Um Mecanismo de Proteção Ciente de Vias Aéreas Contra Jamming Attacks para a Internet dos Drones

Alisson R. Svaigen[1,2], Azzedine Boukerche[1], Linnyer B. Ruiz[3], Antonio A. F. Loureiro[2]

[1]PARADISE Research Laboratory, University of Ottawa, Ottawa, Canadá

[2]Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brasil

[3]Grupo de Pesquisa Manna, Universidade Estadual de Maringá, Brasil

asvai015@uottawa.ca, boukerch@site.uottawa.ca,
lbruiz@uem.br, loureiro@dcc.ufmg.br

Abstract. *The Internet of Drones (IoD) is a network paradigm where drones will fly over well-defined airways. The Jamming Attack (JA) poses a severe risk to IoD, affecting the drone's trajectory. Although JA has been investigated in UAV-based networks, the current solutions consider the airspace free to fly, whereas IoD allows drones to fly over constrained airspace. To overcome this challenge, this work presents the design of an airway-aware protection mechanism against JA on the IoD. Our experiments pointed out that our solution is a proper protection mechanism to avoid drones suffering a JA since they fly over airways in a responsive time, decreasing the number of jammed drones. Hence, this study advances the protection mechanisms for IoD.*

Resumo. *A Internet dos Drones (IoD) é um paradigma de rede no qual os drones voam por vias aéreas bem definidas. O Jamming Attack (JA) representa um risco grave para a IoD, podendo afetar a trajetória dos drones. Embora o JA tenha sido investigado em redes de UAVs, as soluções existentes consideram o espaço aéreo livre para voo, enquanto na IoD os drones voam num espaço aéreo restrito. Para superar esse desafio, este trabalho apresenta o projeto de um mecanismo de proteção de vias aéreas contra JA em IoD. Nossos experimentos mostraram que nossa solução propõe um mecanismo de proteção adequado para evitar que drones sofram um JA, diminuindo o número de drones afetados. Assim, este trabalho avança os mecanismos de proteção para IoD.*

1. Introdução

Nos últimos anos, os drones vêm despertando o interesse da indústria e da academia. A sua utilização comercial tem aumentado, principalmente em um cenário de *Intelligent Transportation Systems* (ITS) (Menouar et al., 2017). Nesse caso, os drones disputarão o espaço aéreo e o canal de comunicação sem fio. Assim, será obrigatório ter uma rede para gerenciar e fornecer um ambiente justo e confiável, denominado *Internet of Drones* (IoD). Nessa direção, Gharibi et al. (2016) definiram uma arquitetura de rede em camadas para coordenar o acesso de drones ao espaço aéreo, incluindo serviços de navegação.

Uma das principais características de IoD é o conceito de vias aéreas, onde os drones podem voar. São semelhantes às estradas terrestres, possuindo diversas políticas de tráfego como limites de espaço aéreo e de velocidade. Num futuro próximo, espera-se um grande fluxo de drones no espaço aéreo e, assim, será imprescindível que existam espaços delimitados para voo. Assim, os drones devem seguir um plano de voo bem definido, compartilhado com os Provedores de Serviços de Zona (ZSPs) (Gharibi et al., 2016) que representam as autoridades de tráfego IoD.

IoD é um paradigma de rede móvel, mas há diferentes ataques que ameaçam sua segurança, incluindo o *Jamming Attack*, foco deste trabalho. Como IoD representa um poderoso ambiente assistido por outras redes, como celulares e veiculares, os drones são os principais alvos de adversários maliciosos, podendo sofrer diferentes tipos de ataques cibernéticos e físicos (Boccadoro et al., 2021). Esses ataques podem prejudicar a disponibilidade de um nó destino, impedindo-o de realizar suas tarefas.

Jamming Attack (JA) visa tornar um nó indisponível na rede através da interferência na comunicação. JA representa um risco grave no ambiente IoD pois pode tornar indisponível drones da rede. Esses veículos usam o espaço aéreo e se comunicam com outros nós na sua linha de visão (LoS). Assim, eles podem facilmente sofrer um JA. Se o drone for atacado, ele pode apresentar um comportamento incomum, como pousar em locais inadequados, levando a outros ataques, como o seu sequestro.

A relação entre JA e a trajetória do drone tem sido amplamente discutida em redes baseadas em veículos aéreos não tripulados (VANTs) (Wang et al., 2018; Xiao et al., 2018; Mowla et al., 2020; Duo et al., 2020; Gao et al., 2021; Wu et al., 2021), mas esses estudos consideram um espaço aéreo livre para voar. Por outro lado, o paradigma de IoD (Gharibi et al., 2016) demanda aos drones voarem em um espaço aéreo restrito. Até onde sabemos, não há nenhum estudo que investigue o impacto de JA em IoD. Como as vias aéreas limitam o espaço disponível para voo, as soluções atuais não podem ser aplicadas adequadamente. Esses problemas afetam o voo do drone, levando a uma reformulação do planejamento de suas rotas.

Este estudo propõe um mecanismo de proteção ciente de vias aéreas contra JA na IoD. Até onde sabemos, nossa proposta é o primeiro mecanismo de proteção contra JA que considera as restrições de vias aéreas e planejamento de seu uso. Portanto, avançamos no estado da arte dos mecanismos de proteção baseados em IoD, considerandos suas especificidades. Além disso, discutimos detalhadamente o JA no ambiente IoD, seus principais desafios e como ele difere do ataque em ambientes "*free-to-flight*".

Este trabalho está organizado como segue. A Seção 2 apresenta os conceitos e estudos relevantes nesta área. A Seção 3 apresenta o cenário da aplicação e a Seção 4 o mecanismo de proteção proposto. A Seção 5 apresenta a avaliação experimental e a Seção 6 os resultados obtidos. Finalmente, a Seção 7 apresenta nossas considerações finais.

2. Fundamentos

IoD adota o conceito de uma rede robusta e descentralizada para gerenciar e controlar o espaço aéreo, fornecendo serviços de navegação aos drones (Gharibi et al., 2016). IoD pode integrar-se a redes terrestres, ampliando a cobertura da Internet das Coisas (IoT), onde cada drone pode realizar tarefas de forma distribuída. Nesse cenário, Gharibi et al. (2016) propuseram uma arquitetura de controle de rede em camadas para tratar os requisitos de IoD. Em poucas palavras, os drones voam por rotas previamente definidas em vias aéreas existentes. A autoridade de rede ZSP gerencia e fornece todas as informações de navegação. O espaço aéreo é dividido em diferentes zonas, gerenciadas por pelo menos um ZSP. IoD pode ser modelada como um grafo $G = (V, E)$, onde V é o conjunto composto por *waypoints* tridimensionais que os drones devem alcançar com base em seus planos de voo e E é o conjunto de segmentos de via aérea, delimitado por dois *waypoints* $w_1, w_2 \in V$, tal que $\langle w_1, w_2 \rangle \in E$. Portanto, o planejamento da rota de um drone é um subgrafo $G' = (V', E') \subset G$.

