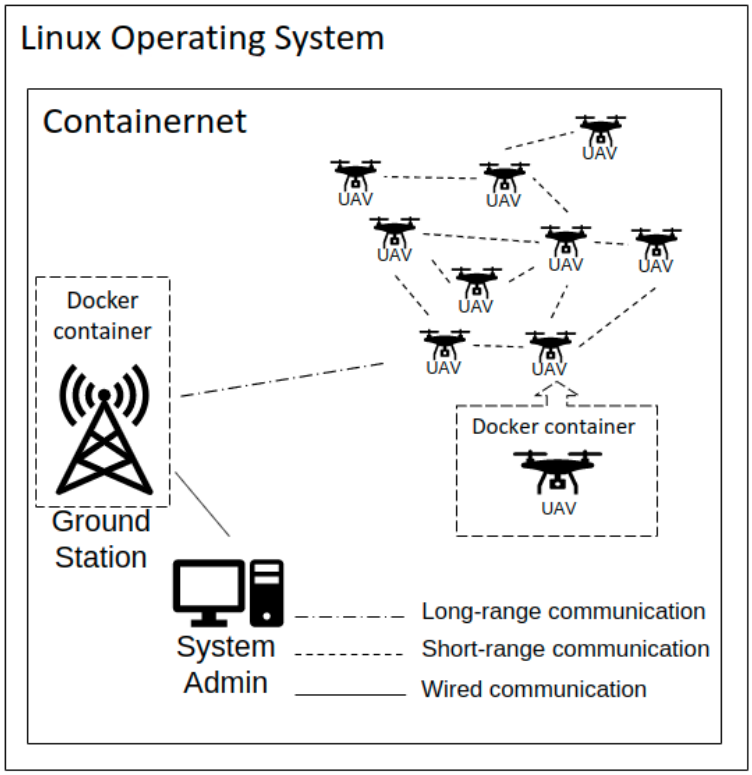
Assim, para alcançar a implementação da arquitetura descrita no presente Capítulo, diversas ferramentas foram utilizadas de acordo com as características presentes nas FANETs.

4.1 *Containernet-Wifi*

Mininet é uma das ferramentas mais utilizadas para emular redes virtuais em um único computador. É fácil de configurar, possui uma comunidade ativa para manter e desenvolver recursos e é lançado sob licença BSD (*Berkeley Software Distribution*) *Open Source* [LANTZ; HELLER; MCKEOWN, 2010]. Vários ramificações do *Mininet* surgiram ao longo do tempo, incluindo o *Mininet-Wifi* [FONTES et al., 2015][FONTES; ROTHENBERG, 2019] que visa emular cenários de comunicação reais em várias arquiteturas de rede, como comunicação ad-hoc usando o padrão de rede IEEE 801.22bgn.

Outra derivação do *Mininet* [PEUSTER; KARL; ROSSEM, 2016] é o *Containernet* [FONTES; ROTHENBERG, 2019], que permite a criação de instâncias de nós no *Mininet* por meio de um contêiner *Docker*, auxiliando na emulação de *hardware* e na utilização de recursos. *Docker* é uma ferramenta que permite a criação e gerenciamento de contêineres isolados comumente usados na construção e implantação de aplicativos. Cada contêiner é uma unidade virtualizada com recursos próprios que, no caso do *Containernet*, pode representar um dispositivo pertencente à rede emulada.

Figura 8 – Arquitetura traduzida para o ambiente simulado



A plataforma escolhida para emular a prova de conceito sugerida foi o *Containernet-Wifi*, que combina os benefícios do *Mininet-Wifi* e do *Containernet*.

A arquitetura proposta pela Figura 7 pode ser traduzida para o esquema apresentado pela Figura 8, onde cada drone e a estação base são representados por nós distribuídos numa rede de contêineres.

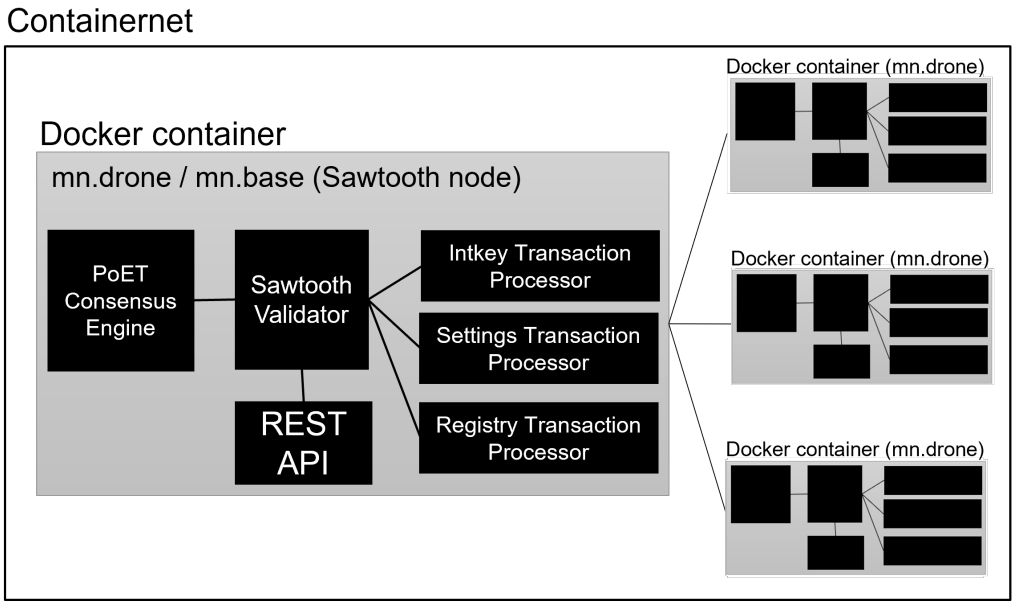
4.2 Hyperledger Sawtooth com algoritmo de consenso PoET

Mantido pela *Linux Foundation*, o *Hyperledger Sawtooth* é uma plataforma modular de código aberto que fornece uma arquitetura *blockchain* usando algoritmos de consenso como tolerância a falhas bizantinas práticas (PBFT) e prova de tempo decorrido (PoET).

A escolha desta plataforma considerou sua flexibilidade, o grau de engajamento da comunidade que mantém a solução, a possibilidade de expansão da solução no processo de comunicação entre componentes *on-chain* e *off-chain*, extensibilidade para redes privadas e públicas, e a possibilidade de realizando a verificação paralela de transações.

O *Hyperledger Sawtooth* também fornece PoET SGX (*Software Guard Extensions*), mas a solução está limitada a dispositivos equipados com a tecnologia Intel ® SGX, o que está fora do escopo deste trabalho, pois o objetivo principal é simular o uso da BC em

Figura 9 – Estrutura de drone simulada



peers com restrições em recursos e consumo de energia.

Além disso, atualmente, o *Sawtooth* PBFT [SEELEY, 2019] apresenta algumas limitações, como número mínimo de *peers* conectados e falta de suporte para redes dinâmicas. Por esse motivo, o algoritmo de consenso escolhido foi o algoritmo *Proof of Elapsed Time* (PoET).

Ele também permite uma ampla população de pares que podem suportar um enxame de FANET, e o custo do processo de eleição é mais barato do que algoritmos de consenso CFT usuais, como *Proof-of-Work*.

A estrutura de um drone habilitado com o *Hyperledger Sawtooth* pode ser observada na Figura 9, onde cada contêiner Docker representa um drone (*Sawtooth node*), que é composto por pelo mecanismo de consenso, um sistema validador e um conjunto de processadores de transação, além da API REST para comunicação externa.

Um componente importante para o presente trabalho é o *Intkey Transaction Processor*, pois ele é utilizado para simular a troca de informações sobre as coordenadas dos drones. O *Intkey Transaction Processor* trabalha processando as transações enviadas pelo validador e atualizando o estado do bloco caso a validação seja bem-sucedida.

4.3 Esquema de comunicação e monitoramento do sistema

Os registros de atividade e desempenho de cada um dos drones e estações base emulados foram coletados por meio do *Google cAdvisor* e *Prometheus* e depois traduzidos para gráficos com o *Grafana*.

Na simulação proposta, foram propostos dois cenários. No primeiro cenário, a transmissão das coordenadas é realizada em uma rede desprotegida usando interfaces REST. As coordenadas de destino são propagadas entre os drones ao longo de cinco etapas, sendo validadas por cada drone com a estação base para verificar sua legitimidade.

Já no segundo cenário, a FANET está conectada através do *Hyperledger Sawtooth* para simular a confiabilidade da transmissão de informações entre os drones e a estação base, mesmo quando um nó validador perde a comunicação com os demais membros da rede.

A propagação de coordenadas no cenário desprotegido foi feita usando interfaces REST para simular a propagação de coordenadas de destino entre os drones. Um *script* escrito em *Python* foi executado para transmitir as informações recebidas pelos drones na FANET através de uma interface REST desprotegida (ver Algoritmo 1).

Para simular este esquema REST desprotegido, a FANET valida o comando recebido para alterar as coordenadas de destino verificando com a BC se a ordem é legítima.

A GCS, por sua vez, atua como entidade centralizadora para validar se as coordenadas foram de fato enviado por elas. Antes de propagar as coordenadas entre si na FANET, os drones esperam a resposta da GCS da atual coordenada de destino. Se dados informados pela GCS não correspondem à ordem recebida, os drones da FANET não propagam a informação e os dados da missão não são alterados.

Algorithm 1 Algoritmo REST desprotegido para definir as coordenadas

1: $payload \leftarrow endereço, latitude, longitude, altitude$
2: **if** *propagate* é verdade **then**
3: $\quad url \leftarrow drone_{(REST)} : 5000/propagate$
4: **else**
5: $\quad url \leftarrow localhost_{(REST)} : 5000/locations$
6: **end if**
7: $arquivodevalidação \leftarrow payload$
8: $resultado \leftarrow post(url, payload, httpheaders)$

Contudo, como uma FANET precisa de um grau de autonomia para operar, mesmo sem uma entidade centralizadora, casos críticos precisam ser atendidos, como alteração de coordenadas para evitar colisões ou mudança de rota por impossibilidade de alcançar o destino final. Diante disso, e para fins ilustrativos, foi propositalmente permitido que na presente simulação uma FANET pudesse aceitar o comando de alteração de rota de um agente que pertença a FANET ou uma GCS autorizada a se comunicar com aquela rede.

Nessas circunstâncias, um drone que pertence à FANET ou uma GCS que estão comprometidos por ações maliciosas podem realizar alterações não esperadas.

A presente simulação reproduz uma topologia em que os drones usam uma interface REST simples apenas com o objetivo de comparar com os casos de teste do *Hyperledger*

Sawtooth. Em um cenário real, sistemas distribuídos aplicam metodologia de autenticação e autorização como camadas de segurança para garantir um canal confiável através dessas interfaces, o que significa que mais recursos computacionais devem ser levados em consideração.

5 Simulação dos Cenários

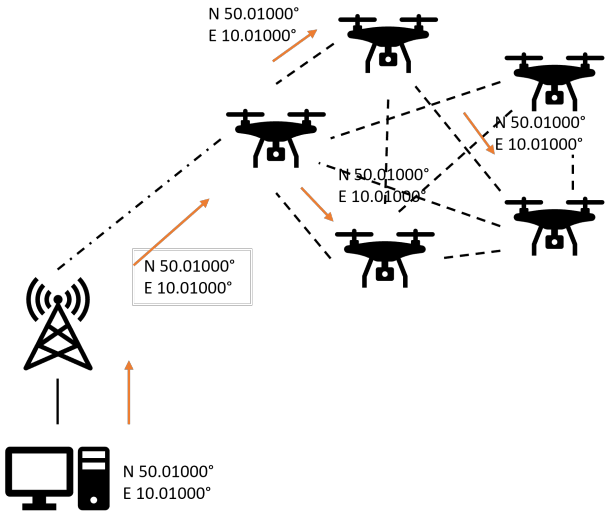
O objetivo da simulação é reproduzir a trajetória de um enxame FANET, que é composto por cinco drones e uma estação terrestre, para demonstrar como cada dispositivo interage com os outros através da troca de pacotes de mensagens, com a missão específica de alcançar um determinado destinado estabelecido por uma coordenada composta com os dados de latitude, longitude e altitude (Figura 10).

Durante a simulação, cada drone é capaz de enviar e receber informações entre a estação terrestre e outros drones no enxame. A estação terrestre (GCS), por sua vez, pode enviar comandos e atualizações para os drones e receber informações dos drones no enxame. Pela da troca de informações em tempo real, os drones são capazes de se adaptar rapidamente a mudanças no ambiente e trabalhar juntos para cumprir a missão com sucesso.

A simulação inclui a criação de um objetivo de missão e o estabelecimento de uma trajetória a seguir. Como a proposta do trabalho é analisar a eficácia da incorporação da tecnologia de *blockchain* na transmissão de dados em drones com baixa capacidade computacional e recursos limitados e fornecer um *framework* de simulação para comunidade cientifica em trabalhos futuros, não estão sendo considerados detalhes tais como coleta das informações de sensores, monitoramento da posição dos outros drones no enxame para evitar colisões e garantir a segurança de toda a operação.

Ao demonstrar a eficácia da validação de informações entre os dispositivos em um enxame FANET por meio da blockchain, este trabalho contribui para o desenvolvimento da segurança tecnologias autônomas em diferentes áreas, como transporte, agricultura,

Figura 10 – Fluxo de comunicação proposto pela simulação



logística e muitas outras.

Conforme descrito na Seção 4.1, *Mininet* é um *framework* popularmente utilizado para simulação de redes em diferentes casos de uso e em variadas topologias. Esta ferramenta e sua variante *Mininet-Wifi* ([FONTES et al., 2015]) tem sido utilizada em diversos estudos sobre redes veiculares, como por exemplo os trabalhos de [SALEH; FATHY, 2023] e [KALININ; KRUNDYSHEV; SEMIANOV, 2017].

