

GILSON EVANDRO JUST JUNIOR

**Percurso Dinâmico de VANTs Colaborativos  
com Restrições de Tempo no Contexto de  
Agricultura de Precisão**

Curitiba

2019

GILSON EVANDRO JUST JUNIOR

**Percurso Dinâmico de VANTs Colaborativos com  
Restrições de Tempo no Contexto de Agricultura de  
Precisão**

apresentada ao Programa de Pós-Graduação  
em Informática da Pontifícia Universidade Ca-  
tólica do Paraná como requisito parcial para  
obtenção do título de Mestre em Informática.

Pontifícia Universidade Católica do Paraná  
Programa de Pós-Graduação em Informática

Orientador: Marcelo Eduardo Pellenz  
Coorientador: Luiz Augusto de Paula Lima Junior

Curitiba  
2019

GILSON EVANDRO JUST JUNIOR

Percurso Dinâmico de VANTs Colaborativos com Restrições de Tempo no Contexto de Agricultura de Precisão

GILSON EVANDRO JUST JUNIOR. – Curitiba, 2019-

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

Orientador: Marcelo Eduardo Pellenz

Tese – Pontifícia Universidade Católica do Paraná  
Programa de Pós-Graduação em Informática, 2019.

1. VANTs. 2. Colaborativos. 3. UAANET. 4. SRP 5. Planejamento de Percurso I. Pellenz, Marcelo Eduardo.

II. Pontifícia Universidade Católica do Paraná. III. Programa de Pós-Graduação em Informática.

GILSON EVANDRO JUST JUNIOR

**Percurso Dinâmico de VANTs Colaborativos com  
Restrições de Tempo no Contexto de Agricultura de  
Precisão**

apresentada ao Programa de Pós-Graduação  
em Informática da Pontifícia Universidade Ca-  
tólica do Paraná como requisito parcial para  
obtenção do título de Mestre em Informática.

Trabalho aprovado. Curitiba, 19 de março de 2025:

---

**Marcelo Eduardo Pellenz**  
Orientador

---

**Luiz Augusto de Paula Lima Junior**  
Coorientador

---

**XXXX XXXXXX XXXX XXXXX**  
Convidado 1

---

**XXXX XXXXXX XXXX XXXXX**  
Convidado 2

Curitiba  
2019

# Resumo

VANTs - Veículos Aéreos Não-Tripulados - estão representando um mercado com potencial bilionário nos últimos anos devido a sua presença tanto no contexto civil, quanto no militar. Aplicações nas mais diversas áreas têm surgido nas últimas décadas com o intuito de resolver problemas de forma simples, e sem exposição de vidas humanas a riscos desnecessários. Diversos sistemas de automação têm surgido em função de VANTs colaborativos devido a sua grande flexibilidade ao criar e desfazer redes móveis, e na área de agricultura de precisão não é diferente. Serviços que antes um humano levava um dia inteiro para realizar, estão sendo realizados por VANTs em poucos minutos, e com muito mais eficiência e precisão. O presente projeto tem como objetivo a definição, implementação e análise de um modelo que vise uma plataforma de VANTs colaborativos, permitindo otimizar percursos dos VANTs no processo de coleta de dados de sensores estacionários em solo. A aplicação do modelo VRP - *Vehicle Routing Problem*, será proposta de forma a aprimorar os percursos dos VANTs dentro da janela de transmissão destes sensores, programada de forma dinâmica pelos VANTs através de aprendizado à cada passagem. Por meio do desenvolvimento de um simulador próprio para o modelo proposto, serão validadas as técnicas e comparados resultados em termos de eficiência com outras soluções possíveis encontradas na literatura.

**Palavras-chave:** VANTs, Colaborativos, UAANETs, SRP, Planejamento de Percurso.

# Abstract

UAVs - Unmanned Aerial Vehicles - are representing a market with billionaire potential in recent years due to their presence in both civil and military contexts. Applications in various areas have emerged in recent decades to solve simply problems, and without exposing human lives to unnecessary risks. Several automation systems have emerged from collaborative UAVs because of their great flexibility in creating and uncreating mobile networks, and in the area of precision agriculture is no different. Services that once a human took a full day to perform are being performed by UAVs within minutes, and with much more efficiency and precision. This project aims to define, implement and analyze a model that aims at a platform of collaborative UAVs, allowing to optimize UAV pathways in the process of data collection from ground stationary sensors. The application of the VRP - Vehicle Routing Problem model will be proposed in order to improve the UAV paths within the transmission window of these sensors, dynamically programmed by UAVs through learning at each pass. Through the development of a simulator for the proposed model, the techniques will be validated, and results compared in terms of efficiency with other possible solutions found in the literature.

**Keywords:** UAVs, Collaborative, UAANETs, SRP, Path Planning.

# Lista de ilustrações

Figura 1 – Agricultor realizando mediação do solo. . . . .	16
Figura 2 – (a) VANT tipo asas-fixas, (b) VANT tipo asas-rotativas . . . . .	21
Figura 3 – Modelo de comunicação ad-hoc. . . . .	24
Figura 4 – Transmissão do <i>streaming</i> de vídeo através da rede <i>ad-hoc</i> . . . . .	25
Figura 5 – Transmissão do <i>streaming</i> de vídeo através da rede <i>ad-hoc</i> . . . . .	25
Figura 6 – Atuação dos sensores na comunicação Veículo-Veículo (V2V). . . . .	28
Figura 7 – Modelo de comunicação VANT-ECT, e VANT-VANT. . . . .	29
Figura 8 – Comunicação VANT – VANT provê conectividade entre todos os elementos da rede. . . . .	30
Figura 9 – Problemas na comunicação centralizada. . . . .	32
Figura 10 – Comunicação em redes DTN. . . . .	36
Figura 11 – Modelo de mobilidade <i>Random WayPoint</i> . . . . .	37
Figura 12 – Sistema de subáreas criados para mapear a superfície. . . . .	39
Figura 13 – Tela Inicial do <i>UAV Simulator</i> . . . . .	47
Figura 14 – Funcionalidades do <i>UAV Simulator</i> . . . . .	48
Figura 15 – Funcionalidades do <i>UAV Simulator</i> . . . . .	49
Figura 16 – Funcionalidades do <i>UAV Simulator</i> . . . . .	51
Figura 17 – Painel de saída do <i>UAV Simulator</i> . . . . .	52
Figura 18 – Painel de visualização das ECs. . . . .	53
Figura 19 – Área de Simulação e Informações sobre Objetos. . . . .	54
Figura 20 – Proposta de otimização de percurso dos autores. . . . .	56
Figura 21 – Resultados para 10 sensores, em (KASHUBA V. I. NOVIKOV, 2015). . . . .	57
Figura 22 – Variação da Quantidade de Sensores em (KASHUBA V. I. NOVIKOV, 2015). . . . .	57
Figura 23 – Resultado do UAV Simulator sobreposto ao apresentado por Kashuba. . . . .	58
Figura 24 – Valores obtidos em (KASHUBA V. I. NOVIKOV, 2015), e em nosso simulador (UAVSIMULATOR, 2019). . . . .	59
Figura 25 – Valores obtidos em (KASHUBA V. I. NOVIKOV, 2015), e em nosso simulador (UAVSIMULATOR, 2019). . . . .	59
Figura 26 – Pseudocódigo referente ao percurso de Kashuba. . . . .	60
Figura 27 – Modelo matemático do <i>NearestPoint</i> . . . . .	61
Figura 28 – Exemplo de rota gerada pelo <i>NearestPoint</i> . . . . .	62
Figura 29 – <i>UAV Simulator</i> (UAVSIMULATOR, 2019) com o modelo <i>NearestPoint</i> aplicado. . . . .	62
Figura 30 – Simulador apresentado os três modelos de percurso. . . . .	63
Figura 31 – Resultados comparativos. . . . .	64

Figura 32 – Diferença entre os modelos <i>NearestPoint</i> e <i>Kashuba</i> . . . . .	64
Figura 33 – Variação dos resultados obtidos em cada modelo para o percurso definido para o VANT. . . . .	65
Figura 34 – Comparativo de vitórias entre <i>NearestPoint</i> e <i>Kashuba</i> . . . . .	66
Figura 35 – Cenário com alta densidade de sensores. . . . .	66
Figura 36 – Cenário com baixa densidade de sensores. . . . .	67
Figura 37 – Algoritmo criado para representar o trabalho de <i>Kashuba</i> . . . . .	68
Figura 38 – (a) Disposição quadrada e hexagonal (b) de sensores no simulador. . .	69
Figura 39 – Representação do sistema programado de acordo com o percurso do VANT. . . . .	72
Figura 40 – Processo de reprogramação de <i>Timeslots</i> . . . . .	73
Figura 41 – Pseudo algoritmo que descreve o cálculo do <i>Timeslot</i> . . . . .	74
Figura 42 – Base de dados criada pelo VANT. . . . .	75
Figura 43 – VANT alimentando a lista de permissão/restrição. . . . .	75
Figura 44 – VANT descobrindo um nova EC. . . . .	75
Figura 45 – VANT utilizando a consulta à sua base de dados. . . . .	76
Figura 46 – Saída dos registros de tomada de decisão do VANT. . . . .	76
Figura 47 – Saída dos registros de tomada de decisão do VANT. . . . .	77
Figura 48 – Representação gráfica do sistema de restrições do VANT. . . . .	77
Figura 49 – Quantidade de VANTs – Cenário 01. . . . .	79
Figura 50 – Distância Percorrida – Cenário 01. . . . .	80
Figura 51 – Tempo de voo – Cenário 01. . . . .	80
Figura 52 – Quantidade de Saltos – Cenário 01. . . . .	81
Figura 53 – Quantidade de Decisões – Cenário 01. . . . .	81
Figura 54 – Quantidade de <i>Timeslots</i> – Cenário 01. . . . .	82
Figura 55 – Distância Percorrida e Distância da Rota – Cenário 01. . . . .	83
Figura 56 – Quantidade de VANTs – Cenário 02. . . . .	84
Figura 57 – Distância Percorrida – Cenário 02. . . . .	84
Figura 58 – Tempo de Voo – Cenário 02. . . . .	85
Figura 59 – Quantidade de Saltos – Cenário 02. . . . .	85
Figura 60 – Quantidade de decisões tomadas pelos VANTs – Cenário 02. . . . .	86
Figura 61 – Quantidade de <i>Timeslots</i> – Cenário 02. . . . .	87
Figura 62 – Distância Percorrida e Distância da Rota – Cenário 02. . . . .	87
Figura 63 – Quantidade de VANTs – Cenário 03. . . . .	88
Figura 64 – Distância Percorrida – Cenário 03. . . . .	89
Figura 65 – Tempo de voo – Cenário 03. . . . .	89
Figura 66 – Quantidade de Saltos – Cenário 03. . . . .	90
Figura 67 – Quantidade de Decisões – Cenário 03. . . . .	90
Figura 68 – Quantidade de <i>Timeslots</i> – Cenário 03. . . . .	91