O *Jamming Attack* (JA) é um tipo de *Distributed Denial of Service* (DDoS), o que

significa que o JA afeta a disponibilidade dos nós no ambiente. O JA ocorre quando um único adversário (ou um grupo deles) interfere na comunicação entre um conjunto de nós da rede através da inundação dos canais de comunicação da rede. Assim, qualquer nó pode usar a rede, prejudicando o serviço prestado (Mowla et al., 2020). Nos últimos anos, o JA tem sido investigado em VANTs como ameaça e como mecanismo de segurança. Especificamente, alguns estudos focaram na análise de como o JA interfere na trajetória do drone. A maioria desses trabalhos considera um *jammer* terrestre e estacionário. Wang et al. (2018) discutiram como o JA afeta a trajetória do drone. Eles formularam esse desafio como um problema de otimização. Diferentes estudos recentes expandiram esse contexto (Duo et al., 2020; Gao et al., 2021; Wu et al., 2021), explorando a qualidade do serviço (QoS) envolvida, considerando a vazão e o atraso da comunicação; usando transmissores secundários; e considerando um cenário onde drones coletam dados de uma rede de sensores sem fio (WSN), respectivamente. Alguns estudos investigaram o impacto de JA em outras redes móveis que integram drones como um nó de comunicação de retransmissão, como VANETs (Xiao et al., 2018). Recentemente, estratégias baseadas em aprendizado de máquina foram consideradas para mecanismos *antijamming* inteligentes para redes de drones (Sedjelmaci et al., 2017; Mowla et al., 2020), propondo uma abordagem baseada em aprendizado por reforço e um mecanismo distribuído baseado em aprendizado federado e por reforço, respectivamente.

Vale ressaltar que JA tem sido amplamente investigado em VANTs. No entanto, esses estudos assumem que os drones podem voar livremente sobre o espaço aéreo e, consequentemente, regiões livres da ação de JA podem ser alcançadas. Por outro lado, drones têm espaço aéreo limitado para voar em IoD. Visto que uma via aérea é afetada por um JA se não possuir um segmento livre do ataque, o planejamento de trajetória do drone deve ser reformulado. Também espera-se que uma determinada via aérea tenha um fluxo de tráfego constante em um ambiente real. Assim, a identificação de uma via aérea comprometida leva a um novo plano de voo de todos os drones que voarão por essa via aérea. Um sistema de gerenciamento deve lidar com esses problemas e se comunicar com os drones afetados rapidamente. Esses aspectos representam sérios riscos para a disponibilidade de IoD, sendo desafios em aberto.

3. Cenário da aplicação

No ambiente IoD, um JA tem o potencial de tolher o canal de comunicação e, assim, influenciar a mobilidade e o planejamento da trajetória do drone (Vadlamani et al., 2016). A Figura 1 ilustra este conceito, onde as vias aéreas são representadas pelos segmentos cinza. Um adversário emite um sinal de *jamming* (ondas vermelhas) comprometendo o espaço aéreo de um segmento de via aérea (laranja). Todos os drones que atravessam a região afetada não poderão se comunicar com nenhum nó, tornando-os indisponíveis para a rede (drones vermelhos). Assim, a rede IoD precisa detectar essa situação e refor-

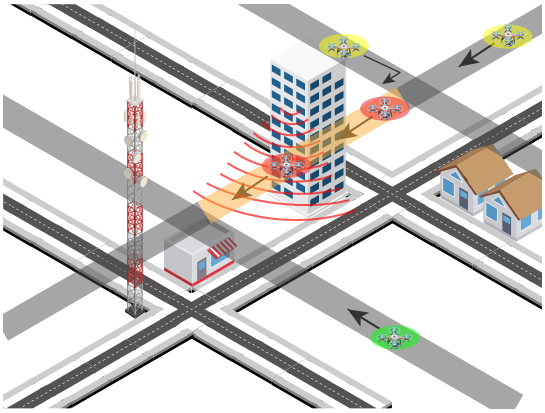

Figura 1. Ataque de *jamming* em um ambiente de IoD

mular o planejamento de trajetória dos drones que sobrevoarão a via aérea comprometida (drones amarelos). Os drones que não são afetados pelo JA podem seguir seu planejamento de rota atual (drone verde), sem perda de desempenho ou disponibilidade.

O cenário de aplicação é definido da seguinte maneira: um conjunto $\mathcal{D}$ de drones que sobrevoam as vias aéreas, um conjunto $\mathcal{P}$ de planos de voos definidos no sistema IoD, um conjunto $\mathcal{Z}$ de ZSPs que gerenciam o espaço aéreo IoD, um canal λ de comunicação IoD, e um limiar de interferência τ no canal de comunicação. Consideramos a existência de múltiplos adversários estacionários $\mathcal{A}$ distribuídos pelo ambiente. Também assumimos que um determinado adversário sempre realizará o ataque pela melhor configuração considerando seus objetivos. Um adversário pode realizar ataques de curto e longo prazos. Um drone afetado por um ataque pode perder a comunicação com o ZSP e deixar de receber o sinal GPS. Nesse caso, o drone geralmente paira e depois pousa no solo, o que facilita um sequestro proposital ou até mesmo um roubo por parte do atacante.

A ocorrência de JA no ambiente IoD pode ser formulada como uma função geral F que calcula a interferência causada pelos atacantes em λ em um determinado momento t. Um JA ocorre quando $F \geq \tau$, indicando que a comunicação entre os drones e/ou ZSPs com a rede IoD está severamente prejudicada. Se F indicar que há drones afetados por um JA, as vias aéreas das rotas desses drones podem estar comprometidas, levando os ZSPs a evitarem, em tempo real, que outros drones passem por essas regiões comprometidas, o que pode levar a uma reformulação de trajetórias. Para superar esses problemas, propomos um mecanismo de proteção, que é apresentado abaixo.