Considerando o propósito do presente trabalho, era importante que se pudesse gerenciar e monitorar os recursos computacionais de cada drone simulado. Diante disso, a variante do *Mininet-Wifi* conhecida como *Containernet* foi elegida como plataforma para a simulação, pois é possível executar cada nó de uma rede simulada como se fosse um contêiner do *Docker*.

Foi-se criado uma ramificação do projeto *Containernet* no GitHub, onde foi possível contribuir com projeto original, trazendo as funcionalidades de simulação de FANETs capazes de se comunicar incorporando um esquema de comunicação baseado em *blockchain*.

Como resultado deste trabalho, foram introduzidas 11.760 linhas de código no projeto original em 52 novos arquivos. A maioria das modificações foi realizada seção de exemplos do projeto, localizada no diretório *"examples/"*.

A estrutura do presente trabalho pode ser identificada principalmente nos seguintes diretórios e/ou arquivos:

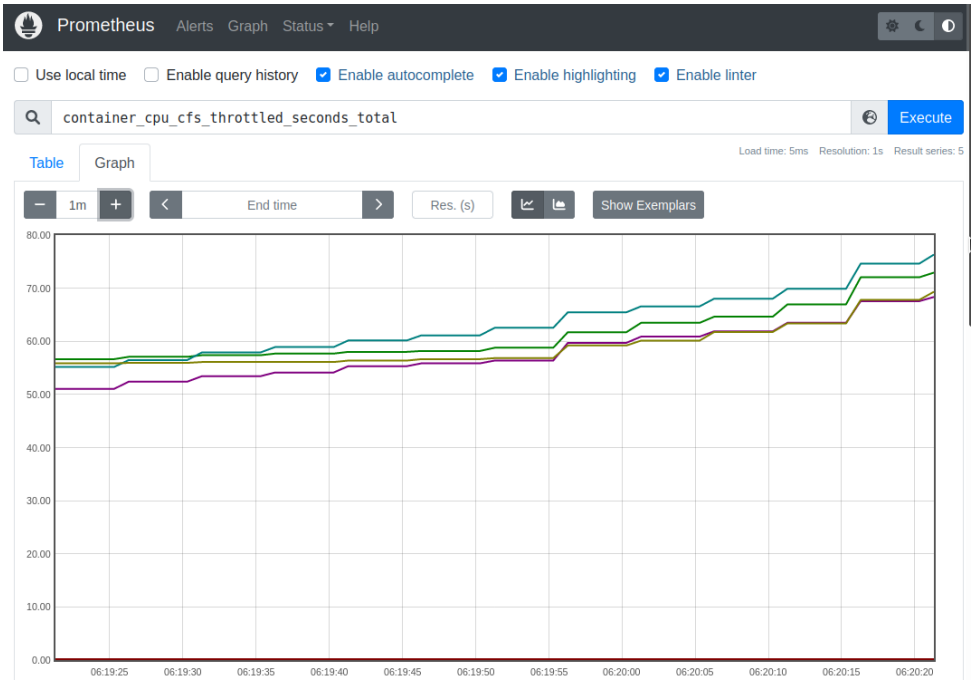
- *examples/example-containers/rest_scripts*: Contém os *scripts* utilizados por cada drone para simular a transmissão de dados usando uma interface REST.
- *examples/example-containers/sawtooth_scripts*: Contém os *scripts* dos drones para realizar a transmissão das informações de trajetórias de destino no cenário onde se utiliza o *framework Hyperledger Sawtooth* para comunicação através de *blockchain*.
- *examples/example-containers/sawtoothAll.Dockerfile*: Arquivo que contém as informações necessárias para criação da imagem do *Docker* utilizado para inicializar cada drone e estação base da FANET.
- *examples/fanet-sawtooth*: Contém o código-fonte para simulação dos cenários e seus respectivos estágios durante o trajeto dos drones.
- *results*: Local de armazenamento dos resultados no caso de uma execução bem-sucedida da simulação.

Durante esta rota, em estágios distintos são executados alguns tipos de modificações não autorizadas na interface REST dos drones (vide Seção 4.3) com o objetivo de alterar o caminho da FANET.

Foi aplicado um algoritmo de consenso (PoET) (conforme explicado na Seção 4.2) para garantir a integridade e disponibilidade dos dados relacionados à execução da trajetória até o destino. O Containernet-Wifi *framework* [PEUSTER; KARL; ROSSEM, 2016] permitiu a emulação de topologia de rede.

Durante a execução dos cenários, o sistema foi monitorado utilizando as ferramentas de monitoramento e observabilidade mencionadas na Seção 4.3. O *Prometheus* (Figura 11) associado ao *cAdvisor* (Figura 12) capturaram a utilização de recursos de processador e memória, leitura e escrita de disco de cada drone, enquanto o *Grafana* reinterpretava os dados para gerar o painel com visualização de gráficos mais compreensíveis (Figura 13).

Figura 11 – Exemplo da interface gráfica do *Prometheus* capturando a execução da simulação



5.1 Especificação de hardware dos drones

A simulação foi realizada em um computador equipado com processador Intel® Core™ i7-4720HQ CPU @ 2,60GHz 2,59GHz, com 16,0 GB e com sistema operacional Ubuntu 18.04.6 virtualizado através do *Hyper-V* utilizando 6,0 GB de memória dinâmica compartilhada e quatro processadores virtuais.

Com o *Containernet* foi possível definir os recursos de processamento e memória de cada drone no código-fonte da plataforma. Como, por exemplo o trecho do código 5.1:

Figura 12 – Exemplo da interface gráfica do *cAdvisor* capturando a execução da simulação

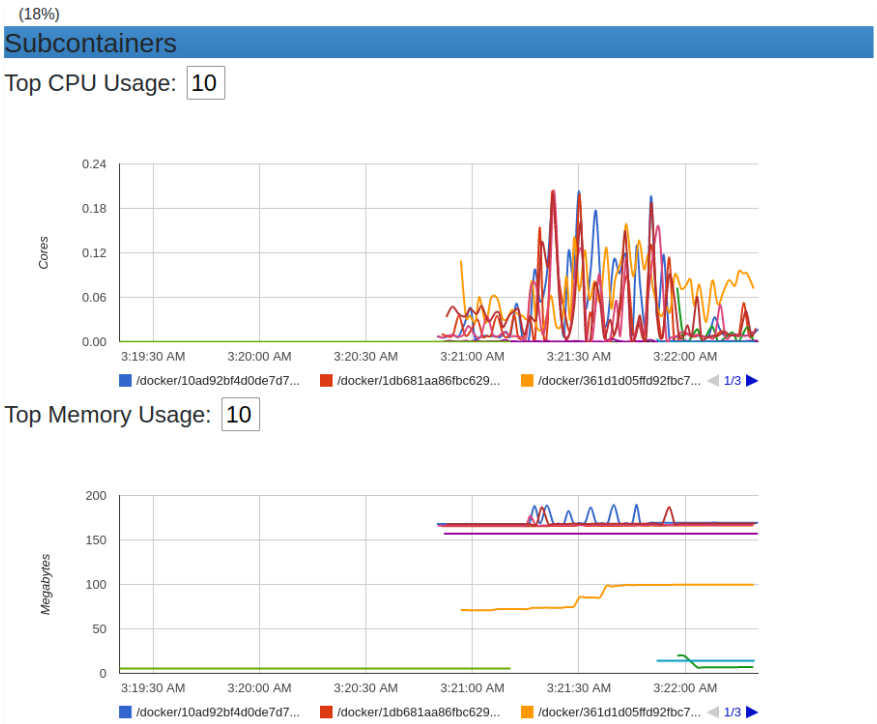
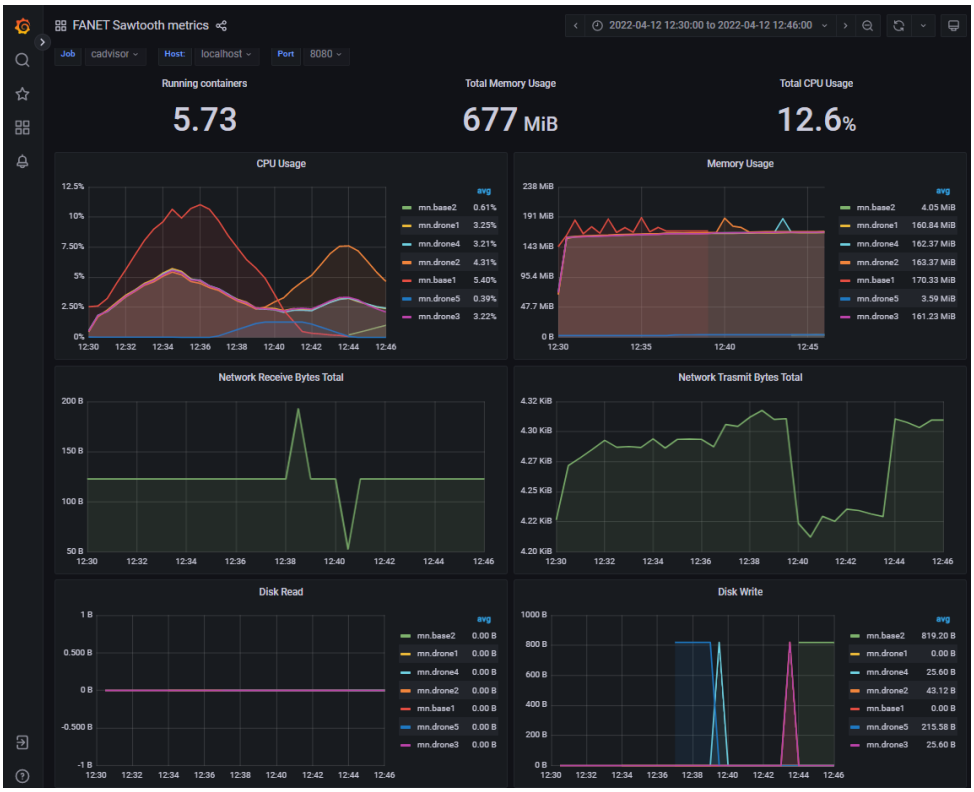


Figura 13 – Painel do *Grafana* com os resultados de uma simulação bem-sucedida



Listing 5.1 – Exemplo de código-fonte com a configuração do drone

```
drone = net.addStation(name,
                       ip=my_ip,
```

```
mac=my_mac,
cls=DockerSta,
dimage=docker_image,
ports=ports,
volumes=["/tmp/" + name + "/data:/data"],
mem_limit=900182016,
cpu_shares=2,
cpu_period=50000,
cpu_quota=10000,
position='30,60,10')
```

Em relação à conectividade cada drone possui um adaptador de rede sem fio, conectado em modo ad-hoc de malha aberta através do canal 5, usando o protocolo de roteamento avançado B.A.T.M.A.N (*Better Approach To Mobile Ad-hoc Networking*) operando na camada 2 [LANG, 2007]. A configuração dos adaptadores foi realizada como no seguinte trecho de código-fonte 5.2:

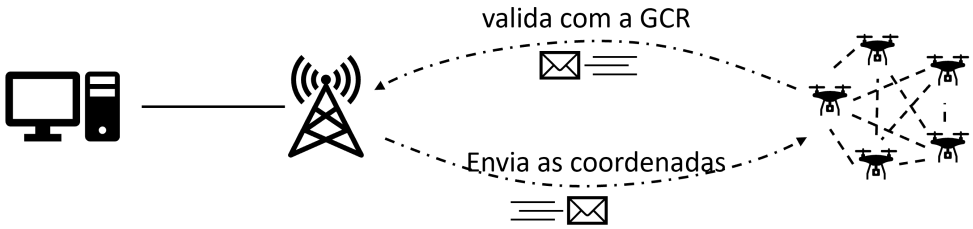
Listing 5.2 – Trecho de código com a configuração dos adaptadores de rede

```
def add_link(net: Containernet, node: any):
net.addLink(node, cls=adhoc,
            intf=str(node.name) + '-wlan0',
            ssid='adhocNet', proto='batman_adv',
            mode='g', channel=5, ht_cap='HT40+')
```

Todos os drones simulados são quadrotores. Os seguintes modelos de controladores de drones disponíveis no mercado foram escolhidos como referência para emulação da topologia FANET:

- **Drone 1** (mn.drone1): equipado com processador Intel *Aero Ready to Fly Drone* Atom® X7-Z8750 de 4 GB com LPDDR3-1600[INTEL, 2019].
- **Drone 2** (mn.drone2): equipado com processador da família OSD33x com 1GB de RAM DDR3[SYSTEMS, 2022].
- **Drone 3** (mn.drone3): equipado com processador *Holybro* PX4 Vision Atom x5-z8350 (até 1,92 GHz) com 4 GB de RAM [HOLYBRO, 2022].
- **Drone 4** (mn.drone4): equipado com um *Raspberry Pi4* com 2GB de RAM [PI, 2019].
- **Drone 5** (mn.drone5): equipado com um *Jetson Nano* ARM Cortex-A57 3 GB LPDDR4 [NVIDIA, 2021].

Figura 14 – Fluxo de comunicação no cenário desprotegido



Os cenários de ataque a serem analisados neste trabalho são descritos nas subseções a seguir.

5.2 Definição de casos de uso em um cenário desprotegido

Conforme descrito no preâmbulo deste capítulo, o primeiro cenário simula a transmissão das coordenadas ao longo das cinco etapas em uma rede desprotegida usando interfaces REST para simular a propagação das coordenadas de destino entre os drones.

O esquema de comunicação descrito na Seção 4.3 utiliza o o algoritmo 1 para receber os dados de localização de destino da estação base e propagar as coordenadas entre os drones. Então, cada drone valida com a estação base se a informação é legítima, conforme Figura 14.