Figura 69 – Distância Percorrida e Distância da Rota – Cenário 03. . . . .	91
Figura 70 – Quantidade de VANTs – Cenário 04. . . . .	92
Figura 71 – Distância Percorrida – Cenário 04. . . . .	93
Figura 72 – Tempo de Voo – Cenário 04. . . . .	93
Figura 73 – Quantidade de Saltos – Cenário 04. . . . .	94
Figura 74 – Quantidade de Decisões – Cenário 04. . . . .	94
Figura 75 – Quantidade de Timeslots – Cenário 04. . . . .	95
Figura 76 – Distância Percorrida e Distância da Rota – Cenário 04. . . . .	95
Figura 77 – Quantidade de VANTs – Cenário 05. . . . .	96
Figura 78 – Distância Percorrida – Cenário 05. . . . .	97
Figura 79 – Tempo de Voo – Cenário 05. . . . .	97
Figura 80 – Quantidade de Decisões – Cenário 05. . . . .	98
Figura 81 – Quantidade de Timeslots – Cenário 05. . . . .	98
Figura 82 – Quantidade de VANTs – Cenário 06. . . . .	99
Figura 83 – Distância Percorrida – Cenário 06. . . . .	100
Figura 84 – Tempo de Voo – Cenário 06. . . . .	101
Figura 85 – Quantidade de Saltos – Cenário 06. . . . .	101
Figura 86 – Quantidade de Timeslots – Cenário 06. . . . .	102
Figura 87 – Quantidade de VANTs – Cenário 07. . . . .	103
Figura 88 – Distância Percorrida – Cenário 07. . . . .	103
Figura 89 – Tempo de Voo – Cenário 07. . . . .	104
Figura 90 – Quantidade de Saltos – Cenário 07 . . . . .	104
Figura 91 – Quantidade de Decisões - Cenário 07 . . . . .	105
Figura 92 – Quantidade de Timeslots – Cenário 07 . . . . .	105
Figura 93 – Comparativo entre Quantidade de VANTs. . . . .	106
Figura 94 – Comparativo entre asDistância Percorrida pelos VANTs. . . . .	107
Figura 95 – Comparativo entre Tempo de Voo dos VANTs. . . . .	107
Figura 96 – Comparativo entre Quantidade de Saltos dos VANTs. . . . .	108
Figura 97 – Comparativo entre Tomadas de Decisões pelos VANTs. . . . .	108
Figura 98 – Comparativo entre Quantidade de TimeSlots. . . . .	109
Figura 99 – Tempo de voo em relação à sensores não descobertos pelo VANT. . . .	110
Figura 100 – Conexões do VANT com Estações de Coleta em função do tempo. . . .	110
Figura 101 – Decisões tomadas pelo VANT. . . . .	111
Figura 102 – Análise com 20 timeslots. . . . .	112
Figura 103 – Tempo médio para primeira tomada de decisão do UAV. . . . .	113
Figura 104 – Distância percorrida e Estações de Coleta Pendentes. . . . .	113
Figura 105 – Percurso do VANT, em função dos pontos de encontro das ECs. . . .	114
Figura 106 – Diferença de percurso entre caminho planejado e sobrevoado. . . . .	114
Figura 107 – Rota original planejada antes do percurso. . . . .	116

Figura 108–Quantidade de timeslots por cenário. . . . .	117
Figura 109–Variações de Tempo de cada timeslot para quantidade de ECs. . . . .	117
Figura 110–Variação de tempo de Broadcast para cada cenário. . . . .	118
Figura 111–Tempo médio de Broadcast x Tempo programado por EC. . . . .	119
Figura 112–Tempo de broadcast e Sleep da Estação de Coleta. . . . .	119
Figura 113–Ciclo de transmissão e hibernação das ECs. . . . .	120
Figura 114–Estado da EC até que sejam todas encontradas. . . . .	121
Figura 115–Percentual de redução de percurso do VANT. . . . .	122
Figura 116–Distância percorrida, e EC pendentes em função do tempo. . . . .	123
Figura 117–Decisões por VANT em função das ECs pendentes. . . . .	123
Figura 118–Tempo de simulação para 15m/s e 25m/s. . . . .	124
Figura 119–Diferença de Timeslots entre Conexões do VANT. . . . .	125
Figura 120–Conexões do VANT em função dos timeslots. . . . .	125
Figura 121–Tempo de Simulação com e sem restrição de timeslots. . . . .	126
Figura 122–Distância percorrida pelos VANTs com e sem restrição de timeslots. . .	127

# Lista de tabelas

Tabela 1 – Cenários configurados para Simulações. . . . .	78
---	----

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Objetivo	16
1.2	Hipótese de Pesquisa	17
1.3	Justificativa e Problematização	17
1.4	Contribuição	18
1.5	Publicações	18
1.6	Estrutura do Trabalho	19
<b>2</b>	<b>ESTADO DA ARTE</b>	<b>20</b>
2.1	O que é um VANT	20
2.2	VANTs Colaborativos	21
2.3	Comunicação entre VANTs Colaborativos	23
2.3.1	Redes <i>ad-hoc</i>	24
2.3.2	MANETs	25
2.3.3	VANETs	27
2.3.4	FANETs	28
2.3.5	UAANETs	29
2.3.6	Sistemas Centralizados	31
2.4	Protocolos de Roteamento para Redes Móveis	33
2.4.1	Protocolos Reativos	33
2.4.2	Protocolos Proativos	34
2.4.3	Protocolos Geográficos	34
2.4.4	Protocolos Híbridos	35
2.4.5	DTN – <i>Delay Tolerant Networks</i>	35
2.5	Modelos de Mobilidade	36
2.5.1	<i>Random WayPoint</i>	37
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>38</b>
3.1	Planejamento de Percurso	38
3.2	SRP – <i>Swarm Routing Problem</i>	39
3.3	<i>Middleware</i> para VANTs Colaborativos	41
3.4	Simuladores de Mobilidade	42
3.4.1	NS2 - <i>Network Simulator 2</i>	43
3.4.2	MATLAB	43
3.4.3	<i>The One</i>	44
3.4.4	OMNET++	44

3.4.5	Outros Trabalhos Relacionados . . . . .	45
4	<b>UAV SIMULATOR . . . . .</b>	<b>47</b>
5	<b>OTIMIZAÇÃO DO PERCURSO PLANEJADO . . . . .</b>	<b>55</b>
5.1	Modelo de Kashuba . . . . .	55
5.2	Validação do Modelo . . . . .	57
5.3	Modelo NearestPoint . . . . .	60
5.4	Resultados Comparativos . . . . .	63
5.5	Discussão dos Resultados . . . . .	67
6	<b>PROGRAMAÇÃO DOS SENSORES <i>TIMESLOTS</i> . . . . .</b>	<b>71</b>
6.1	Metodologia . . . . .	77
6.2	Resultados Comparativos . . . . .	105
6.3	Discussão dos Resultados . . . . .	109
7	<b>CONCLUSÃO . . . . .</b>	<b>128</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>131</b>

# 1 Introdução

As primeiras pesquisas sobre o tema de Veículos Aéreos Não Tripulados – VANTs, ou mais popularmente conhecidos como drones, tiveram início após a primeira guerra mundial com seus primeiros protótipos ([MAXA MOHAMED SLIM BEM MAHMOUD, 2017](#))([MOHAMED J. AL-JAROODI, 2013](#)). Durante muitos anos a principal força por trás da pesquisa e desenvolvimento nessa área foi a militar. Os primeiros testes foram realizados em 1917, quando dois inventores Lawrence e Elmer Sperry projetaram e construíram a primeira aeronave não tripulada batizada de “Torpedo Aéreo” ([CHAVES, 2013](#)).

Apenas durante a década de 1970 iniciou-se a era moderna dos VANTs, com os Estados Unidos e Israel desenvolvendo projetos de pequeno porte para aeronaves não tripuladas mais baratas que seus precursores. Segundo o autor ([CHAVES, 2013](#)), em 1991 após as operações do VANT Pionner que foi utilizado em mais de 300 missões durante a operação *Desert Storm* que a utilização de VANTs passou a ser vista com maior interesse pela indústria militar. Em 2001, logo após o atentado das torres gêmeas é que pode se considerar como um “divisor de águas” na história dos VANTs, pois após esse ataque os investimentos em pesquisas e aquisições nessa tecnologia aumentaram consideravelmente. Em abril de 2001, cinco meses antes do atentado terrorista, a previsão de investimentos na área de VANTs pelo Departamento de Defesa do EUA era de cerca de 300 milhões de dólares para os anos de 2000 e 2001. Pouco mais de um ano após ao ataque, em dezembro de 2002, a projeção de investimento para a área de VANTs subiu para 2.1 bilhões. Em 2008 a projeção foi de 3,4 bilhões de dólares. Em 2009 subiu para 3,6 bilhões de dólares. E 2010 chegou à 4,9 bilhões de dólares.

Com o amadurecimento da tecnologia, redução dos custos das peças, fácil acessibilidade e a rápida disseminação do conteúdo sobre a internet, drones nos dias de hoje são comercializados até como brinquedos para crianças e adultos, controlados até mesmo por seus smartphones via Bluetooth. Inúmeros desenvolvedores de sistemas abertos, tanto de hardware quanto software para VANTs tem impulsionado a popularidade desses equipamentos, tais como o projeto DroneCode ([DRONECODE, 2017](#)) apoiado pela fundação Linux, que tem o intuito de prover códigos de sistemas de auto navegação e controle para seus usuários, além de outras como a ArduPilot ([ARDUPILOT, 2017](#)), PX4 ([PX4, 2017](#)), Pararazzi UAV ([PAPARAZZI, 2017](#)), e OpenUAV ([OPENUAV, 2017](#)) onde existem fóruns, casos de sucesso, publicações e até mesmo tutoriais de como projetar o VANT para sua própria aplicação.

Inúmeras aplicações tem surgido com a popularização dos VANTs, tanto em meios civis, quanto militares como o mapeamento de áreas, busca e resgate, entrega de mercadorias,

cinografia, vigilância de fronteiras, segurança, na engenharia civil para monitoramento e acompanhamento de rachaduras em locais de difícil acesso, auxílio no içamento de cabos de alta tensão, aplicações em desastres naturais tais como monitoramento de incêndios florestais, exposição a resíduos tóxicos ou mesmo cenários envolvendo contaminação por radiação (MAXA MOHAMED SLIM BEM MAHMOUD, 2017)(MOHAMED J. AL-JAROODI, 2013)(MOHAMED J. AL-JAROODI, 2014)(CHAVES, 2013).

Veículos aéreos não tripulados tem ganho bastante atenção devido a sua flexibilidade de voo, pequeno tamanho e baixo custo. Um estudo recente indica que investimentos na área de VANTs em 2017 apenas nos Estados Unidos podem chegar à 5 bilhões de dólares (MOHAMED J. AL-JAROODI, 2013)(MOHAMED J. AL-JAROODI, 2014). A expectativa é que para 2018 os investimentos no mercado mundial para essa área cheguem à 11,9 bilhões de dólares, números que indicam um mercado promissor para investimentos (CHAVES, 2013). De acordo com estimativas do Banco Mundial, até o ano de 2050 o nosso planeta deverá estar habitado por cerca de 9,7 bilhões de seres humanos. A previsão de aumento na produção na agricultura até lá é de 70 % em relação ao que é produzido atualmente (BANCOMUNDIAL, 2017), e a tecnologia de VANTs aplicada à “Agricultura de Precisão” será decisiva para que tal número possa ser atingido.