4. Mecanismo de proteção contra *Jamming Attacks* em IoD

Nossa solução propõe proteger a IoD de JAs e permitir que os ZSPs tomem contramedidas para evitá-los, desde a sua detecção até a potencial reformulação do planejamento de voo do drone. Este mecanismo é um sistema distribuído onde drones e ZSPs cooperam para evitar esses problemas, tendo quatro etapas principais: (i) o mecanismo tenta detectar a ocorrência de JA em um ou mais drones baseados na falta de comunicação com o ZSP; (ii) verificar as vias aéreas afetadas para evitar que outros drones passem por essa região; (iii) verificar quais drones passarão por essas vias e definir estratégias para alterar essas rotas a um custo mínimo; e (iv) enviar as novas rotas para os drones considerando suas posições atuais e os ZSPs associados. A seguir, discutiremos os três primeiros passos em detalhes, apresentando como o problema é modelado em nosso mecanismo e os algoritmos correspondentes que os definem formalmente. Como o último passo trata de uma transmissão de dados tradicional, ele não será discutido em detalhes.

4.1. Detecção de ataque de interferência (JA)

Em redes móveis sem fio baseadas em radiofrequência, JA afeta a qualidade do sinal de comunicação medida pelo RSSI (Wang et al., 2018). Do ponto de vista de IoD, quanto menor a taxa de transferência alcançável, maior a taxa de erro de bits transmitidos. Assim, funções básicas, como atualização da posição do drone, não podem ser executadas. Em nossa abordagem, a atualização da localização do drone segue a estratégia de melhor esforço, i.e., um determinado drone $d \in \mathcal{D}$ apenas envia uma mensagem atualizando sua localização e não espera nenhuma confirmação de um ZSP $z \in \mathcal{Z}$. Portanto, z é o nó responsável por identificar um JA sobre d. No entanto, outros fatores podem levar a uma perda de comunicação entre d e z como uma falha do hardware de d. Como esses fatores podem representar casos falso positivos, a detecção de JA não pode ser baseada na falta de atualização da localização de um único drone. Assim, dada a última posição de d, o

mecanismo também armazena as informações de RSSI do último d de tal forma que é possível inferir se o drone estava entrando na região de JA pouco antes de não atualizar mais sua posição. Abaixo, apresentamos nossa abordagem de detecção de JA.

Vamos considerar que em cada atualização de localização, um drone d envia dois dados adicionais $d.I$: o tempo previsto para a próxima atualização de localização $d.I.t_{next}$, e um contador de verificação $d.I.c$ cujo valor inicial é 0. Temos ainda três estruturas de dados auxiliares: uma fila de prioridade $\mathcal{D}_{updt} \subset \mathcal{D}$ que armazena a última atualização recebida de cada drone na região de z, ordenada pela mais antiga; um hashmap $\mathcal{D}_{RSSI} \subset \mathcal{D}$ que armazena um registro dos k RSSIs mais recentes das atualizações anteriores de cada drone por meio de um *buffer* circular; e um conjunto $\mathcal{L}_{susp}$ que armazena locais suspeitos onde um invasor pode estar. Quando d envia uma mensagem de atualização de localização para z, as tuplas de d em $\mathcal{D}_{updt}$ e $\mathcal{D}_{RSSI}$ são atualizadas.

O Algoritmo 1 descreve o monitoramento recorrente sobre a atualização do drone que um determinado ZSP z realiza. Além das estruturas de dados mencionadas, ele requer como entrada um valor limite γ que indica o número de verificações sem que seja necessário um drone atualizar sua posição. Inicialmente é tomado o tempo do sistema atual (Linha 1) e uma fila de prioridade auxiliar $\mathcal{D}_{counters}$, inicializada vazia (Linha 2), que armazena os drones que não atualizaram suas posições no tempo esperado. As informações relacionadas ao drone d com a previsão de tempo mais cedo são extraídas de $\mathcal{D}_{updt}$ (Linha 3) para verificar se o horário da próxima atualização do local já passou . Esse processo continua até que um drone d extraído tenha um tempo de previsão maior que o tempo atual do sistema, indicando que os drones restantes são aqueles em que ainda não alcançaram a hora de sua próxima atualização (Linhas 4–12). Se o tempo de previsão da próxima atualização de d já passou, é necessário verificar se $d.I.c$ atinge o limite γ (Linha 5). Em caso afirmativo, d não atualiza sua localização no último monitoramento de γ, indicando que um possível JA pode estar em andamento. Para verificar essa situação, são extraídos os RSSI das últimas mensagens de d (Linha 6). O algoritmo também prevê uma localização l, que indica onde d estaria se tivesse atualizado (Linha 7), e a detecção JA é feita (Linha 8). Por outro lado, o atraso na atualização pode ocorrer devido a outros problemas, diferentes de um JA. Nesse caso, o contador é incrementado e os dados de $d's$ são adicionados à fila auxiliar $\mathcal{D}_{counters}$ (Linhas 9-11) e o drone seguinte com o carimbo de data/hora mais antigo é extraído (Linha 12). Quando todas as atualizações atrasadas foram processadas, os drones com um novo contador $d.I.c$ são adicionados à fila de prioridade principal D_{updt} (Linha 13), mantendo o gerenciamento do z conciso.

Quando o Algoritmo 1 identifica a não atualização de posição de um drone d em um intervalo de tempo γ, o ZSP tenta detectar se há um JA em andamento, como descrito no Algoritmo 2. A entrada desse algoritmo requer a última localização atualizada l de d, o histórico RSSI das últimas mensagens de d e a lista dos locais suspeitos atuais $\mathcal{L}_{susp}$. O JA pode ser detectado de duas formas: (i) caso a localização de d esteja próxima a uma localização suspeita anterior; ou (ii) a curva RSSI indica falta de comunicação. Em (i), se a distância entre o local suspeito l_{susp} e o local do drone l (Linhas 1-5) for menor ou igual a um limite, inferimos que a região apresenta uma falta de comunicação recorrente (Linhas 2-3), o que indica a presença de um JA. Assim, o ZSP delega ao sistema de nuvem IoD a tarefa de calcular a região de risco (Linha 4). Se l não corresponder a nenhum local suspeito anterior, tratamos os valores RSSI de d como uma série temporal e analisamos a curva de tendência. Caso indique uma falta de comunicação, inferimos que o drone ao se