Como FANETs podem ser utilizadas em missões críticas, a falha na comunicação com uma entidade centralizadora não pode impedir a autonomia de suas funções. Caso o sensor de um dos membros da FANET detecte alguma alteração que possa comprometer a missão, como por exemplo a presença de obstáculos ou a impossibilidade de completar a rota pelo traçado original, o impedimento deve ser comunicado aos demais UAVs para que o ajuste seja realizado, independente da comunicação com a estação base.

Nestes casos, se o esquema de validação depende de uma estação base, como no exemplo simulado, a rede pode ficar facilmente vulnerável a agentes maliciosos.

Para demonstrar o cenário em que a FANET utiliza uma interface REST desprotegida, foram estabelecidas cinco etapas caracterizadas pelas interações entre a estação base e os drones da FANET.

- Estágio 1A: no primeiro estágio, o Drone 3 recebe o destino de sua missão da estação base (GCS1) e o drone propaga as coordenadas de destino através do esquema REST desprotegido para garantir que todos os outros drones recebam as informações atualizadas. A integridade das informações relacionadas à solicitação é feita validando as informações com a estação base.

- Estágio 2A: o Drone 2 recebe uma modificação de coordenada legítima do GCS1,

valida as informações com o GCS1 e propaga a atualização das informações da missão para os outros UAVs.

- Estágio 3*A*: a FANET é comprometida quando um drone pertencente à rede modificado por ações maliciosas (Drone 5) tenta enviar um comando de atualização de destino falso para os demais UAVs, com possibilidade de validação das informações com GCS1. Como a conexão com a GCS1 ainda existe, a mensagem imprópria de atualização das coordenadas pode ser validada.

- Estágio 4*A*: a conexão com GCS1 é perdida e não é possível validar as alterações de coordenadas. O Drone 5 comprometido novamente tenta enviar um comando de atualização de destino falso para os outros UAVs.

- Estágio 5*A*: A FANET entra no alcance do sinal de comunicação com uma estação base comprometida (GCS2) que já estava autorizada a se juntar à rede. Esta estação emite uma ordem à FANET, que recebe um comando de atualização do destino da missão, sem a possibilidade de validar as informações com a GCS1.

5.3 Definição de casos de uso no cenário com PoET

Neste cenário, a estratégia foi configurar a FANET e a GCS1 como membros de uma rede conectada através do *Hyperledger Sawtooth* em modo *permissioned* (vide conceito na Seção 2.4.3). O objetivo é simular a confiabilidade da transmissão das informações trocadas entre os drones e a GCS, mesmo quando um nó validador perde a comunicação com os demais membros da rede, e medir a utilização dos recursos computacionais de cada participante da FANET no curso da simulação.

Ao contrário do cenário anterior, a validação dos dados é feita por todos os membros da FANET, já que todos os drones estão configurados como validadores e de forma descentralizada (vide Seção 2.1.1.2 e 2.4.2). Conforme esclarecido no Capítulo 4, o algoritmo de consenso elegido para ser utilizado pelo protocolo do *Sawtooth* é o PoET. Como o consenso é baseado no tempo, um temporizador é utilizado para determinar qual será o nó da rede responsável por criar o próximo bloco na *blockchain*.

No cenário simulado, durante a inicialização da rede, a GCS1 é configurada para atuar como o nó líder responsável por gerar o bloco gênese para inicializar a *blockchain*. As mensagens contendo os dados das coordenadas de destino são processadas pelo *Intkey Transaction Processor* (vide Seção 4.2) e seguem nos blocos com um temporizador gerado aleatoriamente. Durante o processo de troca de mensagens, o drone que receber o temporizador com o menor valor, é eleito o líder do consenso e será responsável em gerar o próximo bloco. Portanto, se espera que a rede seja CFT mesmo com a perda da comunicação com a GCS.

Foram simulados os seguintes cenários com a introdução do Sawtooth nos dispositivos que compõem a FANET proposta:

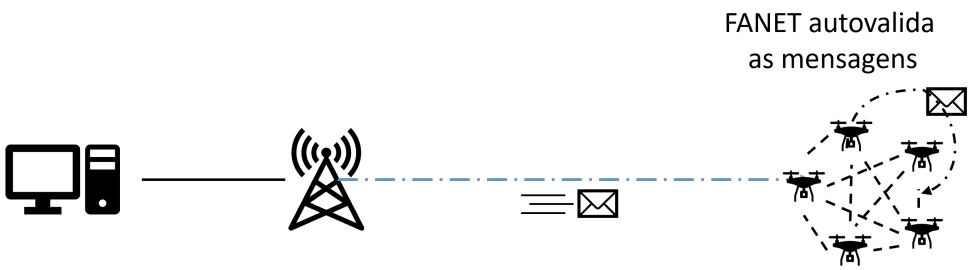
- Estágio $1B$: No estágio inicial, a FANET recebe uma coordenada de destino da GCS1 e as informações da missão são enviadas à rede *Sawtooth* para validar as informações.
- Estágio $2B$: a GCS1 envia alterações nas coordenadas e a rede *Sawtooth* valida a atualização das informações.
- Estágio $3B$: um UAV comprometido na FANET (Drone 5) tenta enviar um comando de atualização de destino falso para os outros UAVs pela interface REST desprotegida, tal como no cenário $3A$.
- Estágio $4B$: a conexão com o GCS1 é perdida e não é possível validar as alterações de coordenadas com o auxílio do GCS1. Neste interim o Drone 2 precisa reorganizar as coordenadas do destino devido a um desvio de emergência legítimo.
- Estágio $5B$: a FANET entra no alcance de comunicação da GCS2 que está comprometida por agentes maliciosos e ingressa na rede, para tentar alterar as coordenadas de destino através da interface REST desprotegida.

Observe que nesta topologia, os agentes maliciosos tentam acessar e modificar as coordenadas da missão pela utilização dos mesmos *endpoints* da interface REST desprotegida. Contudo, no presente cenário o esquema de comunicação não utiliza essas interfaces, pois o protocolo *Sawtooth* possui suas próprias APIs REST para se comunicar (vide 4.2 e Figura 9). O drone comprometido (Drone 5) e a segunda estação base (GCS2) estão fora da rede *Sawtooth*, portanto não podem transacionar informações nem validar os blocos usando o *Hyperledger Sawtooth*.

5.4 Resultados

A simulação demonstrou que no cenário em que a FANET utiliza a metodologia baseada em comunicação RESTFul desprotegida, a FANET operou conforme o esperado

Figura 15 – Fluxo de comunicação numa rede habilitada com *Hyperledger Sawtooth*



entre as Etapas $1A$ a $3A$, sendo malsucedida na prevenção de ações maliciosas perpetradas por dispositivos da rede comprometidos.

Por sua vez, a FANET habilitada com o esquema de comunicação baseado no *Hyperledger Sawtooth* conseguiu prevenir ações maliciosas mesmo perdendo comunicação com a GCS.

Como o escopo do presente trabalho é também analisar o custo computacional implícito na utilização da *blockchain* para evitar a alteração nos dados da missão de controle em dispositivos com baixa capacidade computacional, durante todo o processo foram gerados relatórios baseados no monitoramento do sistema simulado.

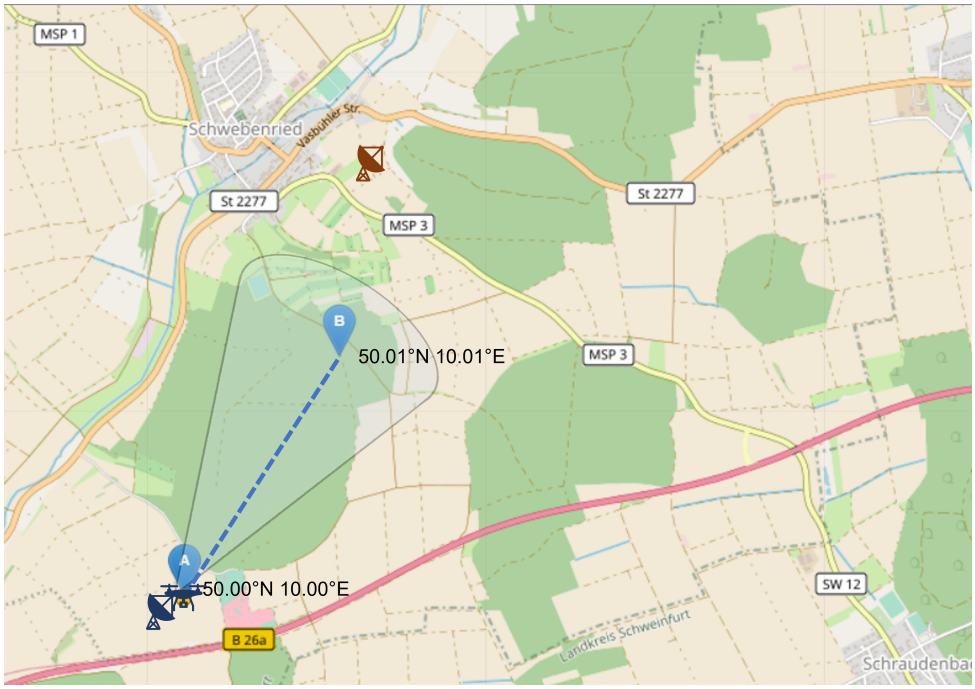
5.4.1 Simulação de rede desprotegida

Na simulação com a FANET desprotegida descrita em 5.2, os cinco estágios foram executados por aproximadamente 15 minutos. Os resultados foram coletados com o log de eventos e as métricas monitoradas usando os métodos mencionados em 4.3 e Figuras 11, 12 e 13.

Nas três primeiras etapas, as FANETs conseguem validar os comandos de alteração de coordenadas finais com sucesso.

A GCS envia as coordenadas de destino tanto na Etapa $1A$ utilizando os endpoints da interface REST (utilizando o modelo do algoritmo 1 explicado na Seção 5.2), estabelecendo o ponto final as coordenadas 50.01ºN, 10.01ºE, conforme Figura 16.

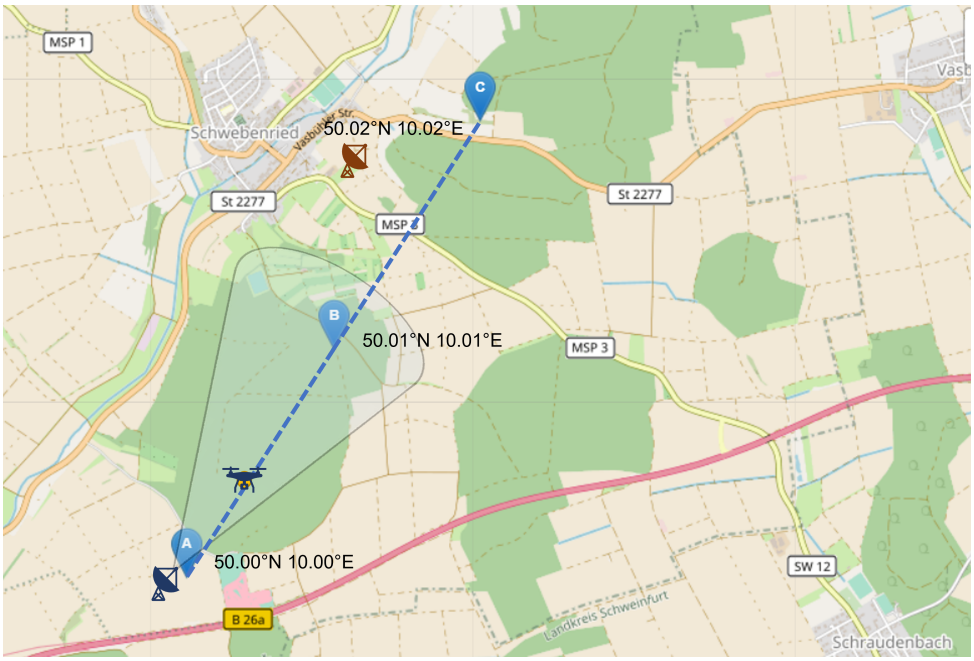
Figura 16 – Etapa 1: A GCS envia as coordenadas de destino (50.01ºN, 10.01ºE)



Em seguida, a GCS envia um comando à FANET para atualizar as coordenadas

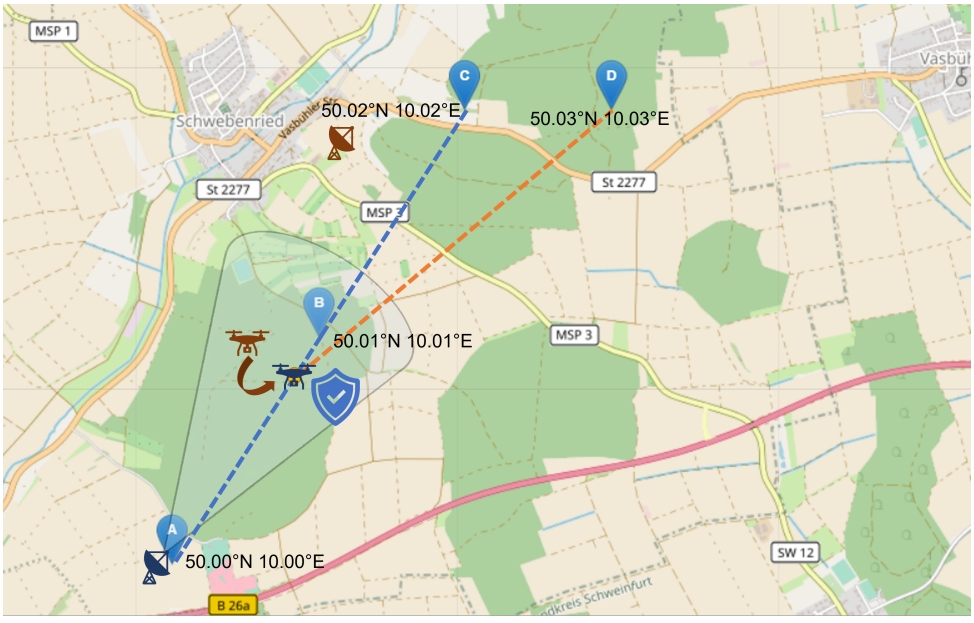
do destino final para 50.02ºN, 10.02ºN, sendo atendido com sucesso na Etapa 2*A* (Figura 17).