Além disso, nas palavras do presidente da Associação Brasileira de Agricultura de Precisão (AsBraAP) Paulo Molin – “Globalmente, com o crescimento da população, estamos encarando pressão na produção agrícola para produzir mais comida, nas mesmas terras e com recursos limitados. Precisamos integrar tecnologia para sermos mais eficientes” (POVO, 2018), destaca o especialista numa entrevista no IV encontro sobre Agricultura de Precisão sediado em Curitiba em janeiro de 2018.

Alguns países já estão na vanguarda, como é o caso do Japão que já está investindo e utilizando tecnologias como essas a seu favor, hoje o país já conta com mais de 2000 VANTs atuando na agricultura (JORGE, 2014). Recentemente uma matéria publicada (MOHAMED J. AL-JAROODI, 2017) apresenta que um VANT é capaz de realizar o serviço de pulverização de um hectare  $10.000m^2$  de plantação em 10 minutos, serviço que antes levava um dia inteiro de trabalho para uma pessoa. Se pensarmos em vários drones operando de forma coordenada, podemos reduzir ainda mais esse tempo. Além de realizar a pulverização, o drone ainda fotografa a área abaixo com imagens de alta resolução dando ao agricultor um feedback de como está sua plantação em toda a área percorrida, algo que antes era possível apenas indo pessoalmente até o local. Isso mostra a eficiência da automação aplicada a essa área. Além disso, a empresa Kubota Comportation desenvolveu um sistema que é capaz de analisar as imagens da lavoura e determinar via inteligência artificial o lugar, e a quantidade exata de pulverização sem a necessidade de intervenção humana.

Também no Brasil já há iniciativas neste sentido, segundo dados da EMBRAPA

– Empresa Brasileira de Pesquisa Agropecuária, a utilização de VANTs na lavoura traz uma série de vantagens, um exemplo são sistemas que são desenvolvidos para realizar diagnósticos da lavoura através do tratamento das imagens realizados pelos VANTs, com isso é possível determinar plantas invasoras, falhas de plantio, medir porcentagem de cobertura vegetal, tendo um gerenciamento mais preciso da lavoura. Em uma lavoura, cada área possui suas particularidades, e é a possibilidade de identificar essas diferenças causadas por pragas, doenças ou mesmo deficiências que torna essa tecnologia tão benéfica. Segundo o pesquisador da EMBRAPA, tratar as áreas como não uniformes possibilita o agricultor rural a economia de insumos, redução de impactos ambientais e o aumento da produtividade (EMBRAPA, 2017).

Os valores de VANTs voltados para agricultura podem variar de 5 a 300 mil reais dependendo de suas capacidades de sensoriamento (EMBRAPA, 2017). Sensores simples operando no espectro visível de frequência têm finalidade de análise da biomassa. Porém para análise do estresse nutricional são necessários sensores infravermelhos. Para estresse hídrico é necessária a utilização de câmeras térmicas, e para detecção de doenças são necessários sensores hiperspectrais. Além disso, é necessário treinamento para operação dos drones, além da sua manutenção. Portanto essa é uma área que tem atraído muitos jovens para a agricultura além de criar áreas de pesquisa e desenvolvimento, criando desde cursos técnicos, graduações e pós-graduações voltados para o mercado civil, industrial e agrícola (PUCPR, 2017).

Outro fator de extrema importância é o gerenciamento correto de recursos hídricos no solo, ou seja, saber o momento correto de fornecer água para a planta além da dosagem correta para cada tipo de plantação é de extrema importância, pois o uso excessivo de água pode causar estresse hídrico nas plantas, além de infiltração de água no solo levando os nutrientes do solo raso para o solo mais profundo onde as raízes não alcançam, causar processos erosivos no solo, até mesmo processo de apodrecimento das raízes das plantas.

Saber o momento correto de irrigar o solo, em que a ausência de umidade começa a comprometer a absorção de água pela planta é fundamental, portanto para isso são utilizados sensores instalados no solo que realizam a medição da tensão de água e umidade no solo, garantindo assim um gerenciamento mais eficiente da plantação, economizando recursos hídricos além de prover redução de custos ao agricultor (COELHO, 2015) e melhor gerenciamento do plantio.

Porém, a coleta e interpretação dos dados dos sensores ainda é realizada de forma presencial e manual (COELHO, 2015), conforme representado pela Figura 1, uma área que ainda pode ser automatizada com utilização de VANTs para leitura e coleta dessas informações.



Figura 1 – Agricultor realizando mediação do solo.

Fonte: (COELHO, 2015)

## 1.1 Objetivo

VANTs colaborativos podem ser aplicados no cenário acima para resolver o problema de transmissão dos dados lidos pelos sensores instalados nas lavouras. Ao utilizar rede de VANTs que não possuem restrições em termos de movimentação através da lavoura e aproveitando a velocidade e flexibilidade com que se movem, podemos utilizá-los de forma a possibilitar o envio e recepção de dados dos dispositivos em áreas sem cobertura celular, por exemplo.

Assim o desafio é estabelecer rotas adequadas para VANTs colaborativos de forma a cobrirem um conjunto de dispositivos em solo com a restrição adicional de que, por questões de economia de energia, tais dispositivos que chamaremos de **Estações de Coleta**, possuam janelas constantes de comunicação (transmissão e recepção). Estas podem ser ajustadas de forma a gradativamente otimizar as rotas futuras dos VANTs. Ou seja, a janela de comunicação de cada estação de coleta poderá ser ajustada de acordo com o planejamento de percurso dos VANTs que as sobrevoarão, garantindo assim maior eficiência energética por parte do sistema. Este projeto busca um modelo que vise a minimização da quantidade de VANTs, percurso percorrido e tempo gasto para sobrevoar de todas as estações pelos VANTs, em função da quantidade de estações de coleta distribuídas sobre uma determinada área, reprogramando e respeitando a janela de tempo estipulada pelo sistema de comunicação destas estações, de acordo com o percurso do VANT.

Pesquisas sobre VANTs colaborativos tem recebido muita atenção nos últimos anos de governos, universidades, centros de pesquisa, além da própria indústria comercial, pois

ao aplicar o conceito de uma rede ad-hoc formada por estes nós móveis - VANTs, podemos criar redes de comunicação e cobertura em larga escala independentemente de qualquer infraestrutura existente ou não no local de interesse. Estas são redes baratas (em relação as centralizadas), descartáveis e dinâmicas que se auto organizam e se auto regeneram em caso de falhas, portanto sua confiabilidade é alta quando bem estabelecida, além de todos os outros serviços que esta rede pode estabelecer sobre a agricultura que serão abordados durante o trabalho.

## 1.2 Hipótese de Pesquisa

Para este projeto, será definido um modelo de comunicação e determinação de rotas mínimas aplicado sobre um simulador para validação do modelo SRP – *Swarm Routing Problem*, coordenando o planejamento do percurso dos VANTs com a janela de comunicação das estações de coleta em solo.

Em seguida, serão realizados testes com o sistema programando as janelas de transmissão de cada estação de acordo com seu histórico de sobrevoo para aquela região, garantindo assim um tempo menor de broadcast pelas mesmas, ajustando assim tais intervalos de comunicação com o percurso de cada VANT, reduzindo o tempo de transmissão desnecessário, melhorando a performance do sistema. Além disso, será aplicado um sistema de restrição de *timeslots* para gerenciamento das escolhas do VANT de ir não para aquela estação de coleta de acordo com histórico.

Comparações entre modelos programáveis e não programáveis das estações de coleta poderão ser realizadas para mensurar o impacto da programação sobre suas janelas de transmissão sobre, além do sistema modelar de forma dinâmica a quantidade VANTs, percurso e tempo mínimo necessários para percorrer  $n$  estações de acordo com suas respectivas janelas de transmissão.

Estamos considerando para o projeto que os VANTs são homogêneos, ou seja, não existem diferenças em suas características mecânicas, eletrônicas ou mesmo aerodinâmicas, e possuem uma bateria que se extingue, portanto será necessário um método para substituição dos VANTs em curso e troca de informações entre eles sobre estações já programadas e pendentes. Além disso as estações de coleta em solo serão consideradas homogêneas (mesmo alcance de transmissão, mesma frequência e período de ativação, possuindo estes instantes de comunicação programáveis pelos VANTs).

## 1.3 Justificativa e Problematização

A principal riqueza do nosso país provém da agricultura. O problema é que atualmente grande parte dos processos que vão desde a preparação do solo até a colheita,

passando pelo plantio e crescimento da cultura (com monitoramento do solo, de incidência de pragas e doenças, etc.) ainda não fazem uso das tecnologias de comunicação e dos dispositivos amplamente acessíveis no contexto da Internet das Coisas (IoT – *Internet of Things*). Em particular, o monitoramento e controle das condições de vastas lavouras por meio de sensores e/ou atuadores (com severas restrições de bateria e comunicação) é de grande importância ao aumento da produtividade das culturas (COELHO, 2015). O problema é que a medição ainda é realizada de forma local e manual pelos agricultores, que entram na lavoura com seus equipamentos, sensores e baterias realizando a medição em pontos específicos da área de plantação.

Neste contexto, sensores/atuadores possuem grandes restrições de energia e, portanto, o seu modo típico de funcionamento é alternando intervalos de “*stand-by*” com momentos em que efetuam medidas e transmitem os dados (no caso de sensores). De qualquer modo, é necessário haver uma infraestrutura que possibilite a comunicação entre os dispositivos e os servidores onde os dados serão processados (“*analytics*” ou “*Big Data*”). Apesar de existirem especificações de frequências e protocolos específicos para comunicação de dispositivos com estações rádio base (*eNodeBs*) (LTE-M, NB-IoT), atualmente todas elas são experimentais. Além do mais, no contexto das culturas (particularmente aquelas envolvendo grandes áreas), não há atualmente perspectiva no horizonte de uma cobertura ampla e total que possibilite a comunicação com dispositivos que possuam as restrições de energia e transmissão mencionadas acima.

## 1.4 Contribuição

Colaborar com a área de desenvolvimento de aplicações para VANTs colaborativos, aplicados à sistemas para agricultura de precisão abordando um sistema de automação para leitura de dados sensoriais do solo de forma eficiente, atualmente realizado de forma manual. Como já mencionado anteriormente, a presente proposta vai ao encontro de projetos atualmente em curso na cooperação com a Nokia, que está trabalhando com desenvolvimento de sensores de IoT voltados à agricultura de precisão.

## 1.5 Publicações

Esta dissertação de mestrado ainda não possui publicações realizadas até este momento, porém o artigo já se encontra em andamento.