Algorithm 1: ZSP-Check-Drone-Updates

Input : $\mathcal{D}_{updt}, \mathcal{D}_{RSSI}, \mathcal{L}_{susp}, \gamma$

1 $t_{sys} \leftarrow$ horário atual do sistema
2 $\mathcal{D}_{counters} \leftarrow \emptyset$
3 $d \leftarrow$ desenfileirar($\mathcal{D}_{updt}$)
4 **while** $d.I.t_{next} < t_{sys}$ **do**
5 **if** $d.I.c = \gamma$ **then**
6 $d_{RSSI} \leftarrow$ obtenha o *buffer* circular de d a partir de $\mathcal{D}_{RSSI}$
7 $l \leftarrow$ predictLocation($d.l, d.I.t_{next}$)
8 ZSP-JA-Detect($l, d_{RSSI}, \mathcal{L}_{susp}$)
9 **else**
10 $d.I.c \leftarrow d.I.c + 1$
11 $\mathcal{D}_{counters} \leftarrow \mathcal{D}_{counters} \cup \{d\}$
12 $d \leftarrow$dequeue($\mathcal{D}_{updt}$)
13 $\mathcal{D}_{updt} \leftarrow \mathcal{D}_{updt} \cup \mathcal{D}_{counters}$

aproximar de l, seu poder de comunicação diminuiu, o que representa um JA (Linhas 6-7). Em alguns casos, pode ocorrer um falso negativo, onde o JA não pode ser detectado com base nos dois casos. No entanto, isso não significa que não há ataque. Assim, l é adicionado como local suspeito para outros casos de detecção (Linhas 8-9).

Algorithm 2: ZSP-JA-Detect

Input : $l, RSSI, \mathcal{L}_{susp}$

1 **foreach** $l_{susp} \in \mathcal{L}_{susp}$ **do**
2 $dist \leftarrow$ calcDistance(l, l_{susp})
3 **if** $dist \leq dist_{max}$ **then**
4 Cloud-Calc-Hazard-Region(l)
5 **return**
6 **if** *a série temporal expressa por RSSI representa uma curva tendendo a uma perda de comunicação* **then**
7 Cloud-Calc-Hazard-Region(l)
8 **else**
9 $\mathcal{L}_{susp} \leftarrow \mathcal{L}_{susp} \cup \{l\}$

4.2. Região de perigo

Após o mecanismo detectar um JA em l, deve-se isolar essa região, chamada *Hazard Region* (HR), para que outros drones evitem voar nas proximidades. Dado que l representa um único ponto em um sistema de navegação mais amplo, a delimitação precisa de HR é um desafio atual. Considerando a arquitetura IoD em camadas (Gharibi et al., 2016), os drones devem voar por vias aéreas definidas no grafo subjacente. Assim, l faz parte de um segmento de linha aérea delimitado por $\langle w_1, w_2 \rangle \in G.E$, tal que o vetor de direção é $w_1\vec{w}_2$. Para delimitar a HR, o sistema considera três casos distintos, descritos a seguir:

1. $\langle w_1, w_2 \rangle \in G.E$ onde l deve fazer parte da HR;
2. $\forall \langle u_1, u_2 \rangle \in G.E$, se $\langle u_1, u_2 \rangle$ estiver dentro de uma região esférica de risco com raio r das coordenadas geográficas de l, então deve fazer parte da HR;

3. $\forall\langle u_1, u_2\rangle \in$ HR, $\forall\langle v, u_1\rangle \in G.E$, se o nó v tiver u_1 como o único nó adjacente, então $\langle v, u_1\rangle$ deve fazer parte da HR. Ou seja, segmentos de via aérea que atingem um segmento já acometido e não possuem outro segmento de via aérea a seguir, também devem fazer parte da HR;

O Algoritmo 3 define esses casos, onde é necessária a localização prevista do drone l, e gera como saída um subgrafo $G_{HR} \subset G$ que representa a HR. Assumimos que o sistema em nuvem possui estruturas de dados apropriadas para representar $G.E$.

Algorithm 3: Cloud-Calc-Hazard-Region

Input : l
Output: G_{HR}

1 $G_{HR} \leftarrow \emptyset$
2 **foreach** $\langle w_1, w_2\rangle \in G.E$ **do**
3 $dist \leftarrow$ calcule a distância de l para o segmento de reta $\langle w_1, w_2\rangle$
4 **if** $dist \leq r$ **then**
5 $G_{HR}.V \leftarrow G_{HR}.V \cup \{w_1, w_2\}$
6 $G_{HR}.E \leftarrow G_{HR}.E \cup \{\langle w_1, w_2\rangle\}$
7 $G_{rmnd} \leftarrow G - G_{HR}$
8 **while** $(\exists v \in G_{rmnd}.V, outdegree(v) = 0) \wedge (v \in G_{HR})$ **do**
9 **while** $\exists\ \langle u, v\rangle \in G_{rmnd}.E$ **do**
10 $G_{HR}.E \leftarrow G_{HR}.E \cup \{\langle u, v\rangle\}$
11 $G_{rmnd}.E \leftarrow G_{rmnd}.E - \{\langle u, v\rangle\}$
12 $G_{HR}.V \leftarrow G_{HR}.V \cup \{v\}$
13 $G_{rmnd}.V \leftarrow G_{rmnd}.V - \{v\}$
14 Cloud-Reformulate-PP(G_{HR})

Inicialmente, o subgrafo G_{HR} é inicializado vazio (Linha 1). A seguir, todas as arestas de G são visitadas (Linhas 2-6) para verificar se o segmento aéreo está próximo o suficiente de l considerando o raio r. Em caso afirmativo, o segmento é incluído no HR (Linhas 4-6). Como a distância entre l e sua via aérea é próxima de 0 (e menor que r), essa etapa trata os casos (1) e (2). Com a HR inicial estabelecida, o sistema verifica se ainda há segmentos de via aérea que não possuem outro segmento posterior – caso (3). Assim, definimos um subgrafo $G_{rmnd} \subset G$ cujas arestas ainda não pertencem à HR (Linha 7), o que significa que $G_{rmnd}.E \cap G_{HR}.E = \emptyset$. Se houver um segmento de via aérea $\langle u, v\rangle \in G_{rmnd}$ que vai para a HR através de outra via aérea posterior, isso significa que $\langle u, v\rangle$ atinge um nó $v \in G_{rmnd}.V$ cujo grau de saída é 0. Isso acontece porque os segmentos de via aérea afetados estão em G_{HR}. Um caso particular pode ocorrer quando a topologia de vias aéreas possui nós "sumidouros", que representam, por exemplo, uma garagem de uma empresa de drones. Para evitar este caso, é necessário verificar se o nó $v \in G_{HR}$ (Linhas 8-13).