Figura 17 – Etapa 2: A GCS atualiza as coordenadas de destino para 50.02ºN, 10.02ºE



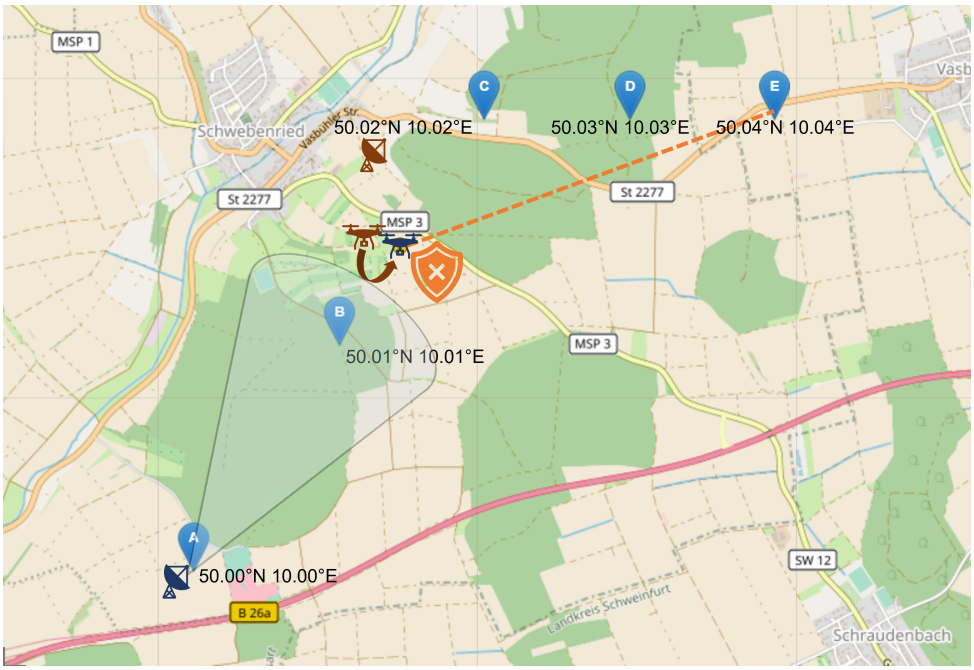
Na Etapa 3*A*, um agente malicioso infiltrado na FANET que está pré-programado por um *Insider* para realizar o ataque de *Modificação* e alterar os dados originais da missão (*Data tampering*, vide Seção 2.3.3), envia a ordem à FANET para definir as coordenadas de destino 50.02ºN, 10.03ºE. A Figura 18 ilustra como o mecanismo de validação consegue impedir a ação maliciosa com sucesso.

Figura 18 – Etapa 3: O drone 5 pertencente à rede está comprometido e tenta atualizar as coordenadas de destino para 50.02ºN, 10.03ºE



À partir da Etapa 4A, a FANET se distancia e perde a comunicação com a GCS1. Diante da impossibilidade de validar a requisição de mudança de coordenadas, a ação maliciosa do Drone 5 é executada com sucesso para alterar as coordenadas de destino para 50.02ºN, 10.04ºE (Figura 19).

Figura 19 – Etapa 4A: O drone 5 comprometido e tenta novamente atualizar as coordenadas de destino para 50.02ºN, 10.04ºE, estando a FANET fora de alcance da GCS1



Por fim, na Etapa 5A, a GCS2 previamente autorizada a se conectar à FANET está igualmente comprometida por agentes maliciosos e usa sua conexão com a rede para enviar novos dados de coordenadas de destino 50.02ºN, 10.04ºE. Conforme ilustrado na Figura 20, a mudança não autorizada de coordenadas é realizada com sucesso.

A Figura 21 resume os resultados de performance de recursos computacionais obtidos durante a execução das Etapas 1A a 5A. Cada linha pontilhada cinza representa a transição entre as etapas simulados. Os cenários destacados com fundo vermelho representam as etapas onde houve a simulação em que a FANET foi comprometida por ações maliciosas.

Entre todos os dispositivos pertencentes à FANET (incluindo a estação base nesta análise), o pico de consumo da CPU ocorreu ao final da execução do Etapa 1A no Drone 3 quando atinge $11,51\%$, levando em consideração que este é o drone responsável pela propagação de coordenadas para a FANET no início da Etapa 1A. Posteriormente, na Etapa 2A, o Drone 2 atinge o pico de consumo da CPU de $10,74\%$ e permanece ativo até o final da simulação. Também é possível observar um aumento no consumo da CPU do Drone 3, pois ele recebe a solicitação de mudança de coordenadas da GCS2 comprometida durante o Etapa 5A.

Figura 20 – Etapa 5*A*: A GCS2 comprometida acessa a rede FANET e atualiza as coordenadas de destino para 50.05ºN, 10.05ºE, estando a FANET fora de alcance da GCS1

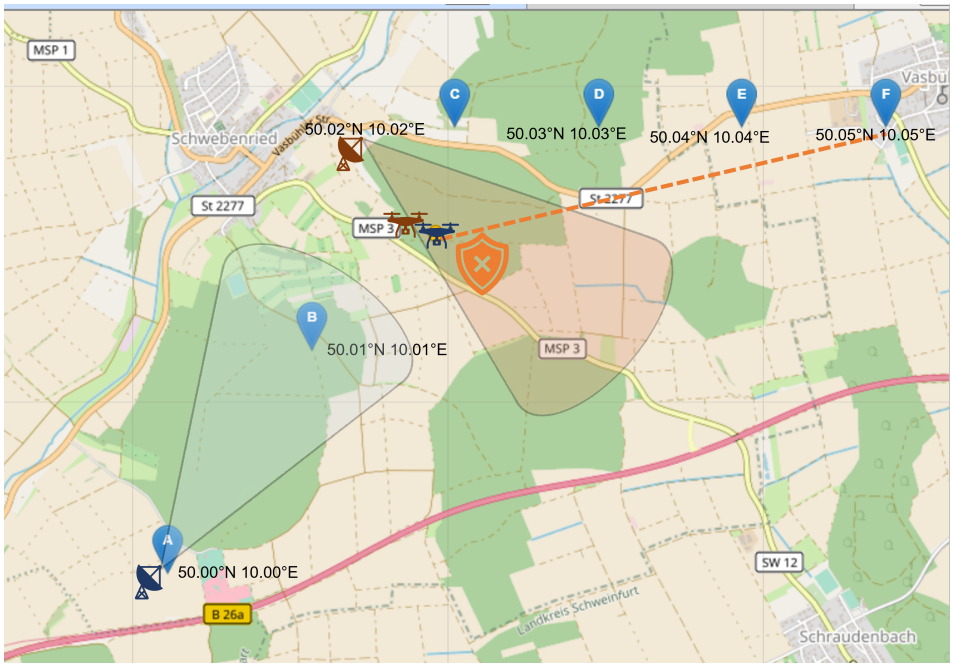
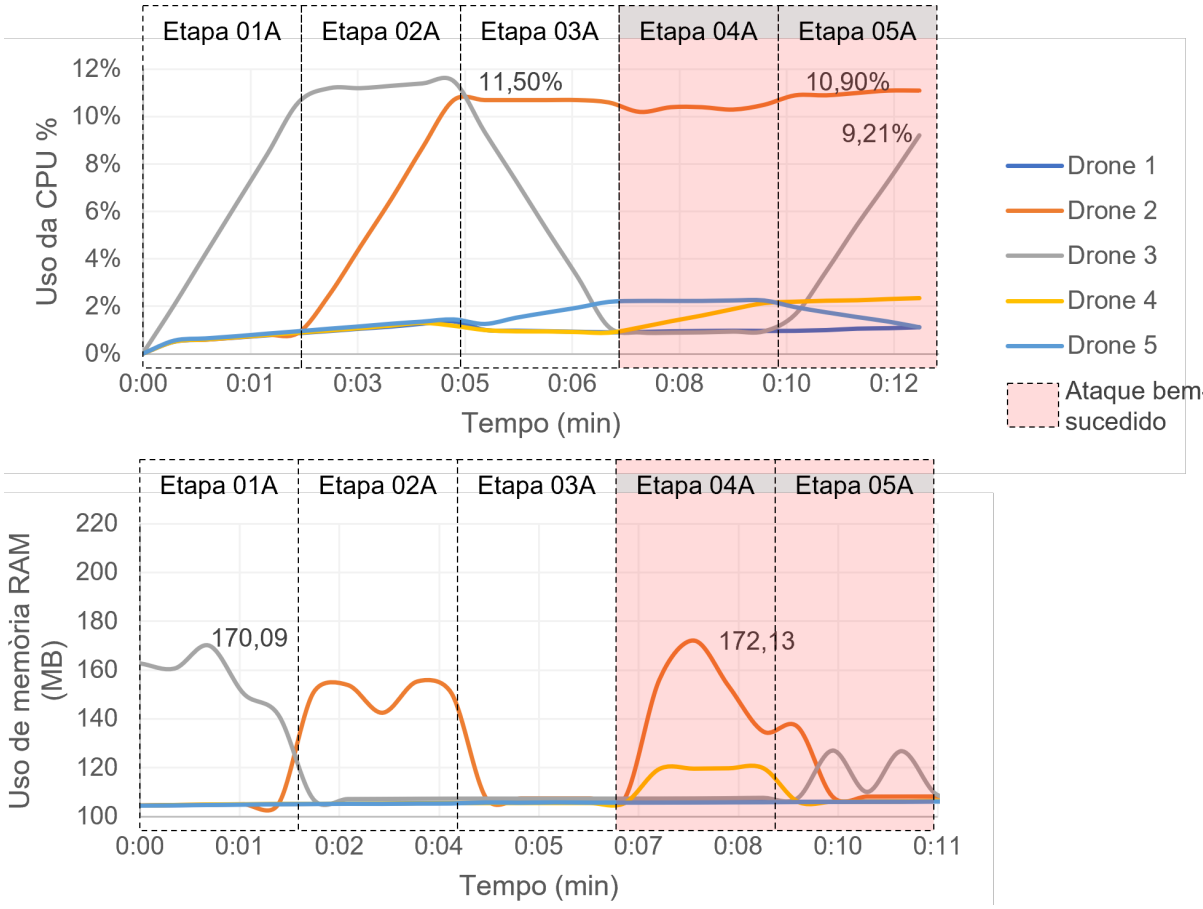


Figura 21 – Recursos consumidos pelas estações nos cenários desprotegidos



Quanto ao consumo de memória, os peers pertencentes à FANET mantiveram-se estáveis entre 99, 66 MB e 102, 31 MB, com três picos mais altos de 162, 21 MB, 121, 14 MB e 120, 90 MB no Drone 3 (mn. drone3) e dois picos de 148, 11 MB e 164, 16 MB no Drone 2 (mn.drone2).

5.4.2 Simulação FANET habilitada com Hyperledger Sawtooth

A simulação usando a comunicação entre os pares da FANET usando o Hyperledger Sawtooth com o mecanismo de consenso que usa o algoritmo PoET foi executada aproximadamente na mesma quantidade de tempo e condições da execução anterior.

Durante a simulação nas Etapas $1B$, $2B$ e $3B$, as FANETs se comportaram similarmente aos cenário descrito na Seção 5.4.1. Ou seja, na Etapa $1B$ a GCS envia as coordenadas de de destino 50.01ºN, 10.01ºE; na Etapa $2B$ a GCS atualiza as coordenadas de destino para 50.02ºN, 10.02ºE; finalmente, na Etapa $3B$, o Drone 5 tenta alterar maliciosamente e sem êxito as coordenadas de destino para 50.02ºN, 10.03ºE. Tudo como ilustrado nas Figuras 16, 17 e 18, respectivamente.

Contudo, ao contrário do Cenário REST desprotegido, na Etapa 04B a FANET recebe novamente a requisição do Drone 05 comprometido para alterar as coordenadas para o destino 50.02ºN, 10.03ºE e consegue rechaçar a ação maliciosa, mantendo sua rota ao destino. Somado a isso, o Drone 2 precisa fazer um modificação de coordenadas por motivos de emergência e a ação é validada e executada com sucesso, alterando a rota para o destino 50.02ºN, 10.04ºE, conforme Figura 22.

Durante a Etapa 05B, a GCS2 comprometida e previamente autorizada a se juntar à FANET quando no alcançe de sinal, se conecta à rede e tenta alterar as coordenadas de destino para 50.02ºN, 10.05ºE. A ação maliciosa é malsucedida (Figura 23, já que a GCS2 a interface REST desprotegida não pertence aos nós da *blockchain*.

Conforme apresentado na Figura 24, o consumo máximo da CPU não excedeu 11.03% (GCS1 ou mn.base1 durante o início da Etapa $2B$). O consumo médio de drones habilitados para Sawtooth variou de 1, 76% a 5, 72% de uso da CPU.

A memória permaneceu estável em uma média de 162, 00 MB a 167, 72 MB entre os pares Sawtooth com poucos picos não superiores a 189, 73 MB na GCS1 (mn.base1), 188, 96 MB no Drone 2 (mn.drone2) e 188, 44MB no Drone 4 (mn. drone4).