## 1.6 Estrutura do Trabalho

O trabalho está dividido como segue. O capítulo 2 trata de VANTs, seus principais conceitos, tipos de comunicação e diferenças entre tais. Protocolos de roteamento e modelos de mobilidade. No capítulo 3 estão trabalhos relacionados, tais como Planejamento de Percurso, Protocolos de Roteamento entre VANTs, Middleware e Simuladores de Mobilidade. No capítulo 4 é apresentado o simulador desenvolvido para a realização deste trabalho. No capítulo 5 são apresentados os resultados obtidos para otimização de percurso do VANT, e finalmente no capítulo 6 são explanados os conceitos envolvidos para a reprogramação dos sensores de acordo com o percurso do VANT.

## 2 Estado da Arte

Neste capítulo serão apresentadas as principais tecnologias e conceitos envolvidos para a proposta do trabalho de qualificação. Inicialmente são descritas as principais características de VANTs, e em seguida a concepção de sistemas de VANTs colaborativos: como ocorre a comunicação e quais os tipos de redes existentes, comparando sistemas centralizados com os descentralizados, além de modelos já concebidos como, *ad-hoc*, FANETs - *Flying Ad-hoc Networks*, MANETs - *Mobile Ad-Hoc Networks*, UAANETs - *Unmanned Aerial Ad-hoc Networks* e DTNs - *Delayed Tolerant Networks*.

Além disso, os principais protocolos de roteamento para redes móveis são detalhados, sejam eles reativos, proativos, geográficos ou híbridos, além de verificar alguns modelos de mobilidade dos nós dessa rede. O modelo de mobilidade interfere significativamente na performance da rede de VANTs colaborativos, seja esse padrão baseado em grupo ou entidade. Este tema deve ser levado em consideração em nosso trabalho.

### 2.1 O que é um VANT

O termo VANT - Veículo Aéreo Não Tripulado, segundo a ABA – Associação Brasileira de Aeromodelismo é: “Um veículo capaz de voar na atmosfera, fora do efeito do solo, que foi projetado ou modificado para não receber um piloto humano que é operado por controle remoto ou autônomo”. (CHAVES, 2013)

Basicamente o VANT, ou popularmente mais conhecido como “**drone**”, é composto por um conjunto de sistemas eletromecânicos, baterias, processadores, memórias, transmissores e receptores para comunicação sem fio com outros drones (VANT – VANT) ou mesmo com a ECT – Estação de Controle Terrestre (VANT – ECT), além de serem equipados com um conjunto de sensores voltados para sua aplicação específica, tais como câmeras fotográficas (seja no espectro visível ou não), sistemas de geoposicionamento (GPS), sensores dos mais variados tipos como temperatura, humidade, pressão, radiação, memória para armazenamento dos dados coletados, processadores para o tratamento desses dados, além de compartimento de carga, também conhecido como *payload* (MAXA MOHAMED SLIM BEM MAHMOUD, 2017)(MOHSENI R. HASSAN, 2010).

Além disso, os VANTs podem ser classificados como asas-fixas (*fixed-wings*) conforme Figura 2-a, ou asas-rotativas (*rotary-wings*) Figura 2-b. Também podem ser classificados por peso, sistema de decolagem (VTOL – *Vertical Take-off Landing*) ou em curto espaço (STOL – *Short Take-Off Landing*), com controle *on-board* (autônomo) ou *off-board* (controlado por operador), onde muitos autores o classificam como veículos aéreos

semiautônomos (CHAVES, 2013).



Figura 2 – (a) VANT tipo asas-fixas, (b) VANT tipo asas-rotativas

## 2.2 VANTs Colaborativos

Uma área de pesquisa que tem recebido muita atenção e investimento nos últimos anos é a área de VANTs colaborativos. Existem diversas vantagens em trabalhar com VANTs colaborativos pois certos cenários, tais como busca e resgate, onde o tempo é fator determinante para o sucesso da operação, a divisão do trabalho entre vários VANTs bem coordenados é altamente viável. Além disso, o compartilhamento de recursos entre VANTs é outro fator que traz muita flexibilidade ao objetivo da tarefa. A comunicação VANT-VANT garante o funcionamento em áreas sem qualquer infraestrutura de comunicação, o que é muito interessante para cenários como desertos, florestas, oceanos ou mesmo em casos de vastas propriedades agrícolas (MOHAMED J. AL-JAROODI, 2013)(MOHAMED J. AL-JAROODI, 2014)(MOHSENI R. HASSAN, 2010).

Vários pesquisadores têm se dedicado a estudar, desenvolver e validar técnicas que resolvamos algumas das maiores dificuldades em lidar com tal sistema.

Uma delas é a questão do controle, onde existe um operador que controla um VANT, e os demais VANTs o seguem de forma coordenada, modelo conhecido como enxame. O VANT que é controlado por um operador humano é chamado de líder. Além do controle, questões como o agrupamento, espalhamento e técnicas para evitar a colisão entre os VANTs devem ser levadas em consideração, além do desvio de possíveis obstáculos que possa haver no percurso dos drones. Áreas como DSA (*Detect-Sense-and-Avoid*) tem sido muito estudada para solucionar essa questão (MOHAMED J. AL-JAROODI, 2013)(MOHAMED J. AL-JAROODI, 2014).

O segundo problema é a comunicação em rede em um sistema colaborativo que pode envolver desde problemas de área de cobertura, falta de visada entre o controlador e o VANT líder, questões de confiabilidade da rede, largura de banda, atraso e perda de comunicação devem ser consideradas em um cenário móvel, além disso, a comunicação *ad-hoc* apesar de ser extremamente convergente, é altamente complexa, ainda mais ao lidar com equipamentos, transmissores e receptores heterogêneos ou não (MAHMOUD, 2014). Além disso, devido à alta capacidade de mobilidade do sistema, ela deve se autoconfigurar

constantemente e de forma rápida e dinâmica entre os VANTs, garantindo as rotas de comunicação entre origem e destino. Ou em um cenário onde possa haver desconexões do sistema, o VANT deverá armazenar os dados para transmiti-los assim que retornar para a área de cobertura, nesse caso, uma rede tolerante à atrasos e desconexões (CHENG MOORE, 2012)(BAUMANN F. LEGENDRE, 2008).

Segundo um levantamento realizado por Mohamed et al. (MOHAMED J. AL-JAROODI, 2014), para um sistema de VANTs ser caracterizado como colaborativo, ele deve possuir no mínimo as seguintes características.

- **Sensoriamento Colaborativo:** Vários VANTs devem colaborar de forma a cumprir seu objetivo em conjunto, como mapear uma área, ou inspecionar um objeto utilizando um ou mais sensores. Isso requer uma distribuição de tarefas de forma a realizar o trabalho de forma mais rápida e sem sobreposição. A distribuição das tarefas deve ser distribuída para cada VANT de forma dinâmica, portanto é necessário que haja comunicação constante entre as unidades além de coordenação do que já foi realizado, e do que está pendente.
- **Comunicação Colaborativa:** VANTs necessitam de diferentes meios de comunicação que permitam que interajam entre si, seja na comunicação VANT – VANT, ou na comunicação ECT – VANT, que são tratadas de forma diferenciadas. Além disso, existem diversas tecnologias de comunicação sem fio, frequências, além do fato das antenas de transmissão e recepção dos VANTs possuírem potência limitada para poupar recursos de energia impactando na distância entre eles. Formas centralizadas e distribuídas podem ser adotadas, cada uma com suas devidas vantagens e desvantagens.
- **Processamento Colaborativo:** Tarefas que exigem alto processamento, tais como a análise de imagens ou vídeos em alta definição. Em situações onde os VANTs estão operando em áreas remotas, longe da ECT, o processamento desses dados é fundamental para alguma tomada de decisão no local onde envolva um curto espaço de tempo, portanto a distribuição coordenada desse processamento pode ser realizada através de VANTs próximos ao local.
- **Armazenamento de Dados:** Assim como alguns VANTs podem ter maiores capacidades de armazenamento que outros, é importante que o uso compartilhado desse recurso possa ser aproveitado pelos demais VANTs no local em casos que haja necessidade.
- **Controle Colaborativo:** Controle em tempo real é fundamento para a eficiência e segurança nas operações envolvendo múltiplos VANTs. Em cenários onde existe um controlador humano, ele deverá coordenar apenas um drone, chamado de líder do grupo, e o restante do enxame deverá se mover de acordo com os movimentos deste

líder. Existe também a possibilidade de se tratar de um drone autônomo, onde não há operador humano, esse cenário pode existir caso ocorra perda de comunicação entre ECT e o VANT líder, os demais VANTs do enxame deverão seguir um padrão de movimentação em função deste.

Como pode-se perceber, ao lidar com sistemas colaborativos existem muitos aspectos importantes que devem ser levados em consideração ao tratar-se de múltiplos VANTs pertencentes ao mesmo sistema. Segundo o levantamento realizado por Mohamed et al. (MOHAMED J. AL-JAROODI, 2013), vários estudos nessa linha estão sendo conduzidos ao redor do mundo. Ryan et al. (RYAN, 2007) implementaram um sistema que controla vários VANTs que realizam sensoriamento para uma determinada tarefa sobre o controle de um único operador. Vatchsevanos et al. (VACHTSEVANOS L. TANG, 2004) propuseram uma abordagem inteligente para a coordenação de múltiplos VANTs além de um sistema para detecção de colisões entre as unidades do enxame. Bruggerman et al. (BRUGGEMANN J.J. FORD, 2011) estudaram os aspectos de controle para VANTs colaborativos utilizados na inspeção de estruturas lineares, tais como oleodutos, gasodutos, estradas e linhas de transmissão de alta potência. Rathinam et al. (RATHINAM, 2005) também trabalharam com VANTs colaborativos em um sistema para monitorar estruturas lineares, além disso, como consequência de seu trabalho, realizaram uma análise técnica dos aspectos da comunicação ad-hoc em rede entre VANTs.

## 2.3 Comunicação entre VANTs Colaborativos

O modelo proposto para comunicação entre os VANTs colaborativos irá impactar diretamente no desempenho do sistema. Sistemas centralizados, possuem simplicidade em seus protocolos de comunicação, porém dependem de área de cobertura para seu pleno funcionamento. Sistemas descentralizados são considerados os mais adequados para VANTs colaborativos devido a sua grande flexibilidade e escalabilidade ao entrar ou sair um nó da rede existente.

Estudos nessa área especificamente voltados para área de comunicação entre os VANTs estão sendo conduzidos. Mahmoud et al. em (MAHMOUD, 2014)(MAHMOUD, 2015) estudam a aplicação de redes 4G/LTE através de chips celulares implantados em VANTs para sistemas colaborativos em nuvem, através de uma plataforma de webservices, provendo integração e conectividade de forma simples para os usuários, sugerindo um sistema centralizado de comunicação com os VANTs. No caso o autor propõe um cenário com estrutura já existente, supondo que sempre haverá cobertura para o sistema proposto. Já o autor em (SAHINGOZ, 2013) defende a utilização da comunicação ad-hoc entre VANTs, pois é um cenário que independe de estrutura de comunicação, sugerindo cenários caóticos tais como áreas de desastres naturais, incêndios florestais, cidades devastadas

por furacões, citando as vantagens desse modelo de conexão pelo fato de serem redes autoconfiguráveis, sendo assim os próprios nós (VANTs) atuam como roteadores e pontos de encaminhamento das mensagens para outros nós, desde a origem até o destino da informação.