Quando essas condições são satisfeitas, cada segmento de via aérea que vai para v é adicionado a G_{HR} e removido de G_{rmnd} (Linhas 9-11). No fim, a lista de incidência de v estará vazia, e v é adicionado a G_{HR} e removido de G_{rmnd} (Linhas 12-13). Quando não há mais nós com grau de saída 0, a HR é completamente descoberta e pode ser usada para identificar os drones que precisam de um novo planejamento aéreo (Linha 14).

4.3. Reformulação do planejamento de rota

O Algoritmo 4 descreve a reformulação do planejamento de rota na nuvem IoD. Esse algoritmo identifica se, dado um planejamento de trajetória $p \in \mathcal{P}$, existem vias aéreas na HR que o drone $p.d$ irá voar até o final de sua viagem. Um caminho que não atravesse a HR substitui a parte afetada em caso afirmativo.

Como entrada, o Algoritmo 4 recebe a representação do grafo HR, fornecida pelo Algoritmo 3. Para reformular os planejamentos dos voos, primeiro calculamos quais vias aéreas não são afetadas pela HR (Linha 1). Em seguida, o mecanismo verifica cada plano de voo $p \in \mathcal{P}$ para identificar se demandará uma reformulação (Linhas 2-27). Existem dois nós auxiliares u e v que representam o último ponto geográfico antes do caminho atual entrar na HR e o primeiro ponto geográfico após sair da HR, respectivamente. Ambos são inicializados com valores nulos (Linha 3). Da mesma forma, um caminho auxiliar p' registra o planejamento do caminho reformulado. Ele é inicializado com os mesmos atributos que p exceto o planejamento do caminho, que é dado até a via aérea atual que o drone está voando (Linha 4). Após, verificamos cada segmento de via aérea $e \in p.G'.E'$ desde a via aérea do drone atual até o último (Linhas 5-21). Se a via aérea e estiver na HR (Linhas 6-8) é necessário verificar se é o primeiro segmento dentro da HR, o que significa que u aponta para nulo. Neste caso, u recebe o nó de origem de e (Linhas 7-8).

Quando e não pertence à HR (Linhas 9-21) precisamos considerar duas situações diferentes em relação às arestas anteriores: elas estão fora da HR, o que significa que e apenas segue um "caminho HR livre"; ou parte do caminho anterior está na HR. No primeiro caso, e é apenas adicionado ao caminho em p' (Linhas 10-11). Caso contrário, o ponto de origem $e.w_1$ representa o primeiro ponto fora da HR, sendo associado a v (Linha 13). Portanto, dados os pontos u e v, o algoritmo de Dijkstra calcula um plano de voo mínimo sobre G_{avail} (Linha 14). Se houver um caminho disponível E_{new}, ele será adicionado ao novo caminho, bem como ao segmento atual e (Linhas 15-17). À medida que um segmento de caminho HR é processado e substituído, u e v são definidos como nulos (Linha 18). No entanto, se Dijkstra não retornar nenhum caminho, não há caminho alternativo para a área afetada. Neste caso, o sistema deve enviar uma mensagem de emergência para o drone $p.d$, que seguirá as orientações de sua empresa para informar um novo ponto final (Linhas 20-21).

Após processar os segmentos aéreos de p, o sistema verifica se o último segmento não está dentro da HR. Isso pode ser verificado através de v, que deve ser nulo. Se o segmento estiver dentro da HR (Linhas 23-24) não há caminho alternativo para terminar a viagem. Assim, o drone deve ser informado sobre o problema, o que também ocorre nas Linhas 20-21. Se o planejamento do caminho foi afetado em um ou mais segmentos (verificado através da comparação de arestas de p e p'), o caminho atual é substituído pelo novo e submetido ao ZSP mais próximo de pd, para informar o drone sobre a atualização (Linhas 25-27).

5. Avaliação Experimental

Realizamos uma avaliação experimental por meio de simulações para investigar os seguintes aspectos: (i) como o JA afeta a mobilidade e disponibilidade do drone no ambiente IoD; e (ii) como o mecanismo contribui para a superação desses desafios. Nesta seção, apresentamos o cenário avaliado, os parâmetros de simulação e as métricas.

5.1. Cenário Avaliado

Conforme discutido, até onde sabemos, nossa solução é o primeiro mecanismo de proteção com reconhecimento de vias aéreas contra JA no contexto de IoD. Assim, con-

Algorithm 4: Cloud-Reformulate-PP

Input : G_{HR}

$G_{avail} \leftarrow G - G_{HR}$
foreach $p \in \mathcal{P}$ **do**
  $u, v \leftarrow null$
  $p' \leftarrow \langle p.d, p.G'.E'_{\langle 0..p.airway.order \rangle}, p.airway \rangle$
  for $e \in p.G'.E'$ *tal que* $p.airway.order \leq e.order < |p.G'.E'|$ **do**
    if $e \in G_{HR}.E$ **then**
      if $u = null$ **then**
        $u \leftarrow e.w_1$
    else
      if $u = null$ **then**
        $p'.G'.E' \leftarrow p'.G'.E' \cup e$
      else
        $v \leftarrow e.w_1$
        $E_{new} \leftarrow$ Dijkstra(G_{avail}, u, v)
        if $E_{new} \neq \emptyset$ **then**
          $p'.G'.E' \leftarrow p'.G'.E' \cup E_{new}$
          $p'.G'.E' \leftarrow p'.G'.E' \cup e$
          $u, v \leftarrow null$
        else
          envie uma mensagem para $p.d$ informando que não há plano de vôo possível
          goto próximo $p \in \mathcal{P}$
  if $u \neq null$ **then**
    envie uma mensagem para $p.d$ informando que não há plano de vôo possível
    goto next $p \in \mathcal{P}$
  if $p'.G'.E' - p.G'.E' \neq \emptyset$ **then**
    $p \leftarrow p'$
    ZSP-Drone-PP-Updt(p)

sideramos três cenários para avaliar adequadamente o mecanismo proposto: **Free-JA**: neste cenário, não há JA em andamento. Assim, os drones se comunicarão e se movimentarão o máximo possível, representando uma linha de base "ótima" a ser alcançada; **JA**: uma entidade maliciosa executa um JA de longo prazo a partir do solo em uma rede que não possui nenhum mecanismo de proteção para evitar o ataque; **PM**: a rede IoD tem o mecanismo proposto para evitar JA.