5.5 Discussões

Comparando os resultados entre os dois conjuntos de cenários, o ponto mais relevante é que com a comunicação desprotegida, a tentativa de sequestro foi bem sucedida quando a conexão foi perdida com a estação base, enquanto no caso da comunicação com

Figura 22 – Etapa 4*B*: O drone 5 comprometido e tenta novamente atualizar as coordenadas de destino para 50.02ºN, 10.03ºE, estando a FANET fora de alcance da GCS1; enquanto isso, o drone 2 precisa fazer uma alteração legítima de rota para o destino 50.02ºN, 10.04ºE

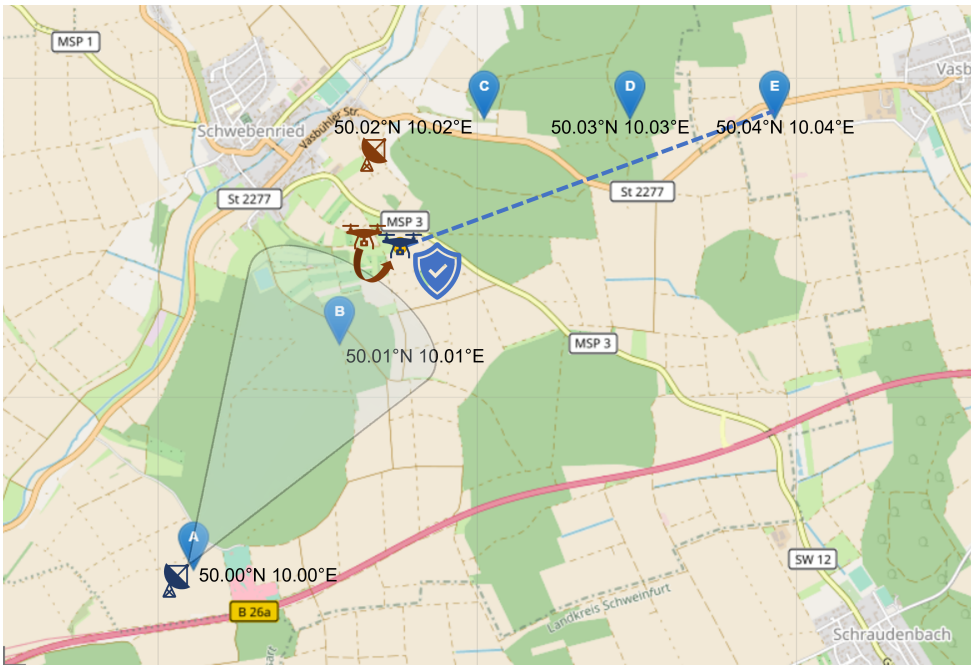


Figura 23 – Etapa 5*B*: A GCS2 comprometido e tenta novamente atualizar as coordenadas de destino para 50.02ºN, 10.05ºE

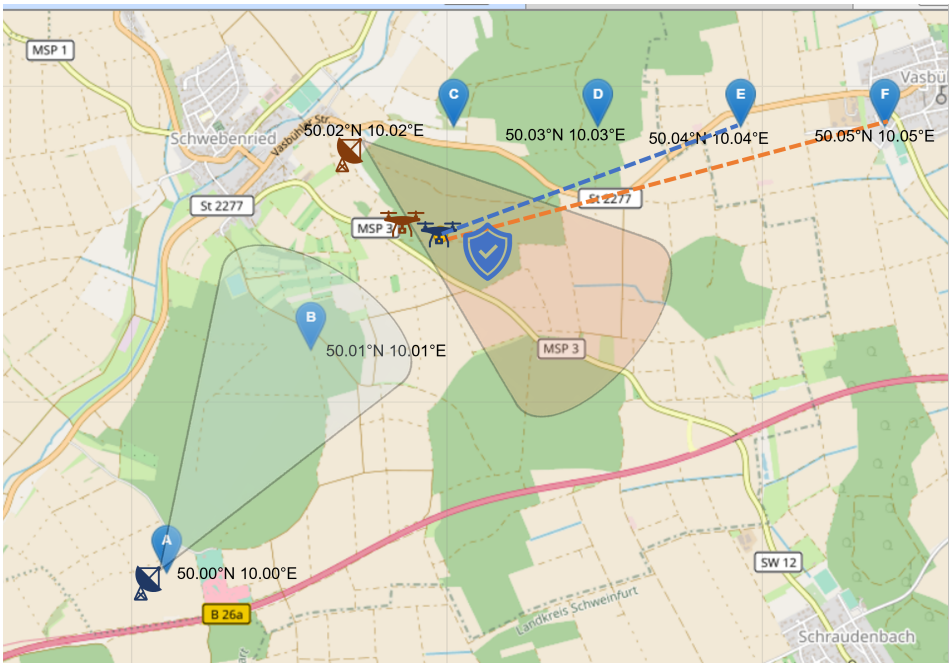
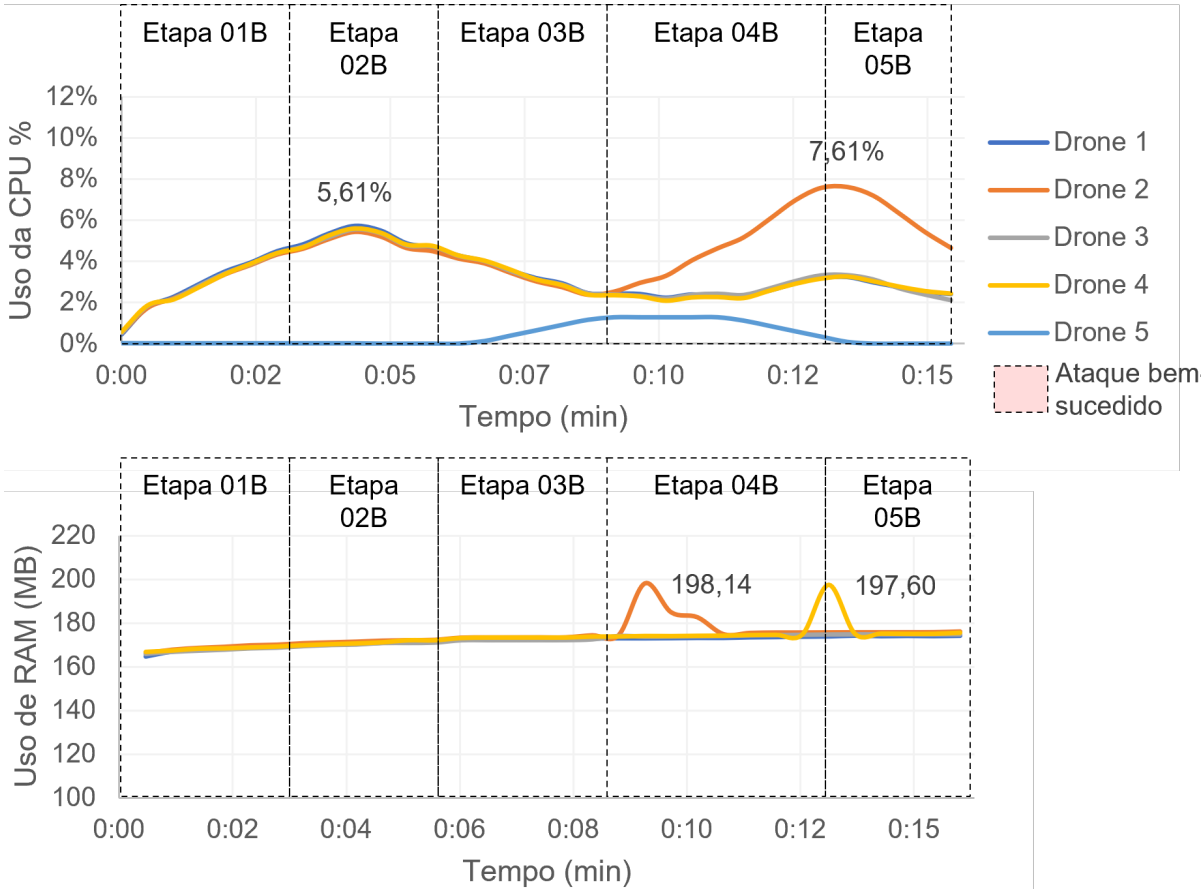


Figura 24 – Recursos usados pelas estações com Hyperledger Sawtooth



Sawtooth, todos os drones permaneceram com suas rotas inalteradas, o que mostra a resiliência da arquitetura com a solução proposta.

No cenário onde o *Hyperledger Sawtooth* foi empregado (Etapas $1B$ a $5B$), todos os drones permaneceram nas coordenadas esperadas, inclusive durante a perda de conectividade com a GCS1 (mn.base1), validando assim a integridade da comunicação através do mecanismo selecionado. Em contraste, conforme destacado na Tabela 9 e ilustrado nas Figuras 19 e 20, nas Etapas $4A$ e $5A$, a FANET foi explorada e o curso foi alterado por agentes maliciosos.

Diante disso, a FANET com *Hyperledger Sawtooth* habilitado, foi capaz de prevenir os ataques de modificação e *data tampering* não autorizado pelo sistema realizado por um *insider* simulado. Portanto, a simulação indica que a aplicação da *blockchain* pode garantir os controle de segurança da **integridade**, **não-repúdio** e a **autorização** (vide Tabela 2.

Do ponto de vista do consumo de recursos, os drones responsáveis por propagar a mudança de coordenadas tiveram picos de uso da CPU de aproximadamente 12%. O Drone 2 trabalhou por 50% da simulação com um uso médio da CPU próximo a 11%, enquanto os outros drones não responsáveis pela validação permaneceram em níveis inferiores (cerca de $1,73\%$ de uso da CPU). No entanto, dos Cenários $1B$ a $5B$, com Sawtooth, todos

Tabela 9 – Destino final por drone (A linha destacada mostra as etapas onde a ação maliciosa foi bem-sucedida)

Etapa	Valor esperado	Drone1	Drone2	Drone3	Drone4	Drone5
$1A$	50.01,10.01	50.01,10.01	50.01,10.01	50.01,10.01	50.01,10.01	50.01,10.01
$2A$	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02
$3A$	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02
$4A$	50.02,10.02	50.40,10.40	50.40,10.40	50.40,10.40	50.40,10.40	50.40,10.40
$5A$	50.01,10.01	50.01,10.01	50.01,10.01	50.01,10.01	50.01,10.01	50.01,10.01
$1B$	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02
$2B$	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02
$3B$	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02	50.02,10.02
$4B$	50.02,10.04	50.04,10.04	50.04,10.04	50.04,10.04	50.04,10.04	50.04,10.04
$5B$	50.02,10.04	50.04,10.04	50.04,10.04	50.04,10.04	50.04,10.04	50.04,10.04

os drones permaneceram com uso da CPU estável, em torno de $3,06\%$ a $4,45\%$, onde os maiores valores correspondem aos *peers* responsáveis por comprometer novos blocos naquele instante de a simulação (Figuras 25 e 26).

Figura 25 – Comparação na utilização de recursos da CPU em ambos os cenários

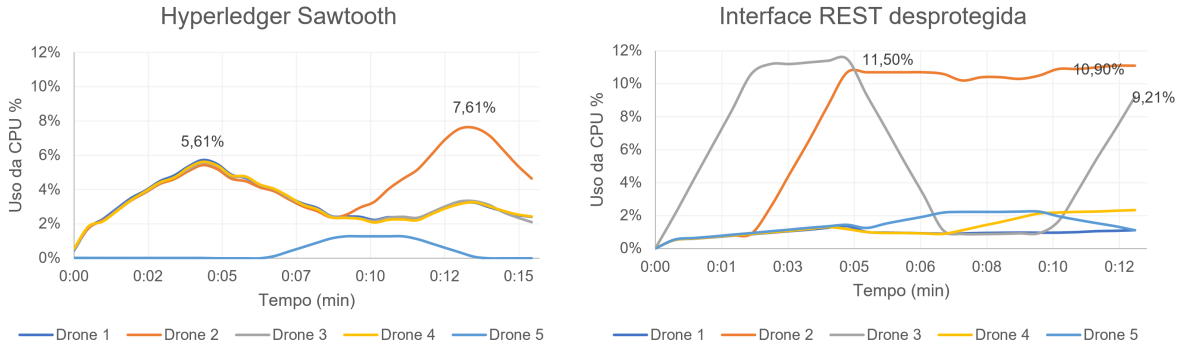
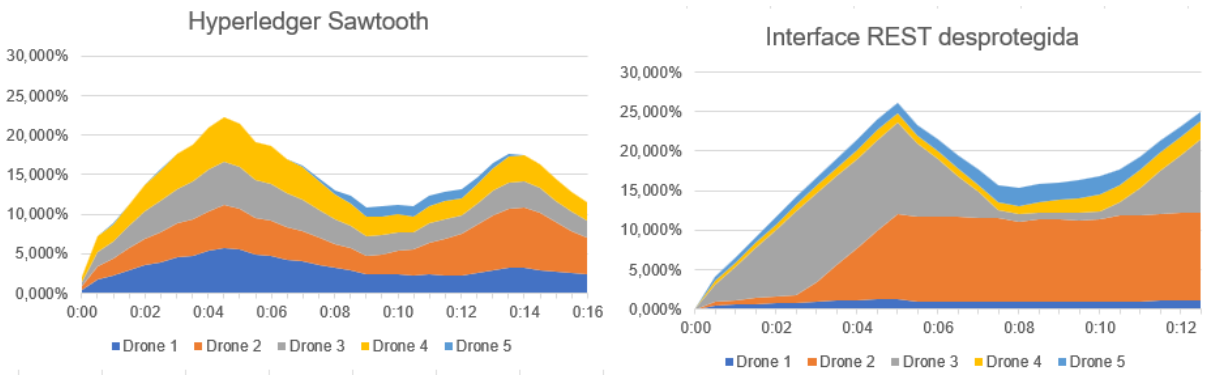


Figura 26 – Comparação da somatória na utilização dos recursos da CPU em ambos os cenários



Já o consumo de memória permaneceu em uma média de 100,0 MB em todos os drones nos cenários simulados com a interface REST desprotegida e, em contraste, o consumo de memória permaneceu estável em uma média de 168,0 MB na FANET com o protocolo *Sawtooth* (Figuras 27 e ref 28). Importante salientar que no referido gráfico, o Drone 5 foi desconsiderado no cenário com o protocolo *Sawtooth*, por não ser um nó de validação, mas por meramente representar um dispositivo intruso.