### 2.3.1 Redes *ad-hoc*

Redes *ad-hoc*, nome proveniente do latim com tradução livre “para este fim” é o conceito de comunicação ponto-a-ponto (P2P - *Peer-to-Peer*), sem a necessidade de uma estrutura centralizadora gerenciando sessão estabelecida entre dois equipamentos ou mais, ou seja, um equipamento se comunica diretamente com seu par sem a necessidade de um negociador para que isso ocorra. Geralmente ela é pensada para cenários onde não existem infraestrutura, ou em casos onde deve-se haver comunicação rápida, barata, descartável. Redes assim se criam ao aproximar os nós, e se desfazem quando se afastam, são também conhecidas como redes espontâneas (RYAN, 2007), conforme apresentada na Figura 3.

Estes nós não agem de forma passiva na rede, apenas enviando ou recebendo informações, eles agem de forma ativa, como roteadores entre os nós de origem e destino, tomando decisões de roteamento, direcionando os dados há quem tomar a decisão no momento baseado em seu protocolo de roteamento.

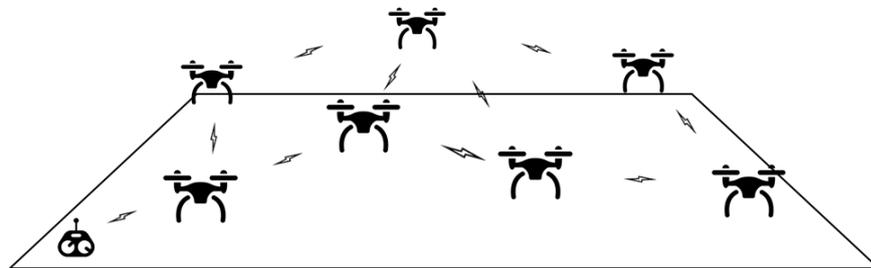


Figura 3 – Modelo de comunicação *ad-hoc*.

Nessa linha de pesquisa, os autores em (SHRIT S. MARTIN, 2017) criam um sistema de para transmissão de streaming de vídeo através de uma rede *ad-hoc* criada por VANTs interconectados entre si de forma dinâmica, utilizando como referência a potência do sinal recebido nas antenas dos VANTs para controlar a distância entre os mesmos, mantendo um mínimo aceitável para manter comunicação sem degradação. Ou seja, conforme a relação sinal/ruído entre os VANTs diminui em função da distância entre os VANTs, outro decola para se posicionar entre os dois VANTs garantindo a comunicação e qualidade da transmissão de vídeo, expandindo a distância de enlace entre origem e destino. Conforme a Figura 4, um drone líder captura imagens através de sua câmera e as envia para seu próximo salto, que repassa o *streaming* para o drone subsequente, até que a transmissão chegue finalmente ao seu destino.

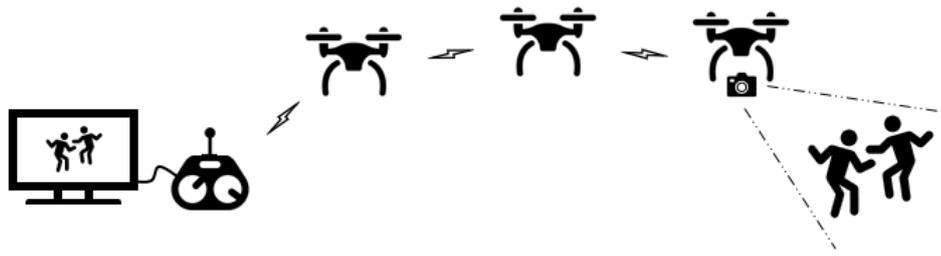


Figura 4 – Transmissão do *streaming* de vídeo através da rede *ad-hoc*.

### 2.3.2 MANETs

Rede tipo MANET – *Mobile Ad-hoc Networks*, são formadas por um conjunto de nós móveis, diretamente interconectados, formando uma rede temporária (ou não) através de tecnologia sem fio, ou seja, sem a necessidade de haver infraestrutura de comunicação ou administração centralizada. Devido à mobilidade, a rede se auto organiza, e se auto configuram (SAHINGOZ, 2014). Redes MANET são formadas por nós tais como smartphones, notebooks, televisores, impressoras, ou mesmo sensores de IoT, que são móveis, possuem limitação de processamento, memória e energia, devido ao seu pequeno tamanho, sua mobilidade é considerada lenta (CHENG MOORE, 2012), conforme apresentado na Figura 5.

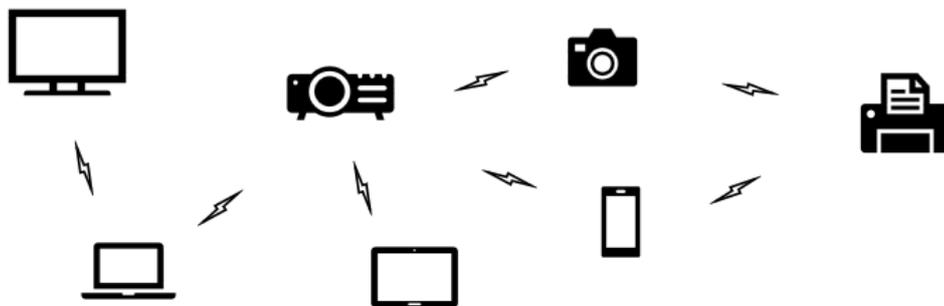


Figura 5 – Transmissão do *streaming* de vídeo através da rede *ad-hoc*.

De acordo Mohamed et al. em (MOHAMED J. AL-JAROODI, 2014), redes enquadradas nesse conceito possuem as seguintes características:

- Topologia dinâmica: Os nós são livres para movimentarem de forma arbitrária, o que significa que a topologia da rede irá se alterar de forma lenta e aleatória, além desse tipo de rede ser baseada em saltos, ou seja, entre a origem da informação e seu destino final, pode passar por vários nós.

- Limitação de banda: Interfaces de transmissão e recepção sem fio possuem menor capacidade em termos de banda, do que interfaces cabeadas. Além disso, a taxa de transmissão real da interface sem fio é geralmente menor que sua taxa de transmissão máxima, devido à ruídos, interferência, degradação do sinal e problemas como multi-percurso da onda eletromagnética (KUMAR, 2012).
- Baixa potência de transmissão: Os nós móveis, em geral possuem limitação de energia em função de suas baterias, portanto a limitação na potência de transmissão de seus circuitos causará um impacto na distância entre os nós para que haja comunicação.
- Controle descentralizado da rede: Sistemas de natureza descentralizada, possuem maior robustez contra falhas comparados aos sistemas centralizados. Ao desligar um *access-point*, por exemplo, todos os dispositivos WIFI conectados a este param de se comunicar, diferente de MANET em que a comunicação ocorre de forma direta entre os dispositivos.

Muitos estudos nessa área estão em andamento, pois assim como as redes MANET possuem uma série de vantagens sobre as redes convencionais centralizadas, elas também possuem seus desafios, tais como:

- Roteamento: Como a topologia da rede sofre constantes alterações devido a mobilidade dos nós, a dificuldade de se encaminhar uma mensagem ou transmissão de um nó de origem até seu destino passa a ser um desafio, pois a rota sofrerá alterações durante a transmissão. Além disso, as rotas podem conter vários caminhos para o mesmo destino, portanto métricas para definir quais as melhores rotas devem ser levadas em consideração, com isso, a topologia da rede torna-se extremamente complexa (UMESH S. MEWADA, 2011). Vários protocolos de roteamento existem para redes MANET e serão discutidos posteriormente neste trabalho.
- Consumo de energia: eficiência energética deve ser levada em consideração durante o desenvolvimento do sistema, pois os nós possuem recursos limitados quando se trata de energia. Pujolle et al. (SENOUCI, 2004) trabalha uma nova variável em termos de protocolo de roteamento para redes móveis ao incluir o nível de energia que os nós possuem na tomada de decisão para encaminhar o tráfego para um nó que esteja com níveis críticos de energia, fazendo com que o tráfego seja direcionado para outra rota mais estável.
- Interconexão: uma rede MANET pode se conectar com uma rede fixa, porém, o protocolo de roteamento deve levar em consideração a coexistência de outros protocolos de roteamento que sejam designados para essa rede.

- Segurança e Confiabilidade: um dos assuntos mais falados atualmente em redes é a segurança. O meio sem fio possui uma série de problemas e quando falamos em comunicação *ad-hoc* deve ser levado em consideração mecanismos de autenticação entre os pares, para evitar que alguém interfira na rede criada, desviando o tráfego, criando escutas ou mesmo criando rotas falsas entre os nós.
- Qualidade de serviço (QoS): Prover níveis diferenciados de qualidade de serviço em um cenário que está mudando constantemente impõe uma série de desafios. O autor Pujolle et al. ([BADIS A. MUNARETTO, 2003](#)) propõe em seu artigo a aplicação de um sistema de qualidade de serviço baseado na diferença do tempo de envio e recebimento da mensagem, que representa assim o atraso (*delay*) entre os nós durante a troca de pacotes de *Hello* na descoberta de seus vizinhos, criando assim uma métrica para prover QoS sobre MANETs.

### 2.3.3 VANETs

Redes tipo VANETS – *Vehicular Ad-hoc Networks*, estão sendo muito discutidas nos últimos anos devido às vantagens atribuídas aos carros autônomos, além das melhorias de prover comunicação direta entre veículos numa rodovia para aumentar a segurança dos passageiros nas rodovias. Por exemplo, garantir distância mínima entre os mesmos, sistemas de comboio para piloto automático, etc. Nessa modalidade de rede, especificamente para veículos terrestres os mesmos são equipados com sensores, transmissores e receptores para que haja comunicação direta com os demais veículos (V2V - *Vehicle-to-Vehicle*), e com sensores (V2I - *Vehicle-to-Infrastructure*) instalados em torno da rodovia ([EZE S. ZHANG, 2014](#)). Seu modelo de mobilidade é previsível, limitado às vias propostas ao veículo.

Um veículo ao aproximar-se de outro à sua frente, criará uma rede momentânea, dados serão recebidos e enviados durante esse período em que houver proximidade, e em seguida a rede V2V se desfaz. No caso do tipo de comunicação V2I, onde os sensores ao lado da pista colhem informações referentes ao veículo e as enviam para um servidor, o tipo de comunicação é estrela, ou seja, centralizado ([KARAGIANNIS O. ALTINTAS, 2011](#)).

Esse conjunto de sensores, receptores e transmissores sem fio permitem criação de aplicações tais como sensoriamento para velocidade excessiva, pré-aviso em caso de colisão do veículo, alertas quando o motorista mudar de pista para situações de ponto cego, ou alta velocidade em um determinado trecho antes da curva, alarmes em caso de avanço sobre semáforo, e encaminhamento de mensagens de alarme cooperativo em caso de acidentes na pista, ou seja, um veículo avisa o próximo sobre o acidente ([KARAGIANNIS O. ALTINTAS, 2011](#)) reduzindo assim a velocidade próximo da região do acidente. É possível criar um sistema de alerta às autoridades e responsáveis médicos próximos ao

local desse acidente. De acordo com o autor, os veículos conectados trocam mensagens de proximidade, distância, e velocidade à cada 100ms entre si.