As vias aéreas no contexto de IoD são o foco principal do nosso estudo. Assim, consideramos duas topologias de vias aéreas distintas, T_1 e T_2, cujos principais atributos estão listados na Tabela 1. Resumidamente, T_1 é uma infraestrutura robusta onde os drones possuem vias aéreas paralelas e diferentes caminhos para voar, projetado seguindo um trecho da estrutura de vias da ilha de Manhattan, NY. Por sua vez, T_2 é uma infraestrutura restrita, onde os drones possuem caminhos limitados para voar, com uma única altitude disponível. Essa topologia segue uma parte da estrutura de vias da cidade de San Francisco, CA.

Tabela 1. Descrição das topologias

Attr.	$\mathbf{T}_1$	$\mathbf{T}_2$
Tamanho da Região	$3 \times 3 \times 0.2\,\mathrm{km}^3$	$4 \times 2 \times 0.1\,\mathrm{km}^3$
#Altitudes disponíveis	3	1
#Nós de via aérea (total)	402	60
#Segmentos de via aérea (total)	997	98

5.2. Parâmetros de Simulação

Realizamos as simulações usando o IoDSim[1], um simulador integrado ao OMNET++ projetado para IoD. A parametrização é definida considerando um cenário realista para que um invasor realize o JA. Da mesma forma, o sistema distribuído pode detectar e agir contra o ataque. Com exceção do cenário livre de *jammers* avaliamos os cenários com um e dois *jammers*. Além disso, consideramos as duas topologias descritas. Cada simulação dura 15 minutos. Definimos 25 drones seguindo um padrão de mobilidade Gauss-Markov com velocidade variando de 5 a 10 m/s. Esse padrão é utilizado em outros trabalhos que consideram o voo de drones sobre vias aéreas, sendo mais realista do que outros modelos de mobilidade, como o linear (Svaigen et al., 2021, 2022). Espalhamos quatro ZSPs no solo, comunicando-se com os drones e um sistema de nuvem.

Em relação à comunicação, os protocolos e configuração de rádio são os mesmos para todos os nós. O mecanismo projetado considera UDP, AODV e CSMA/CA como protocolos de transporte, roteamento e MAC, respectivamente. O rádio possui modulação APSK, frequência de 2,4 GHz e potência de transmissão de 220 mW. O limite SNIR é definido como 4 dB. Para cada combinação do cenário $\times$ topologia $\times$ número de *jammers* (exceto Free-JA com *Jammers*), realizamos 30 experimentos com plano de voos e drones distintos. As posições dos *jammers* e ZSPs foram as mesmas para todas as replicações, definidas de maneira empírica para cobrir o cenário o máximo possível. Portanto, cada experimento reflete o cenário de mobilidade de um drone diferente nas posições dos mesmos atacantes. Os atacantes aplicam JAs contínuos durante todo o período de simulação. No total, foram realizadas 300 simulações. Assim, nossos resultados possuem um intervalo de confiança (IC) de 95%.

5.3. Métricas

Consideramos quatro métricas que medem o desempenho de voo e comunicação. Elas estão descritos a seguir:

- *Taxa de Drones Afetados (TDA)*: calcula a proporção de drones que tem seu canal de comunicação afetado por um JA, dificultando sua disponibilidade e mobilidade na rede IoD. TDA é definida da seguinte forma, onde d_{ja} representa um drone afetado pelo ataque: $\mathit{TDA} = (\sum d_{ja}) \,/\, |\mathcal{D}|$;
- *Taxa de Planos de Voo Afetados (TPVA)*: esta métrica indica o número de planejamentos de trajetos afetados comparado ao número total de trajetos. Inclui o caso em que o planejamento do voo precisa ser reformulado, pois possui segmentos de vias aéreas que pertencem à HR. No entanto, não inclui os casos em que o destino final de voo está numa HR. TPVA é definida da seguinte forma, onde p_a representa um caminho afetado: $\mathit{TPVA} = (\sum p_a) \,/\, |\mathcal{P}|$;

[1]Disponível em: `https://iodsim.manna.team`

- *Taxa de Aumento da Distância de Voo (TADV)*: como alguns planos de voo podem ser reformulados, os novos trajetos são potencialmente mais longos que os afetados. Assim, esta métrica mede a relação entre a distância total entre o primeiro e o novo plano de voo. TADV é definida da seguinte forma, onde p_a é o caminho afetado, p_n é o novo caminho e *dist*() é uma função que calcula a distância total de um determinado caminho: $TADV = dist(p_n) \,/\, dist(p_a)$;
- *Delay Fim-a-Fim de Reformulação de Plano de Voo (DRPV)*: mede quanto tempo um determinado drone recebe seu novo plano de voo desde que um JA é detectado na rede. DRPV é definido da seguinte forma, onde t_{JA} representa o instante da detecção de JA, e t_p é o instante quando o drone recebe o novo plano de voo completamente: $DRPV = t_p - t_{JA}$

6. Resultados e Análise

Esta seção apresenta os resultados das simulações experimentais. Discutimos os valores associados a cada métrica, descrevendo como o mecanismo de proteção contribui para um ambiente IoD mais seguro. Para os resultados associados a taxas, os erros de intervalo são menores que 1 unidade de medida. Para os resultados cuja unidade de medida é dada em segundos, os erros de intervalo são menores do que 1 décimo de segundo. Portanto, eles não estão representados nos gráficos.

6.1. Taxa de Drones Afetados (TDA)

Nossa solução é essencialmente um mecanismo de proteção para evitar que drones passem por uma região de JA. Como a TDA mede quantos drones estão bloqueados, é uma métrica adequada para avaliar o desempenho do mecanismo. A Figura 2.a mostra os resultados de TDA para cada topologia e o número de *jammers*, comparando os drones afetados nos cenários *JA* e *PM*. Em relação ao segundo cenário, o resultado é agrupado em duas classes. A primeira (barras azuis) representa o drone que é completamente tomado pelo JA, o que significa que o drone perdeu completamente a comunicação com a rede. O segundo conjunto (barras laranjas), por sua vez, representa os drones que não estão devidamente bloqueados, mas não conseguem completar seu voo, pois seu destino pertence à HR. Assim, embora o drone ainda tenha o seu canal de comunicação ativo, não pode continuar a voar para este destino. Nesse caso, a rede envia um alerta de emergência ao drone afetado e, em seguida, tomará sua própria decisão de como proceder, por exemplo, definindo e enviando ao ZSP um novo destino, fora da HR.