Figura 27 – Comparação na utilização de uso de RAM em ambos os cenários

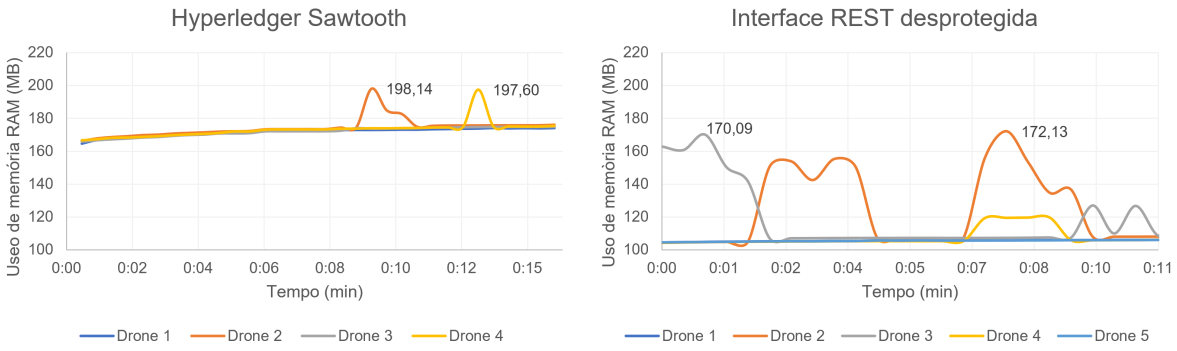
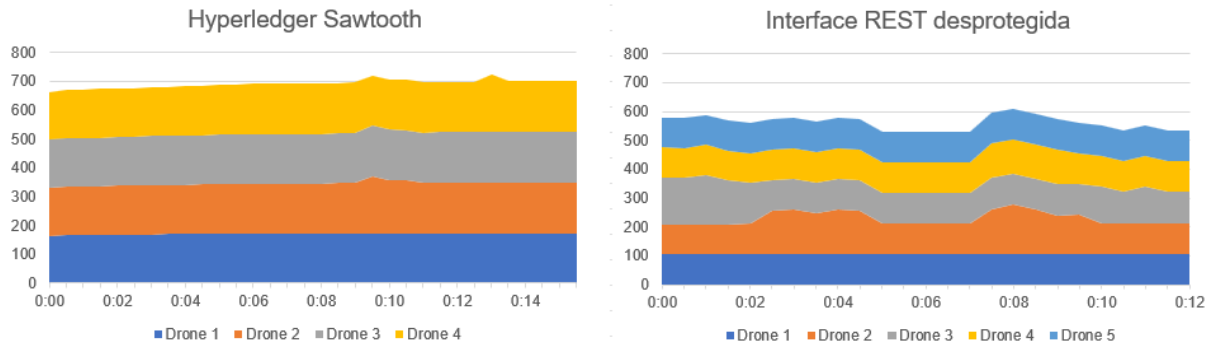


Figura 28 – Comparação da somatória na utilização de uso de RAM em ambos os cenários



Quanto à escrita da dados em disco, o cenário com *Sawtooth* apresentou uma grande performance em relação ao cenário com a interface REST desprotegida. Todo o trabalho foi executado em memória na FANET com *Sawtooth*, enquanto que a atividade de escrita de dados foi intensa no cenário desprotegido, conforme Figuras 29 e 30.

Como em ambos os cenários a simulação tratou de transacionar uma pequena quantidade de dados relacionados à coordenada de destino da FANET, o volume de escrita consequentemente é pequeno. Contudo, em cenários reais, a transmissão de dados da missão e dos sensores é maior, o que implicaria em maior atividade de escrita.

Como o *Sawtooth* trabalhou com todo o processamento, validação e transação dos dados em memória, assume-se que o custo energético seria menor já que, em geral, o uso de memória consome menos energia do que a escrita em disco [BAHN; CHO, 2020].

Apesar da solução proposta fornecer uma arquitetura CFT, ela não evita que o sistema apresente um comportamento arbitrário (bizantino), característica discutida no Capítulo 2. A arquitetura CFT é projetada para garantir que todos os nós em um sistema distribuído cheguem a um consenso sobre um determinado valor ou decisão, mesmo que alguns dos nós falhem ou enviem informações incorretas. Isso é alcançado por meio de algoritmos de consenso, como o algoritmo de consenso de BFT, que é especialmente projetado para lidar com falhas bizantinas.

No entanto, mesmo com uma arquitetura CFT, o sistema ainda pode apresentar um comportamento arbitrário se um ou mais nós decidirem enviar informações incorretas ou agir de forma mal-intencionada. Esse comportamento é conhecido como comportamento bizantino e pode levar a decisões incorretas ou conflitos no sistema. Isso poderia ser garantido com o uso do algoritmo de consenso PBFT, que não é recomendado em redes pequenas, pois requer pelo menos quatro pares para funcionar dada a proporção $\frac{(n-1)}{3}$ [CASTRO; LISKOV, 2002]. Além disso, neste momento, o consenso *Sawtooth* PBFT não fornece desconexão de rede dinâmica.

Figura 29 – Comparação na escrita de dados em disco em ambos os cenários

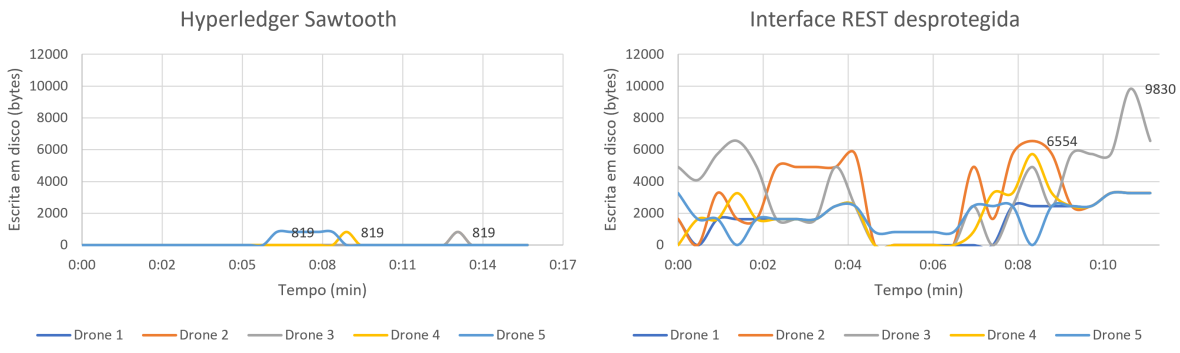
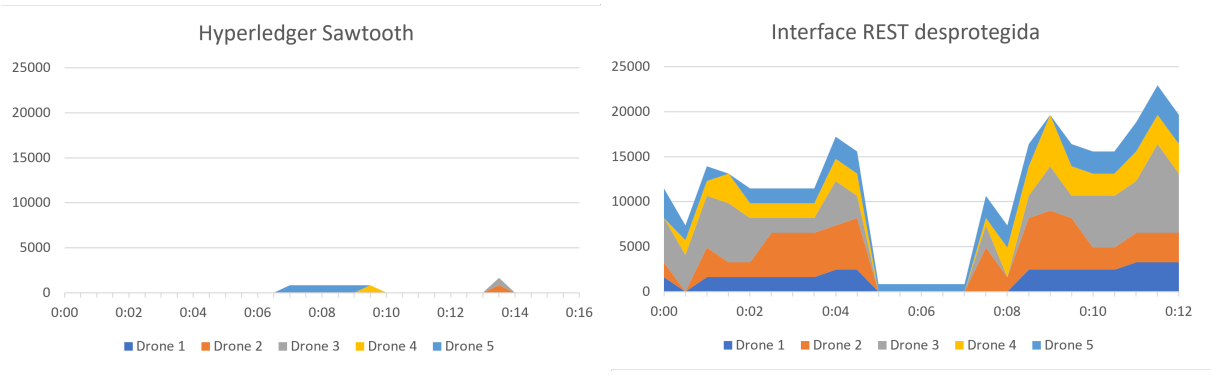


Figura 30 – Comparação da somatória na escrita de dados em disco em ambos os cenários



6 Conclusão

A utilização de veículos aéreos não tripulados (UAVs) tem se mostrado cada vez mais comum em diferentes áreas, tais como agricultura, monitoramento ambiental, inspeções de infraestruturas e até mesmo entregas. No entanto, o uso desses equipamentos também traz consigo uma série de desafios, incluindo a segurança das transmissões de dados realizadas pelos UAVs.

Nesse contexto, a tecnologia *blockchain* tem sido amplamente estudada e utilizada como uma solução para garantir a segurança e privacidade das transações digitais. No caso dos UAVs, a utilização de *blockchain* pode ser uma solução eficaz para proteger as informações transmitidas pelos veículos aéreos não tripulados.

O presente trabalho apresentou uma solução econômica baseada em *blockchain* para proteger os dados transmitidos entre UAVs. Através da realização de testes em uma implementação de código aberto, os resultados obtidos demonstraram que o sistema é capaz de garantir a confiabilidade das transmissões e evitar uma série de ataques cibernéticos.

Um dos principais benefícios da utilização de *blockchain* em conjunto com UAVs é a autenticidade, integridade e confidencialidade dos dados transmitidos pelos UAVs. Além disso, a tecnologia *blockchain* possibilita o controle de acesso aos dados e a proteção da privacidade dos usuários, o que é especialmente importante em aplicações sensíveis. Na simulação realizada, a utilização do esquema proposto mitigou a ação maliciosa de *insiders* de realizar a modificação de dados com o propósito de alterar o trajeto da missão, assegurando desta maneira os controles de segurança da integridade, não-repúdio e autorização na troca de dados.

Outra vantagem da solução proposta é o baixo consumo de recursos computacionais. Os testes realizados mostraram que o uso de CPU é mantido baixo, e a alocação de memória é mantida constante, sem picos significativos de uso.

Apesar dos resultados positivos obtidos, ainda há espaço para melhorias e aprimoramentos na solução proposta. Como futuras pesquisas, pode-se destacar a realização de simulações em redes FANETs *multi-swarm* com uma população maior de *peers* e a utilização de outros algoritmos de consenso para medir o desempenho em diferentes cenários.

Ademais, a utilização de *blockchain* em sistemas de UAVs apresenta-se como uma solução promissora para a proteção de transmissões de dados e para garantir a segurança das operações realizadas por esses veículos aéreos não tripulados. A tecnologia *blockchain* pode ser aplicada em diferentes áreas, como na agricultura, onde pode ser utilizada para

monitorar o crescimento das plantas e identificar possíveis problemas, ou na área de transporte, para garantir a segurança e privacidade das informações transmitidas durante as entregas.

Por fim, é importante destacar que a utilização de *blockchain* em conjunto com UAVs não é a solução para todos os desafios enfrentados por esses equipamentos. Outras questões, como a segurança física dos UAVs e a regulamentação da utilização desses equipamentos, também precisam ser consideradas para garantir uma operação segura e eficiente.

Anexos

ANEXO A – Dados compilados de utilização de RAM, CPU, escrita de disco

Tabela 10 – Consumo de CPU com *Hyperledger Sawtooth*

Data/Hora	Drone 1	Drone 2	Drone 3	Drone 4	Drone 5
12:30:00	0,454%	0,511%	0,503%	0,566%	0,024%
12:30:30	1,760%	1,730%	1,820%	1,850%	0,019%
12:31:00	2,270%	2,200%	2,170%	2,140%	0,018%
12:31:30	2,900%	2,760%	2,820%	2,750%	0,018%
12:32:00	3,510%	3,370%	3,420%	3,390%	0,018%
12:32:30	3,950%	3,830%	3,880%	3,900%	0,017%
12:33:00	4,490%	4,330%	4,400%	4,410%	0,017%
12:33:30	4,820%	4,610%	4,720%	4,690%	0,017%
12:34:00	5,350%	5,080%	5,190%	5,270%	0,017%
12:34:30	5,720%	5,430%	5,570%	5,610%	0,017%
12:35:00	5,490%	5,200%	5,370%	5,400%	0,000%
12:35:30	4,870%	4,640%	4,780%	4,820%	0,000%
12:36:00	4,670%	4,490%	4,730%	4,760%	0,000%
12:36:30	4,280%	4,130%	4,270%	4,290%	0,000%
12:37:00	4,010%	3,900%	4,020%	4,040%	0,125%
12:37:30	3,560%	3,440%	3,540%	3,600%	0,386%
12:38:00	3,180%	3,020%	3,120%	3,120%	0,641%
12:38:30	2,930%	2,740%	2,870%	2,840%	0,902%
12:39:00	2,450%	2,370%	2,440%	2,380%	1,150%
12:39:30	2,430%	2,520%	2,380%	2,350%	1,270%
12:40:00	2,410%	2,950%	2,320%	2,290%	1,270%
12:40:30	2,230%	3,290%	2,180%	2,080%	1,270%
12:41:00	2,390%	4,040%	2,370%	2,240%	1,270%
12:41:30	2,310%	4,630%	2,420%	2,260%	1,270%
12:42:00	2,310%	5,160%	2,360%	2,210%	1,110%
12:42:30	2,580%	6,050%	2,680%	2,570%	0,854%
12:43:00	2,930%	6,970%	3,040%	2,910%	0,593%
12:43:30	3,190%	7,570%	3,320%	3,160%	0,336%
12:44:00	3,240%	7,610%	3,330%	3,260%	0,085%
12:44:30	2,980%	7,190%	3,120%	3,030%	0,000%
12:45:00	2,740%	6,330%	2,690%	2,770%	0,000%
12:45:30	2,520%	5,420%	2,380%	2,550%	0,000%
12:46:00	2,410%	4,650%	2,110%	2,430%	0,000%