Na Figura 6, o motorista ao gerar uma mudança brusca de pista, numa situação de perigo poderá ser alertado sobre sua ação devido ao sensor de proximidade perceber a presença de um outro automóvel próximo a ele durante o processo de mudança de faixa, evitando assim que o motorista continue com essa ação. Além disso, o motorista do caminhão também será informado sobre o movimento perigoso do veículo que está trocando de faixa, podendo assim reduzir a velocidade, evitando assim que problemas maiores possam vir ocorrer.

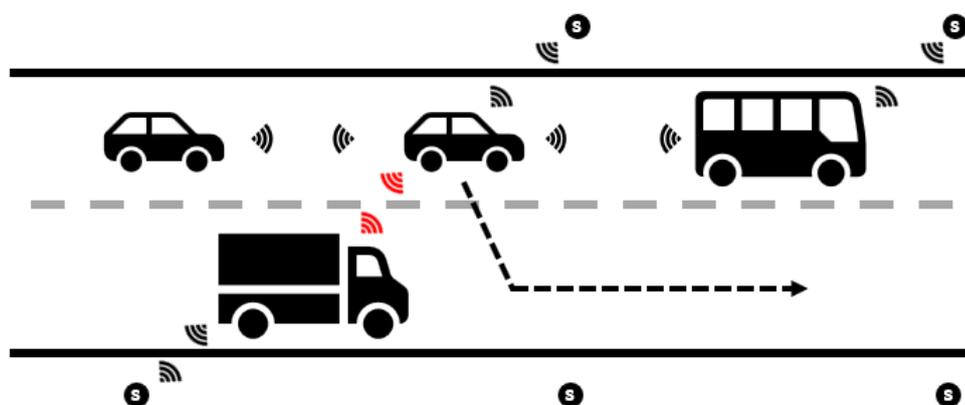


Figura 6 – Atuação dos sensores na comunicação Veículo-Veículo (V2V).

### 2.3.4 FANETs

Redes FANETs – *Flying ad-hoc networks*, são uma ramificação das MANETs e VANETs, porém o conceito é praticamente o mesmo: um sistema de nós móveis que trocam informações através de uma rede *ad-hoc*. Sua principal característica é que os nós têm capacidade de voo, portanto seu modelo de mobilidade não é previsível assim como as VANETs que seguem rodovias, ou lento como as MANETs, onde seres humanos carregam dispositivos móveis tais como *smartphones* ou *notebooks*, além disso, as redes FANET podem ser tripuladas ou não.

Isso traz diversos desafios na forma como as mensagens são trocadas, e como as tabelas de roteamento são criadas e atualizadas nos nós devido às rápidas alterações ocorridas na rede, fazendo com que a topologia mude constantemente (MAHMOUD, 2014). Já foi provado e discutido que a forma como os nós se locomovem em um determinado plano interfere diretamente na performance da rede *ad-hoc*, esse assunto será abordado mais para frente.

Nessa modalidade de rede, assim como nas redes VANET existem duas formas de comunicação, VANT – VANT estabelecendo comunicação *ad-hoc* entre si, sempre de

forma dinâmica entre os pares. Comunicação direta entre VANTs podem ser aplicadas nas mais diversas áreas devido a sua rápida expansão. A comunicação fim-a-fim é dada através de saltos entre os próprios nós, ou pode-se criar uma rede mesh, onde todos os VANTs comunicam-se diretamente com os demais VANTs na rede. Além disso, temos também comunicação do VANT (V2 I), onde os VANTs enviam e recebem informações de controle e sensoriamento com a ECT, conforme apresentado na Figura 7.

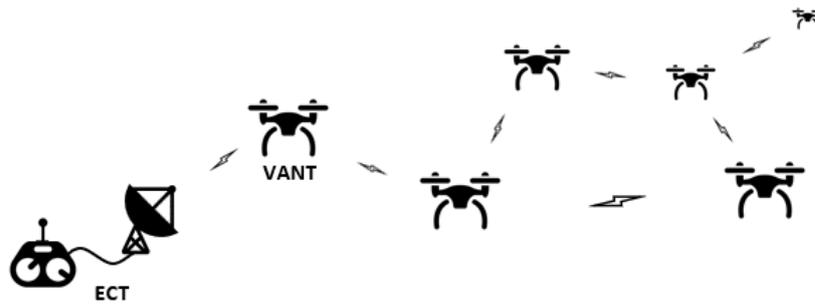


Figura 7 – Modelo de comunicação VANT-ECT, e VANT-VANT.

Utilizando a estrutura acima criada, uma rede FANET pode ser rapidamente implantada devido à não haver necessidade de estrutura fixa, seja onde for. Na literatura é comum encontrar diversos nomes para redes *ad-hoc* criadas por VANTs, tais como *Airbone Networks* (CHENG MOORE, 2012)(FREW, 2008), *Unmanned Aeronautical Ad-hoc Networks* (UAANETs) (SHIRANI M. ST-HILAIRE, 2012)(LI M. ST-HILAIRE, 2012a), *UAV Ad-hoc Networks* (CAI F.R. YU, 2013), *Network of UAVs* (TUCHELMANN, 2011), *Distributed Aerial Sensor Networks* (LI M. ST-HILAIRE, 2012b), e *Flying Ad-hoc Networks*(FANETs) (FREW, 2009), além de outros variados, tais como *Distributed Aerial Sensor Network*, *UAV Fleet Network*, *Network Aerial Robots*, e também *Mobile Ad-Hoc Unmanned Aerial Vehicle Communication Network* (MAXA MOHAMED SLIM BEM MAHMOUD, 2017). Nesse trabalho vamos padronizar a nomenclatura de UAANETs para nos referirmos à nossa rede *ad-hoc* para VANTs colaborativos não tripulados.

### 2.3.5 UAANETs

Diferentemente de uma FANET, uma rede do tipo UAANET – *Unmanned Aerial Ad-hoc Network*, obrigatoriamente é do tipo não tripulada. As FANETs podem ser criadas por sistemas tripulados [20].

Em cenários tais como agricultura, alto-mar, busca e resgate em florestas, ou mesmo cenários caóticos o modelo centralizado não se aplica para redes UAANET. A comunicação *ad-hoc* tem muito a oferecer em situações como estas, pois é uma rede que, como já vimos se auto cria, se auto gerencia e se auto organiza de acordo com a demanda.

Em redes *ad-hoc*, todos os VANTs devem se organizar para que todos saibam para quem encaminhar as mensagens recebidas, agindo como um roteador na rede para cada salto. Nas redes UAANET, a ECT é considerada como um nó dentre todos os demais, ela não precisa ser fixa, pode ser móvel também, ou seja, ela se comunica com o VANT mais próximo, que age como um *gateway*, encaminhando a informação de nó em nó até que chegue ao seu destino.

Na comunicação VANT – VANT, os mesmos podem se comunicar diretamente, ou através de outro VANT, um sistema multi-saltos. Na comunicação VANT – ECT, pode-se considerar o encaminhamento dos dados de telemetria, sensoriamento e geo-localização, além de mensagens de heartbeat, dando a ECT a presença do VANT na rede e uma noção de latência do sistema para aquela aeronave (MAXA MOHAMED SLIM BEM MAHMOUD, 2017).

Em comparação com modelos centralizados, o sistema UAANET possui várias vantagens, dentre elas a escalabilidade, que graças a rápida mobilidade dos VANTs permitem cobrir uma área rapidamente, além disso, a confiabilidade é alta, pois num cenário onde um VANT sofra algum mau funcionamento, não afetará o sistema como um todo, além da realocação de outro VANT para aquela área possa até ocorrer de forma automática devido ao seu sistema de reposicionamento em grupo. O consumo de banda pode ser trabalhado de forma mais eficiente devido ao sistema de saltos entre os VANTs.

Do ponto de vista de segurança, a ausência de um ponto central de falha diminui o risco de ataques ao sistema. Na Figura 8 é possível ver que mesmo o VANT mais distante ainda é acessível por qualquer outro elemento na rede devido a comunicação direta V2V (KARAGIANNIS O. ALTINTAS, 2011).



Figura 8 – Comunicação VANT – VANT provê conectividade entre todos os elementos da rede.

Inúmeras aplicações vêm surgindo com a utilização de sistemas de VANTs colaborativos. Tarefas que necessitavam de helicópteros no passado para supervisionar linhas de transmissão de alta potência, consideradas tarefas altamente caras hoje em dia são facilmente realizadas por VANTs que enviam os dados diretamente para a ECT. Aplica-

ções de monitoramento ambiental, onde pode-se lançar um ou mais VANTs para análise meteorológica, realizando fotografias, medindo temperatura, humidade e velocidade do vento em vários níveis garantindo assim maior precisão nos dados coletados para posterior processamento. Acompanhamento e rastreamento de vida selvagem em florestas ou savanas, onde atualmente esse trabalho é realizado por pesquisadores, biólogos que acompanham a vida destes animais de perto em campo, expondo-se à riscos, além de vários outros tais como doenças do ambiente em que estão. Com câmeras térmicas é possível acompanhar esses animais à noite, inclusive monitorar possíveis ameaças à animais em extinção.

Como já comentado anteriormente, na agricultura VANTs tem auxiliado muitos agricultores na precisão da irrigação do solo, pulverização e análise do plantio de forma detalhada, através de fotos no espectro visível e invisível, dando ao agricultor um feedback preciso da qualidade da plantação e observações que devem ser tratadas de forma específica na lavoura (COELHO, 2015)(JORGE, 2014). Levando em consideração cenários assim geralmente possuem uma área muito extensa, a aplicação do conceito de uma UAANET é perfeitamente aplicável, realizando o serviço em muito menos tempo, dando ao agricultor tempo de sobra para lidar com os problemas específicos do plantio. Além disso, a rede ad-hoc torna-se ainda mais útil, pois áreas rurais geralmente carecem de cobertura de sinal por parte das operadoras, portanto uma rede que se autoconfigura, é essencial em cenários assim.

### 2.3.6 Sistemas Centralizados

De acordo com Maxa et al. (MAXA MOHAMED SLIM BEM MAHMOUD, 2017) um sistema de comunicação centralizado poderia ser implantado para uma UAANET. Cada VANT no ar teria um canal de comunicação direto com a ECT. Ou seja, a ECT é considerada um nó centralizador da rede, e comunica-se com todos os VANT de forma simultânea. Já a comunicação VANT – VANT não seria possível, pois toda comunicação V2V entre dois ou mais VANTs que estejam em ar devem obrigatoriamente passar pela ECT. Esse modelo de comunicação possui algumas restrições. Primeiramente um canal de comunicação deve ser reservado para cada VANT, portanto a escalabilidade do sistema é preocupante, pois existe limitação de banda sobre esse canal, e conforme entram mais VANTs no sistema, deve haver recurso para todos os VANTs na ECT, portanto deve-se evitar latência e atrasos por falta de banda disponível na ECT. Outro ponto negativo é o atraso na comunicação entre dois VANTs, pois a comunicação deverá sempre passar pela ECT.