A TDA mostra que o mecanismo proposto pode evitar que os drones sofram um JA. Para os cenários avaliados, menos de 7% dos drones foram bloqueados. Isso indica que a rede, ao detectar um ataque, reage muito rapidamente e alerta os drones sobre o JA (discutido na Seção 6.4). Por outro lado, há um número significativo de drones afetados pelo ataque, exigindo que o destino final de uma viagem seja alterado. Além disso, a topologia das vias aéreas aliada ao número de *jammers* influenciam no sucesso do ataque. Quanto maior o número de *jammers*, maior o número de drones bloqueados. Ele é potencializado quando a topologia não oferece um número maior de caminhos opcionais, como ocorre com T_2. No entanto, as características da topologia não afetam a proteção fornecida.

6.2. Taxa de Planos de Voo Afetados (TPVA)

Uma vez detectado um JA, o sistema identifica a HR e verifica os trajetos a serem reformulados. A TPVA mede a proporção dessas reformulações considerando o número total de planos de voos processados durante as simulações. Os resultados estão sumarizados na Figura 2.b.

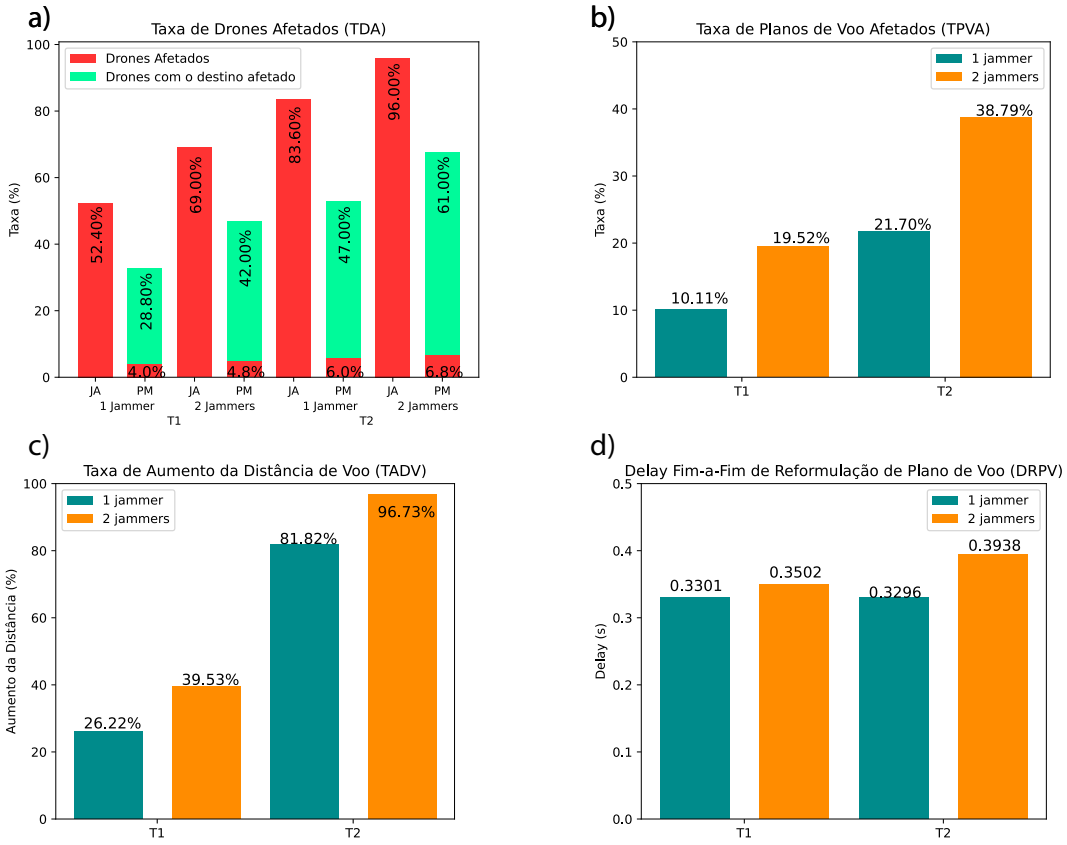

Figura 2. Resultados das métricas (a) TDA; (b) TPVA; (c) TADV; (d) DRPV

Os resultados apontam que o número de planos de voo reformulados aumenta de acordo com o aumento do número de *jammers*, o que também aumenta na topologia de via aérea restrita. Quase 40% dos planos de voo foram reformulados neste cenário. Com mais *jammers* no ambiente realizando ataques, a HR tende a ser mais ampla e, portanto, mais caminhos são afetados. A reformulação do plano de voos também pode causar sobrecarga na IoD. Uma vez feita a reformulação, o sistema precisa transmiti-la para o ZSP mais próximo do drone alvo. O ZSP, por sua vez, também transmitirá o novo plano de voo para o drone, o mais rápido possível.

6.3. Taxa de Aumento da Distância de Voo (TADV)

Além da sobrecarga potencial para transmitir a reformulação de rota, a distância maior de voo da nova rota é uma questão desafiadora que precisa ser avaliada. Considerando que o controle de tráfego IoD tenta formular um planejamento de trajeto da melhor forma possível, é provável que a nova rota terá uma distância maior do que o trajeto original. Avaliamos este aspecto através da métrica TADV cujos resultados estão mostrados na Figura 2.c.

Como ocorre nos resultados anteriores, quanto maior o número de *jammers* maior a distância do caminho reformulado. A topologia T_2, com a presença de dois *jammers*, atinge uma TADV superior a 95%, em média. Isso indica que o JA restringiu severamente a topologia, com poucos caminhos possíveis para os drones voarem. Nesse cenário, o drone precisa voar quase duas vezes o comprimento do caminho original. No entanto, os drones têm limitações SWaP (*Size, Weight, and Power*), o que significa que eles devem

economizar o máximo de energia possível, ocasionando um *tradeoff* no planejamento do serviço prestado.

6.4. Delay Fim-a-Fim de Reformulação de Plano de Voo (DRPV)

Considerando os passos descritos na Seção 4, a última etapa consiste em enviar o planejamento do caminho reformulado para o drone afetado. Esta tarefa envolve a comunicação de pelo menos dois nós: do sistema em nuvem para o ZSP mais próximo ao drone, e do ZSP ao drone, propriamente. Assim, o DRPV mede o atraso fim-a-fim desde o momento em que o JA é detectado até a última mensagem de atualização enviada a um determinado drone em relação ao seu novo plano de voo. A Figura 2.d apresenta esses resultados. Para todos os cenários, o atraso médio é menor que 0,4 segundos, o que representa um tempo de resposta aceitável para alertar os drones sobre a ameaça iminente de JA. Considerando a parametrização de velocidade do drone feita em nossos experimentos, um determinado drone teria voado no mínimo 2 metros e no máximo 4 metros. Em outras palavras, a menos que o drone esteja muito próximo de HR, ele será notificado a tempo sobre o JA.