Tabela 11 – Consumo de CPU no cenário com a interface REST desprotegida

Data/Hora	Drone 1	Drone 2	Drone 3	Drone 4	Drone 5
12:06:00	0,017%	0,017%	0,030%	0,017%	0,015%
12:06:30	0,512%	0,527%	2,050%	0,516%	0,559%
12:07:00	0,598%	0,598%	4,190%	0,605%	0,650%
12:07:30	0,686%	0,696%	6,310%	0,700%	0,743%
12:08:00	0,780%	0,800%	8,420%	0,797%	0,852%
12:08:30	0,871%	0,891%	10,600%	0,888%	0,953%
12:09:00	0,963%	2,500%	11,200%	0,983%	1,060%
12:09:30	1,060%	4,530%	11,200%	1,080%	1,160%
12:10:00	1,150%	6,540%	11,300%	1,190%	1,270%
12:10:30	1,270%	8,700%	11,400%	1,300%	1,360%
12:11:00	1,370%	10,700%	11,500%	1,180%	1,450%
12:11:30	1,010%	10,700%	9,350%	1,000%	1,260%
12:12:00	0,973%	10,700%	7,310%	0,946%	1,520%
12:12:30	0,953%	10,700%	5,230%	0,942%	1,740%
12:13:00	0,929%	10,700%	3,220%	0,915%	1,950%
12:13:30	0,905%	10,600%	1,120%	0,888%	2,190%
12:14:00	0,922%	10,200%	0,892%	1,110%	2,230%
12:14:30	0,949%	10,400%	0,881%	1,370%	2,230%
12:15:00	0,963%	10,400%	0,895%	1,610%	2,230%
12:15:30	0,973%	10,300%	0,929%	1,880%	2,250%
12:16:00	0,966%	10,500%	0,939%	2,130%	2,250%
12:16:30	0,969%	10,900%	1,650%	2,190%	1,970%
12:17:00	1,000%	10,900%	3,470%	2,230%	1,750%
12:17:30	1,060%	11,000%	5,440%	2,250%	1,550%
12:18:00	1,080%	11,100%	7,280%	2,300%	1,360%
12:18:30	1,120%	11,100%	9,210%	2,340%	1,120%

Tabela 12 – Consumo de RAM no cenário com *Hyperledger Sawtooth*

Data/Hora	Drone 1	Drone 2	Drone 3	Drone 4	Drone 5
12:30:30	164,605952	166,293504	165,863424	166,72768	2,994176
12:31:00	166,866944	167,878656	166,8096	167,370752	2,994176
12:31:30	167,75168	168,706048	167,354368	168,22272	2,994176
12:32:00	168,083456	169,213952	167,825408	168,390656	2,994176
12:32:30	168,57088	169,90208	168,448	168,812544	2,994176
12:33:00	169,086976	170,16832	168,759296	169,209856	2,994176
12:33:30	169,705472	170,848256	169,463808	169,828352	2,994176
12:34:00	169,967616	171,220992	169,967616	170,274816	2,994176
12:34:30	170,364928	171,552768	170,201088	170,631168	2,994176
12:35:00	170,876928	172,003328	170,909696	171,15136	2,994176
12:35:30	171,307008	172,220416	170,92608	172,105728	2,994176
12:36:00	171,33568	172,376064	171,085824	172,085248	2,994176
12:36:30	172,27776	173,379584	172,183552	173,129728	2,994176
12:37:00	172,4416	173,50656	172,212224	173,330432	4,030464
12:37:30	172,453888	173,514752	172,220416	173,338624	4,284416
12:38:00	172,470272	173,531136	172,228608	173,34272	4,276224
12:38:30	172,478464	173,535232	172,240896	173,350912	4,36224
12:39:00	172,875776	174,460928	172,589056	173,785088	4,333568
12:39:30	172,978176	174,481408	174,14144	173,805568	4,354048
12:40:00	172,965888	198,144	173,83424	174,063616	4,354048
12:40:30	173,039616	184,963072	173,907968	174,014464	4,354048
12:41:00	173,162496	182,706176	174,043136	174,145536	4,354048
12:41:30	173,150208	174,931968	174,125056	174,227456	4,354048
12:42:00	173,416448	175,53408	174,456832	174,501888	4,354048
12:42:30	173,531136	175,726592	174,53056	174,592	4,354048
12:43:00	173,71136	175,730688	174,690304	174,804992	4,354048
12:43:30	173,842432	175,788032	174,829568	197,595136	4,354048
12:44:00	174,178304	175,837184	174,891008	175,050752	4,354048
12:44:30	173,940736	175,84128	174,8992	175,173632	4,354048
12:45:00	174,071808	175,853568	174,907392	175,06304	4,354048
12:45:30	173,969408	175,869952	174,911488	175,075328	4,354048
12:46:00	174,133248	176,181248	174,923776	175,55456	4,354048

Tabela 13 – Consumo de RAM no cenário com a interface REST desprotegida

Time	Drone 1	Drone 2	Drone 3	Drone 4	Drone 5
12:06:30	104,48896	104,456192	162,803712	104,419328	104,505344
12:07:00	104,636416	104,63232	160,710656	104,628224	104,624128
12:07:30	104,820736	104,792064	170,094592	104,849408	104,759296
12:08:00	104,988672	104,935424	150,253568	104,91904	104,89856
12:08:30	105,099264	105,107456	141,541376	105,078784	105,054208
12:09:00	105,18528	150,614016	107,094016	105,115648	105,160704
12:09:30	105,283584	153,874432	107,106304	105,213952	105,242624
12:10:00	105,295872	142,475264	107,155456	105,259008	105,218048
12:10:30	105,402368	155,29984	107,216896	105,357312	105,34912
12:11:00	105,422848	150,368256	107,261952	105,418752	105,385984
12:11:30	105,590784	107,331584	107,282432	105,439232	105,766912
12:12:00	105,660416	107,331584	107,282432	105,439232	105,660416
12:12:30	105,725952	107,331584	107,282432	105,439232	105,803776
12:13:00	105,730048	107,331584	107,282432	105,439232	105,689088
12:13:30	105,750528	107,331584	107,282432	105,439232	105,693184
12:14:00	105,816064	156,082176	107,298816	119,447552	105,71776
12:14:30	105,857024	172,130304	107,339776	119,56224	105,766912
12:15:00	105,92256	153,423872	107,429888	119,721984	105,824256
12:15:30	105,947136	134,787072	107,589632	119,779328	105,852928
12:16:00	106,00448	136,66304	107,524096	105,984	105,938944
12:16:30	106,037248	108,085248	127,021056	105,975808	105,971712
12:17:00	106,0864	108,085248	110,010368	106,012672	106,012672
12:17:30	106,115072	108,101632	126,775296	106,078208	106,041344
12:18:00	106,164224	108,130304	109,03552	106,336256	106,143744
12:18:30	106,196992	108,146688	109,232128	106,184704	106,217472

Tabela 14 – Escrita em disco no cenário com *Hyperledger Sawtooth*

Time	Drone 1	Drone 2	Drone 3	Drone 4	Drone 5
12:30:00	0	0	0	0	0
12:30:30	0	0	0	0	0
12:31:00	0	0	0	0	0
12:31:30	0	0	0	0	0
12:32:00	0	0	0	0	0
12:32:30	0	0	0	0	0
12:33:00	0	0	0	0	0
12:33:30	0	0	0	0	0
12:34:00	0	0	0	0	0
12:34:30	0	0	0	0	0
12:35:00	0	0	0	0	0
12:35:30	0	0	0	0	0
12:36:00	0	0	0	0	0
12:36:30	0	0	0	0	0
12:37:00	0	0	0	0	819
12:37:30	0	0	0	0	819
12:38:00	0	0	0	0	819
12:38:30	0	0	0	0	819
12:39:00	0	0	0	0	819
12:39:30	0	0	0	819	0
12:40:00	0	0	0	0	0
12:40:30	0	0	0	0	0
12:41:00	0	0	0	0	0
12:41:30	0	0	0	0	0
12:42:00	0	0	0	0	0
12:42:30	0	0	0	0	0
12:43:00	0	0	0	0	0
12:43:30	0	819	819	0	0
12:44:00	0	0	0	0	0
12:44:30	0	0	0	0	0
12:45:00	0	0	0	0	0
12:45:30	0	0	0	0	0
12:46:00	0	0	0	0	0

Tabela 15 – Escrita em disco no cenário com a interface REST desprotegida

Time	Drone 1	Drone 2	Drone 3	Drone 4	Drone 5
12:06:30	1638	1638	4915	0	3277
12:07:00	0	0	4096	1638	1638
12:07:30	1638	3277	5734	1638	1638
12:08:00	1638	1638	6554	3277	0
12:08:30	1638	1638	4915	1638	1638
12:09:00	1638	4915	1638	1638	1638
12:09:30	1638	4915	1638	1638	1638
12:10:00	1638	4915	1638	1638	1638
12:10:30	2458	4915	4915	2458	2458
12:11:00	2458	5734	2458	2458	2458
12:11:30	0	0	0	0	819
12:12:00	0	0	0	0	819
12:12:30	0	0	0	0	819
12:13:00	0	0	0	0	819
12:13:30	0	0	0	0	819
12:14:00	0	4915	2458	819	2458
12:14:30	0	1638	0	3277	2458
12:15:00	2458	5734	2458	3277	2458
12:15:30	2458	6554	4915	5734	0
12:16:00	2458	5734	2458	3277	2458
12:16:30	2458	2458	5734	2458	2458
12:17:00	2458	2458	5734	2458	2458
12:17:30	3277	3277	5734	3277	3277
12:18:00	3277	3277	9830	3277	3277
12:18:30	3277	3277	6554	3277	3277

ANEXO B – Artigo publicado

https://doi.org/10.1109/CCNC51644.2023.10060385

Referências

Aggarwal, S.; Kumar, N.; Tanwar, S. Blockchain-envisioned uav communication using 6g networks: Open issues, use cases, and future directions. *IEEE Internet of Things Journal*, v. 8, n. 7, p. 5416–5441, 4 2021. ISSN 2327-4662. 39, 42

AKHTER, A. et al. A secured privacy-preserving multi-level blockchain framework for cluster based vanet. *Sustainability (Switzerland)*, v. 13, n. 1, p. 1–25, 2021. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85099968102&doi=10.3390/su13010400&partnerID=40&md5=e8708199c819987887de7081f3292e0f>. 39, 41

AL-EMADI, S.; AL-MOHANNADI, A. Towards enhancement of network communication architectures and routing protocols for fanets: A survey. In: . [S.l.: s.n.], 2020. 23

ARIF, M. et al. Integration of 5g, vanets and blockchain technology. In: . [s.n.], 2020. p. 2007–2013. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85101229418&doi=10.1109/TrustCom50675.2020.00275&partnerID=40&md5=76158721ce03ec2b108962515b2b043c>. 39

ASNAFI, M.; DASTGHEIBIFARD, S. A review on potential applications of unmanned aerial vehicle for construction industry. *Sustainable Structures and Materials, An International Journal*, Sustainable Structures and Materials, An International Journal, v. 1, n. 2, p. 44–53, jul. 2018. ISSN 26164787, 26164779. Disponível em: <https://doi.org/10.26392/SSM.2018.01.02.044>. 23

BAHN, H.; CHO, H. Implications of nvm based storage on memory subsystem management. *Applied Sciences*, v. 10, p. 999, 02 2020. 68

CASTRO, M.; LISKOV, B. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems*, Association for Computing Machinery (ACM), v. 20, n. 4, p. 398–461, nov. 2002. Disponível em: <https://doi.org/10.1145/571637.571640>. 69

DASU, T.; KANZA, Y.; SRIVASTAVA, D. Geofences in the sky: Herding drones with blockchains and 5g. In: L., X. et al. (Ed.). *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. Association for Computing Machinery, 2018. p. 73–76. ISBN 9781450358897. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85058671966&doi=10.1145/3274895.3274914&partnerID=40&md5=b2c225bc1e1980166a08cc36cb8b964a>. 18

FERDOUS, M. S.; CHOWDHURY, M. J. M.; HOQUE, M. A. A survey of consensus algorithms in public blockchain systems for crypto-currencies. *Journal of Network and Computer Applications*, v. 182, p. 103035, 2021. ISSN 1084-8045. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804521000618>. 29, 30

FONTES, R. et al. Mininet-WiFi: emulating Software-Defined wireless networks. In: *2nd International Workshop on Management of SDN and NFV Systems, 2015(ManSDN/NFV 2015)*. Barcelona, Spain: [s.n.], 2015. 19, 47, 53

FONTES, R. dos R.; ROTHENBERG, C. E. Mininet-WiFi: Plataforma de emulação para redes sem fio definidas por software. In: *Anais Estendidos do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC Estendido 2019)*. Sociedade Brasileira de Computação - SBC, 2019. Disponível em: <https://doi.org/10.5753/sbrc_estendido.2019.7788>. 47

FU, X.; WANG, H.; SHI, P. A survey of blockchain consensus algorithms: mechanism, design and applications. *Science China Information Sciences*, Springer Science and Business Media LLC, v. 64, n. 2, nov. 2020. Disponível em: <https://doi.org/10.1007/s11432-019-2790-1>. 28, 29, 30