Do ponto de vista da confiabilidade do sistema, problemas geográficos tais como edifícios, ou até montanhas criariam obstruções na comunicação entre o VANT e a ECT, forçando o equipamento a estar sempre em visada direta com a ECT conforme Figura 9. Outra possibilidade é aumentar a potência das antenas do VANT o que elevaria o consumo

de energia do mesmo, algo que não é interessante quando se trata de pequenos drones com capacidade limitada.



Figura 9 – Problemas na comunicação centralizada.

Além disso, um sistema centralizado, é um problema pois apresenta um sistema com ponto central de falha do ponto de vista de segurança. Em caso de problemas no sistema da ECT, ou até mesmo ataques sobre ela, os VANTs perderiam capacidade de comunicação de controle com a base, e comunicação VANT – VANT, que obrigatoriamente passa por ela.

Outra possibilidade levantada pelos autores Frew et al. em (FREW, 2009) e (MOHAMED J. AL-JAROODI, 2013) (MOHAMED J. AL-JAROODI, 2014), é a possibilidade de utilização da rede de telefonia celular existente, criando um sistema de várias células, provendo cobertura para toda a área necessária para que os VANTs tenham liberdade de voo pela ECT. Porém o problema é que cada antena trabalha em uma frequência diferente da sua antena vizinha, com o intuito de evitar interferência sobre as células vizinhas. Utilizando o conceito de células, proveniente da telefonia celular é possível fazer com que os VANTs operem da mesma forma que nossos smartphones, garantindo assim a comunicação constante entre VANT e ECT, enquanto houver cobertura.

Entretanto, o problema desse sistema é a estrutura necessária para que isso funcione. O custo de implantação representa um investimento muito alto dependendo da área de cobertura proposta, além disso, sistemas de VANTs não dão retorno financeiro em termos de comunicação para as operadoras. Ao se tratar de um cenário caótico, onde uma cidade foi destruída por furacões, tsunamis, ou guerra, é comum não haver energia elétrica em cenários assim, portanto não existe sistema de comunicação ou as antenas estão desligadas, impedindo assim que os *smartphones* e, claro, VANTs funcionem. Antenas de telefonia celular, poderiam ser atacadas facilmente desabilitando o sobrevoo de VANTs para aquela região, criando facilmente uma vulnerabilidade para o sistema (MAXA MOHAMED SLIM BEM MAHMOUD, 2017) daquela região.

Além disso, em cenários de florestas densas ou operações de busca e resgate onde não existe estrutura de telefonia celular, também cenários impossíveis para o sistema

centralizado. Uma solução seria utilização de um sistema de satélites, porém a potência de transmissão deve ser mais alta, impactando diretamente sobre o consumo de energia do equipamento.

## 2.4 Protocolos de Roteamento para Redes Móveis

Em redes UAANET, a alta mobilidade dos nós e sua constante mudança de topologia criam um cenário desafiador quando se pretende encaminhar informações através desses nós, entre origem e destino (MAXA G. ROUDIERE, 2015). Para que uma UAANET execute suas tarefas de forma bem-sucedida, tanto os VANTs quanto a ECT devem ser capazes de enviar, receber e repassar o tráfego da rede, seja de controle, ou mesmo de telemetria de cada VANT. Um protocolo de roteamento é necessário para localizar rotas entre dois pontos móveis, de uma forma que não cause impacto na performance da rede móvel (MAXA MOHAMED SLIM BEM MAHMOUD, 2017).

Além disso, devido ao movimento aleatório dos nós, a mudança da árvore é constante, o que dificulta criação e manutenção de rotas através de *multicasts* na rede, fazendo com que o percurso entre os nós de origem e destino possua uma ou mais rotas possíveis, certamente temporárias, tornando a arquitetura do protocolo de roteamento um desafio (MAXA G. ROUDIERE, 2015).

### 2.4.1 Protocolos Reativos

Para essa modalidade de protocolo de roteamento, uma rota é criada apenas sob demanda, ou seja, quando um determinado nó de origem necessita se comunicar com outro nó de destino. Para a criação desta rota, o nó de origem envia um pacote de RREQ – *Route Request*, que será encaminhado via *flooding* para os demais nós aos quais o originador encontra-se conectado. Esse pacote é repassado e a cada salto é acrescentado a informação do nó que o repassou. Este pacote de RREQ ao ser recebido pelo nó de destino, o mesmo envia um RREP – *Route Reply* para o nó de origem utilizando a rota de menor quantidade de saltos devido às marcações do histórico de saltos do RREQ, sendo assim utilizada para o envio dos dados da origem até o destino (MAXA MOHAMED SLIM BEM MAHMOUD, 2017).

Como o processo de descoberta de rota é sob demanda, é possível que ocorra latência até que a rota seja estabelecida, impactando no desempenho da rede de forma geral. Além disso, o processo de *flooding* (inundação) dos pacotes de RREQ na rede podem acarretar em congestionamento, e problemas *buffer overflow* dependendo da capacidade de memória nos nós (SHIRANI M. ST-HILAIRE, 2012).

### 2.4.2 Protocolos Proativos

Protocolos de roteamento proativos utilizam tabelas para armazenamento das informações de referente aos seus nós conectados. Essas tabelas se atualizam de forma periódica, ou em caso de mudanças de topologia da rede. Cada nó compartilha sua tabela de roteamento com seus nós vizinhos, para que atualizem suas tabelas de rota também e mantenham suas informações atualizadas de forma constante (MAXA G. ROUDIERE, 2015).

O critério para estabelecimento de uma rota entre origem e destino é o menor custo, menor caminho ou mesmo maior banda disponível entre os nós. Essa categoria de protocolo de roteamento utiliza métricas link state, ou distance vector para a elaboração da rota entre dois determinados pontos. Segundo Mohseni et al. (MOHSENI R. HASSAN, 2010), esta classe de algoritmo de roteamento não se adequa perfeitamente ao mundo móvel das UAANETs devido as rápidas mudanças na rede e baixa densidade de nós, segundo o autor o cabeçalho aumenta proporcionalmente em função da quantidade de nós e grau de mobilidade impactando em desempenho como um todo para o nó. Já foram realizados diversos experimentos e segundo os autores em (MAXA G. ROUDIERE, 2015) já foi comprovado que em redes UAANET protocolos proativos tendem a gerar muito mais cabeçalhos do que protocolos reativos.

A vantagem dos protocolos proativos é que eles possuem uma tabela de roteamento entre quaisquer nós na rede sempre atualizada, não havendo assim latência ao enviar dados ao destinatário, pois a rota já está previamente definida para cada nó da rede.

### 2.4.3 Protocolos Geográficos

Protocolo de Roteamento Geográfico (GRP) é um protocolo de roteamento baseado em informações geográficas baseado sobre dois pilares: (1) cada nó possui conhecimento de sua própria localização, além da localização de seus vizinhos imediatos, e (2) o nó de origem sabe onde o nó de destino se encontra e quem ele é. A localização dos vizinhos é atualizada constantemente através de pacotes de *Hello*. Os pacotes de dados são roteados através da rede utilizando sempre a localização do nó destino como referência. O GRP não opera com tabelas de roteamento, ele se baseia na premissa de que o roteamento para o nó de destino se dará baseado no que cada nó sabe a respeito de seus vizinhos. Os protocolos de roteamento geográficos mais conhecidos são o *Greedy* e o *Face Routing* (MAXA MOHAMED SLIM BEM MAHMOUD, 2017) (LIN Q. SUN, 2012).

No *Greedy*, o pacote é levado para mais perto do nó de destino à cada salto de forma que o pacote sempre se aproxime do destinatário. Esse sistema costuma falhar apenas quando não existe um vizinho mais próximo do nó destino que o próprio nó atual. Quando isso ocorre, o modo de encaminhamento *Greedy* se altera, e o próximo salto é selecionado

de forma que a informação atravessasse o perímetro da região onde o *Greedy* falhou. Ele continua encaminhando dessa forma até que encontre um nó vizinho que se enquadre novamente no parâmetro *Greedy*, retomando assim esse modo de encaminhamento.

Já o algoritmo *Face Routing* segundo o autor (LIN Q. SUN, 2012) diz que as regiões são consideradas separadas pelas arestas de um gráfico planar. O algoritmo encaminha os dados por tais arestas sempre para o próximo salto mais próximo do destinatário. Esse algoritmo sempre encontra uma rota para o destino.

VANTs são geralmente equipados com sistemas de GPS, é possível utilizar a telemetria da posição de onde o mesmo se encontra, além de sua proximidade, velocidade e direção. Protocolos de roteamento para redes *ad-hoc* tais como o GPSR, GPMOR e MPGR aproveitam-se dessas variáveis para tomar decisões no momento de encaminhar os dados para um ou para outro nó que estiverem em seu alcance.

#### 2.4.4 Protocolos Híbridos

A performance das redes *ad-hoc* dificilmente pode ser modelada devido a quantidade de variáveis e seus valores incertos. Como visto na literatura pesquisada, várias linhas de estudos tentam prover performance de redes *ad-hoc* baseando-se em modelos imprecisos, estabelecimento de links constantemente se criam se destroem em questão de segundos ou menos. Os modelos existentes são imprecisos devido ao comportamento imprevisível dos nós na rede, portanto modelos reativos, pró-ativos, ou mesmo geográficos muitas vezes não são capazes de modelar seu ideal funcionamento para encaminhar mensagens de forma confiável (TAMINE, 2012).

Portanto, modelos híbridos de protocolos de roteamento, aproveitando as melhores características de cada sistema já estudado pode ser colocado em prática para melhorar questões de performance na rede como um todo.

#### 2.4.5 DTN – *Delay Tolerant Networks*

Redes tolerantes a atrasos e desconexões tem ganho muito espaço com a entrada de nós móveis, provendo comunicação mesmo em cenários onde não existe conexão. Seu conceito consiste em um nó que carregue determinada informação, ao se reencontrar com a rede possa transmitir ainda essas informações, não havendo necessidade de comunicação direta entre origem e destino (MAXA MOHAMED SLIM BEM MAHMOUD, 2017).

Um exemplo desse sistema é um VANT que antes de decolar do chão recebe suas instruções de voo através de uma programação fixa, além de dados para serem compartilhados com demais VANTs. Assim que ele sobe, e entre em contato com um segundo VANT, ele transmite as informações para este que se desloca até entrar em contato com demais VANTs. Feito isso, ele copia, ou transmite as informações para estes VANTs.

Cada VANT é responsável por receber as informações, carregá-las e repassá-las para os demais VANTs que entrarem em sua área de cobertura. A Figura 10 apresenta um esboço desta arquitetura. No cenário abaixo é importante deixar claro que não existe comunicação fim-a-fim entre a ECT e todos os VANTs.