Ao contrário das métricas anteriores, a topologia das vias aéreas e o número de bloqueadores não afetam o atraso fim-a-fim. Vale ressaltar que na nossa avaliação os nós possuem uma configuração de rádio adequada para se comunicar na rede, e a LoS melhora a cobertura e a taxa de entrega de mensagens. Outro fator que pode explicar a proximidade entre os atrasos obtidos é que as topologias abrangem um tamanho de região semelhante. Como os ZSPs são colocados em locais estratégicos para aumentar a cobertura, o tempo gasto para entregar as mensagens é menor.

6.5. Discussão Geral

O controle de tráfego do drone através de vias aéreas bem definidas é, sem dúvida, um aspecto fundamental da IoD. Com o constante crescimento dos serviços baseados em drones, é obrigatório que uma autoridade de rede gerencie e controle o voo do drone, impondo os seus limites. Embora o conceito de vias aéreas tenha várias vantagens, representa um risco para evitar um JA, conforme discutimos ao longo deste trabalho. Como os estudos do estado de arte não contemplam que os drones voem por um espaço aéreo restrito, nossa solução supera essa falta.

Os resultados obtidos indicam que o mecanismo proposto é adequado para a proteção de vias aéreas para combater JA em um tempo apropriado. Nossa solução diminui significativamente o número de drones afetados quando comparado a um ambiente sem mecanismos de proteção. Também podemos identificar que topologias de vias aéreas restritas representam um grande desafio ao mecanismo proposto. Podemos notar uma relação entre as características da topologia das vias aéreas e o número de *jammers* realizando JA distintos. Um número maior de *jammers* em uma topologia restrita prejudica significativamente o voo de drones cujo destino esteja dentro de uma região afetada. Como a HR cresce quando mais *jammers* realizam o ataque, as topologias restritas sofrem para fornecer caminhos disponíveis para o voo dos drones, aumentando o número de reformulações de planos de voo e, portanto, a distância que um drone deve percorrer para chegar ao seu destino. Assim, novas estratégias precisam ser projetadas para superar esses problemas.

7. Observações Finais

Este estudo propôs um mecanismo de proteção ciente de vias aéreas contra JA no ambiente IoD. Projetamos formalmente uma estrutura que vai desde a detecção de JA até a reformulação do plano de voo de um drone, quando afetado pelo ataque. O mecanismo proposto pode ser aplicado em ambientes IoD do mundo real, avançando o estado da arte

de mecanismos de proteção para IoD. Realizamos uma avaliação experimental por meio de simulações, considerando ambientes com diferentes topologias de vias aéreas. Nossa solução abordou a proteção adequada em todos os cenários, detectando e alertando drones sobre a ocorrência de um JA em tempo apropriado. Além disso, podemos destacar novos desafios neste campo: topologias de vias aéreas restritas tendem a dificultar a proteção fornecida, enquanto o número de *jammers* aumenta. Esse problema leva a um aumento da distância quando ocorre a reformulação do plano de voo de um drone.

Como direções futuras, planejamos explorar as oportunidades e investigar os desafios atuais, resumidos da seguinte forma: (i) projetar novas estratégias para lidar com drones cujo destino final pertence à região de risco (HR); (ii) aplicar o mecanismo em uma variedade de topologias para avaliar a proteção fornecida em um escopo mais amplo; (iii) estudar a relação entre o JA e a região de risco (HR) detectada; e (iv) aplicar e avaliar o mecanismo em ambientes móveis reais baseados em drones.

Referências

Boccadoro, P., Striccoli, D., & Grieco, L. A. (2021). An extensive survey on the internet of drones. *Ad Hoc Networks*, *122*, 102600.

Duo, B., Wu, Q., Yuan, X., & Zhang, R. (2020). Anti-jamming 3d trajectory design for uav-enabled wireless sensor networks under probabilistic los channel. *IEEE TVT*, *69*(12), 16288–16293.

Gao, Y., Wu, Y., Cui, Z., Yang, W., & Li, N. (2021). Anti-jamming trajectory and power design for cognitive uav communications. In *Iwcmc* (pp. 1370–1375).

Gharibi, M., Boutaba, R., & Waslander, S. L. (2016). Internet of drones. *IEEE Access*, *4*, 1148–1162.

Menouar, H., Guvenc, I., Akkaya, K., Uluagac, A. S., Kadri, A., & Tuncer, A. (2017). Uav-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, *55*(3), 22–28.

Mowla, N. I., Tran, N. H., Doh, I., & Chae, K. (2020). Afrl: Adaptive federated reinforcement learning for intelligent jamming defense in fanet. *Journal of Comm. and Net.*, *22*(3), 244–258.

Sedjelmaci, H., Senouci, S. M., & Ansari, N. (2017). A hierarchical detection and response system to enhance security against lethal cyber-attacks in uav networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *48*(9), 1594–1606.

Svaigen, A. R., Boukerche, A., Ruiz, L. B., & Loureiro, A. (2022). A topological dummy-based location privacy protection mechanism for the internet of drones. In *Ieee icc.* Seoul, Korea (South).

Svaigen, A. R., Boukerche, A., Ruiz, L. B., & Loureiro, A. A. (2021). Mixdrones: A mix zones-based location privacy protection mechanism for the internet of drones. In *Acm mswim'21* (pp. 181–188).

Vadlamani, S., Eksioglu, B., Medal, H., & Nandi, A. (2016). Jamming attacks on wireless networks: A taxonomic survey. *Inter. Jrnl. of Production Economics*, *172*, 76–94.

Wang, H., Chen, J., Ding, G., & Sun, J. (2018). Trajectory planning in uav communication with jamming. In *Wcsp* (pp. 1–6).

Wu, Y., Yang, W., Guan, X., & Wu, Q. (2021). Uav-enabled relay communication under malicious jamming: Joint trajectory and transmit power optimization. *IEEE TVT*.

Xiao, L., Lu, X., Xu, D., Tang, Y., Wang, L., & Zhuang, W. (2018). Uav relay in vanets against smart jamming with reinforcement learning. *IEEE TVT*, *67*(5), 4087–4097.