GARCíA-PEñALVO, F. J. Mapping sistemáticos de literatura. caso práctico de planificación usando parsifal. Zenodo, 2017. Disponível em: <https://zenodo.org/record/1069690>. 32

GUPTA, R. et al. Blockchain-envisioned softwarized multi-swarming uavs to tackle covid-i9 situations. *IEEE Network*, Institute of Electrical and Electronics Engineers Inc., v. 35, n. 2, p. 160–167, 2021. ISSN 08908044. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85090467651&doi=10.1109/MNET.011.2000439&partnerID=40&md5=311a2137e3f72aa77eed0851316f4684>. 39, 40

GUPTA, R. et al. Blockchain-based data dissemination scheme for 5g-enabled softwarized uav networks. *IEEE Transactions on Green Communications and Networking*, Institute of Electrical and Electronics Engineers Inc., v. 5, n. 4, p. 1712–1721, 2021. ISSN 24732400. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85114721989&doi=10.1109/TGCN.2021.3111529&partnerID=40&md5=48aab1ca3250599480b1676c4c0c1e4f>. 18

GUPTA, R. et al. Vahak: A blockchain-based outdoor delivery scheme using uav for healthcare 4.0 services. In: . Institute of Electrical and Electronics Engineers Inc., 2020. p. 255–260. ISBN 9781728186955. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85091514555&doi=10.1109/INFOCOMWKSHPS50562.2020.9162738&partnerID=40&md5=df2d9a6b72dc82bd689c09cc53dabe7d>. 39, 40

HINTZBERGEN, J. *Fundamentos de Segurança da Informação: Com base na ISO 27001 e na ISO 27002*. [S.l.: s.n.], 2018. 24

HOLYBRO. *Px4 Vision*. 2022. Disponível em: <https://holybro.com/products/px4-vision-dev-kit-v1-5>. 56

HU, N. et al. Building agile and resilient uav networks based on sdn and blockchain. *IEEE Network*, Institute of Electrical and Electronics Engineers Inc., v. 35, n. 1, p. 57–63, 2021. ISSN 08908044. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85101133761&doi=10.1109/MNET.011.2000176&partnerID=40&md5=2b698b295eb9ff985f3a666cb678d49e>. 46

IBRAHIM, M. S. M.; SUPERVISOR, N. B. O. *Building a Classification Model for Diseases Discovery from Tweets*. Tese (Thesis) — Sudan University of Science & Technology, dez. 2018. Disponível em: <http://repository.sustech.edu//handle/123456789/23391>. 36

INTEL. *Visão Geral do Intel® aero ready to fly drone*. 2019. Disponível em: <https://www.intel.com.br/content/www/br/pt/support/articles/000023271/drones/development-drones.html>. 56

JACOBSEN, R. H.; MARANDI, A. Security threats analysis of the unmanned aerial vehicle system. In: *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*. IEEE, 2021. Disponível em: <https://doi.org/10.1109/milcom52596.2021.9652900>. 10, 25, 26

Jameel, F. et al. Efficient mining cluster selection for blockchain-based cellular v2x communications. *IEEE Transactions on Intelligent Transportation Systems*, p. 1–9, 2020. ISSN 1558-0016. 39, 40

JOSHI, A.; SHARDA, N. Non-repudiation in electronic commerce. *Journal of Organizational Computing and Electronic Commerce*, Taylor & Francis, v. 11, n. 3, p. 161–173, 2001. 24

KALININ, M. O.; KRUNDYSHEV, V. M.; SEMIANOV, P. V. Architectures for building secure vehicular networks based on sdn technology. *Automatic control and computer sciences*, Pleiades Publishing, Moscow, v. 51, n. 8, p. 907–914, 2017. ISSN 0146-4116. 53

KHAN, M.; HARTOG, F. den; HU, J. A survey and ontology of blockchain consensus algorithms for resource-constrained iot systems. *Sensors*, v. 22, n. 21, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/21/8188>. 29

KUMARI, A. et al. A taxonomy of blockchain-enabled softwarization for secure uav network. *Computer Communications*, Elsevier B.V., v. 161, p. 304–323, 2020. ISSN 01403664. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85089136074&doi=10.1016/j.comcom.2020.07.042&partnerID=40&md5=b4afd68677b55282c44adf8bc4f5fc8f>. 18, 39, 42, 43, 46

LANG, J.-P. 2007. Disponível em: <https://www.open-mesh.org/projects/batman-adv/wiki/Doc-overview>. 56

LANTZ, B.; HELLER, B.; MCKEOWN, N. A network in a laptop. *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks - Hotnets '10*, 2010. 47

LI, W. et al. P-cft: A privacy-preserving and crash fault tolerant consensus algorithm for permissioned blockchains. In: *2021 4th International Conference on Hot Information-Centric Networking (HotICN)*. [S.l.: s.n.], 2021. p. 26–31. 19

Li, W. et al. Blockchain-based data security for artificial intelligence applications in 6g networks. *IEEE Network*, v. 34, n. 6, p. 31–37, 11 2020. ISSN 1558-156X. 39, 42

LOKE, S. W. The internet of flying-things: Opportunities and challenges with airborne fog computing and mobile cloud in the clouds. *CoRR*, abs/1507.04492, 2015. Disponível em: <http://arxiv.org/abs/1507.04492>. 22, 23

MEHTA, P.; GUPTA, R.; TANWAR, S. Blockchain envisioned uav networks: Challenges, solutions, and comparisons. *Computer Communications*, Elsevier B.V., v. 151, p. 518–538, 2020. ISSN 01403664. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85078564989&doi=10.1016/j.comcom.2020.01.023&partnerID=40&md5=8f0b687db72fc4015fbb3a415d87802d>. 39, 42, 44, 46

MEIJERS, J. et al. Blockchain for v2x: Applications and architectures. *IEEE Open Journal of Vehicular Technology*, v. 3, p. 193–209, 2022. 21, 29, 30, 31

MOHAMMED, A. et al. Deep reinforcement learning for computation offloading and resource allocation in blockchain-based multi-uav-enabled mobile edge computing. In: . Institute of Electrical and Electronics Engineers Inc., 2020. p. 295–299. ISBN 9781665405058. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85100042982&doi=10.1109/ICCWAMTIP51612.2020.9317445&partnerID=40&md5=8a8e35c0882f48450cf974698c0201e6>. 38, 39

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. maio 2009. Disponível em: <http://www.bitcoin.org/bitcoin.pdf>. 29

NOFER, M. et al. Blockchain. *Business & Information Systems Engineering*, Springer Science and Business Media LLC, v. 59, n. 3, p. 183–187, mar. 2017. Disponível em: <https://doi.org/10.1007/s12599-017-0467-3>. 29

NOOR, F. et al. A review on communications perspective of flying ad-hoc networks: Key enabling wireless technologies, applications, challenges and open research topics. *Drones*, MDPI AG, v. 4, n. 4, p. 1–14, 2020. ISSN 2504446X. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85092523809&doi=10.3390/drones4040065&partnerID=40&md5=f009a1e25eb14d51f04901210fb92cc6>. 18, 39, 44, 47

NVIDIA. *Jetson modules*. 2021. Disponível em: <https://developer.nvidia.com/embedded/jetson-modules>. 56

ONGARO, D.; OUSTERHOUT, J. In search of an understandable consensus algorithm. In: *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*. USA: USENIX Association, 2014. (USENIX ATC'14), p. 305–320. ISBN 9781931971102. 18

PASANDIDEH, F. et al. A review of flying ad hoc networks: Key characteristics, applications, and wireless technologies. *Remote Sensing*, v. 14, n. 18, 2022. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/14/18/4459>. 22

PEUSTER, M.; KARL, H.; ROSSEM, S. van. Medicine: Rapid prototyping of production-ready network services in multi-pop environments. In: *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. [S.l.: s.n.], 2016. p. 148–153. 47, 54

PI, R. *Buy A raspberry pi 4 model B*. 2019. Disponível em: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>. 56

PRAVEEN, G. et al. Blockchain for 5g: A prelude to future telecommunication. *IEEE Network*, v. 34, n. 6, p. 106–113, 2020. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85084238495&doi=10.1109/MNET.001.2000005&partnerID=40&md5=7a8a61687e50262da3317d66402934be>. 39

RAHMADIKA, S.; LEE, K.; RHEE, K.-H. Blockchain-enabled 5g autonomous vehicular networks. In: . [s.n.], 2019. p. 275–280. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85075913533&doi=10.1109/ICSECC.2019.8907054&partnerID=40&md5=943cebf8e9f2ddaed6b0da0c7218d54c>. 39, 40

RAHMAN, M. et al. Ioev-chain: A 5g-based secure inter-connected mobility framework for the internet of electric vehicles. *IEEE Network*, v. 34, n. 5, p. 190–197, 2020. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85089452668&doi=10.1109/MNET.001.1900597&partnerID=40&md5=b1e8f22cc8da33f92fdb15e777892248>. 39

REEBADIYA, D. et al. Blockchain-based secure and intelligent sensing scheme for autonomous vehicles activity tracking beyond 5g networks. *Peer-to-Peer Networking and Applications*, 2021. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85101585513&doi=10.1007/s12083-021-01073-x&partnerID=40&md5=2006a014b13d1969d26bda6517ef00cb>. 38, 39

RIDHAWI, I. et al. Enabling intelligent iocv services at the edge for 5g networks and beyond. *IEEE Transactions on Intelligent Transportation Systems*, 2021. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85100852092&doi=10.1109/TITS.2021.3053095&partnerID=40&md5=93e067dd09b6d094a5fea61f761511d0>. 39, 42

SAAD, W.; BENNIS, M.; CHEN, M. A vision of 6g wireless systems: Applications, trends, technologies, and open research problems. *IEEE Network*, v. 34, n. 3, p. 134–142, 2020. 23

SALAMH, F.; KARABIYIK, U.; ROGERS, M. A constructive direst security threat modeling for drone as a service. *The Journal of Digital Forensics, Security and Law*, Embry-Riddle Aeronautical University/Hunt Library, 2021. Disponível em: <http://dx.doi.org/10.15394/jdfsl.2021.1695>. 22, 24, 25

SALEH, S. N.; FATHY, C. A novel deep-learning model for remote driver monitoring in sdn-based internet of autonomous vehicles using 5g technologies. *Applied Sciences*, v. 13, n. 2, 2023. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/13/2/875>. 53

SEELEY, L. *Introduction to Sawtooth PBFT*. 2019. Disponível em: <https://www.hyperledger.org/blog/2019/02/13/introduction-to-sawtooth-pbft>. 49

SHRESTHA, R. et al. Evolution of v2x communication and integration of blockchain for security enhancements. *Electronics (Switzerland)*, v. 9, n. 9, p. 1–33, 2020. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85089700491&doi=10.3390/electronics9091338&partnerID=40&md5=e957f8c9d2a2cec6724e46190c690647>. 39, 43

SYSTEMS, O. *Ti am335x system-in-package - octavo - osd335x arm A8, 1GB DDR3*. 2022. Disponível em: <https://octavosystems.com/octavo_products/osd335x/>. 56

TSAO, K.-Y.; GIRDLER, T.; VASSILAKIS, V. G. A survey of cyber security threats and solutions for uav communications and flying ad-hoc networks. *Ad hoc networks*, Elsevier B.V, v. 133, p. 102894, 2022. ISSN 1570-8705. 10, 21, 22, 27, 28

VALAVANIS, K. *Advances in unmanned aerial vehicles: State of the art and the road to autonomy*. [S.l.]: Springer, 2007. 21

VOSVIEWER. 2010. Disponível em: <https://www.vosviewer.com/>. 36

WANG, C. et al. B-tsca: Blockchain assisted trustworthiness scalable computation for v2i authentication in vanets. *IEEE Transactions on Emerging Topics in Computing*, 2020. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85081579686&doi=10.1109/TETC.2020.2978866&partnerID=40&md5=adcbb29a5c759d8958d08fe07a21665f>. 39, 41

WANG, J. et al. Blockchain enabled verification for cellular-connected unmanned aircraft system networking. *Future Generation Computer Systems*, Elsevier B.V., v. 123, p. 233–244, 2021. ISSN 0167739X. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85106301313&doi=10.1016/j.future.2021.05.002&partnerID=40&md5=0dcdf34c9de15876cf0a2f17e241502a>. 18, 46

WANG, J. et al. Lightweight blockchain assisted secure routing of swarm uas networking. *Computer Communications*, Elsevier B.V., v. 165, p. 131–140, 2021. ISSN 01403664. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85097102731&doi=10.1016/j.comcom.2020.11.008&partnerID=40&md5=878e3e2808fe0cc025de64f222f7e5f7>. 40, 43

WANG, S. et al. Secure crowdsensing in 5g internet of vehicles: When deep reinforcement learning meets blockchain. *IEEE Consumer Electronics Magazine*, 2020. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85099094687&doi=10.1109/MCE.2020.3048238&partnerID=40&md5=020787ffc4d6542ca4fa10214c9c0107>. 39, 41

WENSLEY, J. et al. SIFT: Design and analysis of a fault-tolerant computer for aircraft control. *Proceedings of the IEEE*, Institute of Electrical and Electronics Engineers (IEEE), v. 66, n. 10, p. 1240–1255, 1978. Disponível em: <https://doi.org/10.1109/proc.1978.11114>. 18

WHEEB, A. H. Flying ad hoc networks (FANET): Performance evaluation of topology based routing protocols. *International Journal of Interactive Mobile Technologies (iJIM)*, International Association of Online Engineering (IAOE), v. 16, n. 04, p. 137–149, fev. 2022. Disponível em: <https://doi.org/10.3991/ijim.v16i04.28235>. 23

WOHLIN, C. et al. *Experimentation in Software Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. Disponível em: <http://link.springer.com/10.1007/978-3-642-29044-2>. 32, 33, 34