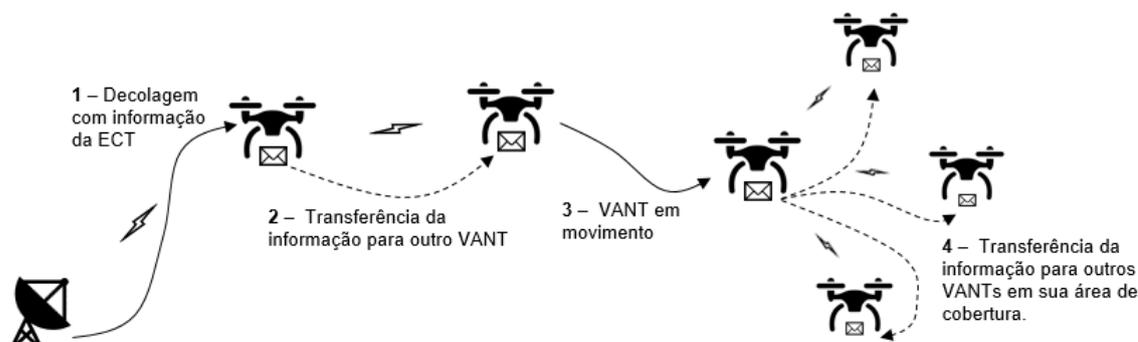


Figura 10 – Comunicação em redes DTN.

## 2.5 Modelos de Mobilidade

Os modelos de mobilidade são indispensáveis para o estudo de redes móveis, diversos estudos apontam que a performance da rede varia significativamente de acordo com o padrão de mobilidade dos nós (BOUACHIR A. ABRASSART, 2014) (KUIPER, 2006). O padrão ou modelo de mobilidade utilizado para redes UAANET depende muito do tipo de aplicação ao qual ela será destinada, por exemplo, um modelo de mobilidade que permita que os nós se movam de forma independente é mais interessante para cenários de busca e resgate, ou mapeamento de uma determinada área.

Os modelos de mobilidade podem ser classificados em dois grupos que são: modelo de mobilidade de trajeto pré-definido, e modelo de mobilidade aleatório. Um carro percorrendo uma avenida dentro de uma cidade, ou mesmo uma rodovia é um caminho do qual o carro está confinado, portanto sua trajetória é conhecida e previsível. Já o modelo de mobilidade aleatório é aplicado onde o conhecimento sobre a área a ser percorrida é mínima, ou seja, em operações de busca e resgate em alto mar, ou mesmo em florestas, o padrão de movimentação dos VANTs será definido durante o próprio percurso. Além disso, os padrões de movimentação aleatórios são classificados em dois grupos: padrão baseado em grupo, e em entidade/nó. O modelo de mobilidade em grupo faz com que os nós se movam de forma agrupada, seguindo um líder. Já o modelo de mobilidade em entidade, os nós movem-se de forma independente um do outro (BIOMO T. KUNZ, 2015).

Outra categoria de modelo de mobilidade é baseada na natureza das entidades que geram esse padrão de movimento. O critério para essa categorização não é a quantidade de

entidades, mas sim seu comportamento. Modelos de mobilidade descrevem seres humanos em certos cenários, por exemplo empregados num escritório podem ser atribuídos aos modelos *Random WayPoint* ou *Random Walk*. Modelos de mobilidade que descrevem comportamentos animais andando em hordas ou bandos, por exemplo o *Random Walk*, *Pursue* ou *Nomadic Community*. Modelos para carros em uma cidade, ou mesmo pedestres em uma calçada que geralmente possuem restrições para seus movimentos, ou seja, devem seguir um curso pré-definido podem ser moldados pelos modelos *Freeway Model* ou *Manhattan Model* (FUJIMOTO K. S. PERUMALLA, 2007).

Para este trabalho, iremos tratar apenas padrões de movimentação bidirecionais, ou seja, 2D. O motivo dessa escolha é que podemos levar em consideração que os VANTs estão operando no mesmo nível de altitude, além disso, situações como decolagem e pouso não precisam estar modeladas nesses cenários.

### 2.5.1 *Random WayPoint*

Esse modelo de mobilidade é frequentemente utilizado devido sua simplicidade e presença em simuladores tais como o NS2 (NS2, 2018), NS3 (NS3, 2018), e o OPNET (OPNET, 2018). Seu funcionamento segue o seguinte princípio, um nó escolhe uma localização de forma aleatória dentro da área de simulação e move-se para aquele ponto em linha reta, utilizando uma velocidade aleatória. Ao chegar ao local escolhido, o nó pausa e sorteia uma nova localização dentro da área de simulação, e assim por diante (BIOMO T. KUNZ, 2015).

É importante ressaltar que esse modelo de mobilidade nem sempre retrata um cenário realístico, por exemplo, na Figura 11 é possível visualizar ângulos agudos (pontos 4, 6, 7 e 8) em suas trajetórias de um ponto aleatório a outro, uma situação impossível para um VANT de asa fixa que possui restrições aerodinâmicas para mudanças bruscas de direção, porém viável para um VANT de asa rotativa.

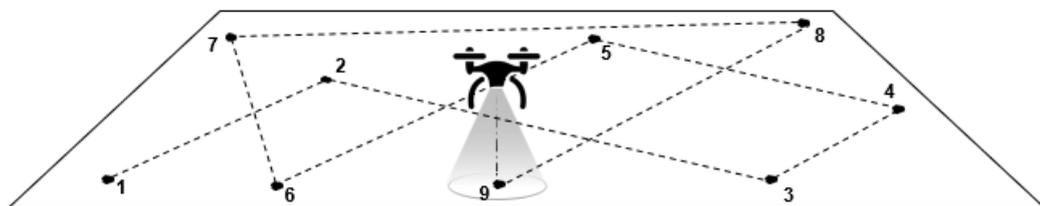


Figura 11 – Modelo de mobilidade *Random WayPoint*.

## 3 Trabalhos Relacionados

Este capítulo tem como objetivo apresentar alguns trabalhos já publicados relacionados de forma mais direta ao tema proposto, além de apresentar importantes conceitos que serão abordados em seguida.

A seção 3.1 aborda a questão do planejamento de percurso, de maneira que os VANTs sejam capazes de cobrir colaborativamente uma determinada área. Na seção 3.2 são abordados temas como SRP – *Swarm Routing Problem*. Na seção 3.3 entramos no assunto de *middleware* para VANTs Colaborativos, facilitando seu gerenciamento. Já na seção 3.4 tratamos sobre alguns dos principais simuladores de mobilidade existentes na literatura, e na seção 3.5 vamos falar sobre outros assuntos relacionados ao tema estudado.

### 3.1 Planejamento de Percurso

Planejamento de percurso é fundamental em vários sistemas de automação, presentes em nosso dia-a-dia, seja na área de robótica, operações militares, busca e resgate e logística. O planejamento de percurso para um único agente é de uma certa forma simples, ao ser comparado com um sistema de múltiplos agentes. Em um cenário onde temos mais de um agente se movendo simultaneamente dentro de uma área compartilhada, o problema se torna mais complexo (WANG, 2011). Os trajetos devem ser distribuídos entre os agentes de forma dinâmica para que seja realizado de forma mais rápida. Isso deve envolver o tratamento de problemas tais como sobrevoo de áreas já percorridas por outros nós, tratamento de colisões com pessoas, veículos, prédios, ou mesmo colisão com outros VANTs.

Com a rápida adoção de sistemas de automação em áreas agrícolas, sistemas de planejamento de percurso tornaram-se um grande diferencial quando se trata de cobrir toda a área da lavoura de forma eficiente. Seja para fotografias, pulverização ou coleta de dados. A automatização do percurso deve ser realizada de forma eficiente, minimizando o tempo e o percurso, além de se adaptar de forma dinâmica aos obstáculos (fixos ou móveis) apresentados pela área a ser percorrida (JIN, 2010). A forma mais eficiente de percurso levará a menos consumo de energia por parte dos VANTs, o que agrega valor dado que seus recursos são limitados.

Além disso, o planejamento de percurso depende diretamente do modelo de mobilidade que é atribuído aos VANTs, e como já vimos anteriormente, o modelo de mobilidade impacta diretamente na performance da rede UAANET.

Vários autores na literatura apontam o modelo de mobilidade *Scanning* ou também conhecida como trapezoidal para mapeamento de superfícies. Os autores citam que a área

total é decomposta em pequenas áreas, criando assim um mapa discreto de toda região (CHAVES, 2013) (JORGE, 2014) (WANG, 2011) (JIN, 2010) (SUJIT, 2009). Conforme os VANTs avançam em seu trajeto/percurso, vão alterando o status dessas pequenas áreas discretas de modo a informar o que já foi percorrido, para sua própria referência e para os demais VANTs, evitando assim problemas de sobrevoos nas áreas já percorridas. A Figura 12 exemplifica, em destaque as subáreas já percorridas das que se encontram ainda pendentes, em branco.

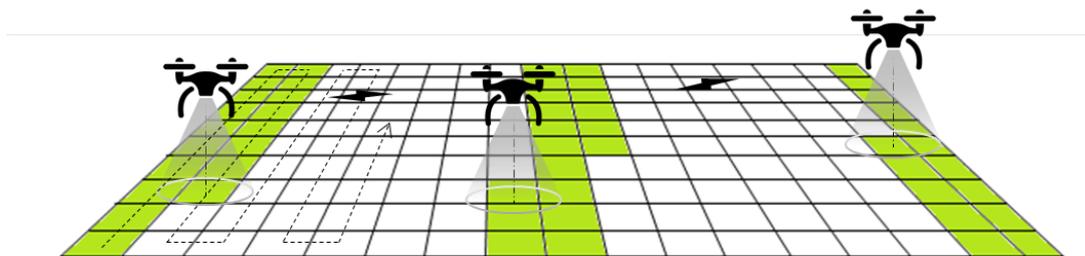


Figura 12 – Sistema de subáreas criados para mapear a superfície.

Em (SUJIT, 2009) os autores Sujit e Beard implementam um algoritmo chamado *Anytime* capaz de lidar com as dificuldades em dividir a área a ser percorrida em diversas subáreas, fazendo a rede UAANET partir de um ponto inicial até o ponto final sendo capaz de lidar com as dificuldades de detectar a presença de obstáculos fixos e dinâmicos, fazendo com que o sistema desvie de tais objetos, e recalcule seu trajeto de acordo com tais desvios. Em (WANG, 2011) os autores Wang e Botea apresentam um algoritmo chamado MAPP (*Multi Agent Path Planning*) onde o índice de precisão chega à 99,7% da área mapeada, com escalabilidade de até 2000 VANTs, realizando comparações entre outros algoritmos tais como o FAR e o WHCA, levando em consideração o desvio de objetos durante a área a ser percorrida. Os autores aprofundam-se na questão do mapeamento da área, porém nada falam a respeito do modo de comunicação entre os VANTs, ou seja, os aspectos de rede, a comunicação entre os VANTs ou mesmo a coordenação das tarefas entre eles não são aspectos abordados.

### 3.2 SRP – *Swarm Routing Problem*

Em (ADAM, 2008) os autores modelam uma solução de problema de rota para grupos de VANTs dentro de uma janela de tempo, criando assim o conceito de SRP - *Swarm Routing Problem*, adaptando assim o conceito de VRPTW - *Vehicle Routing Problem with Time Windows* (TOTH P., 2001) para VANTs colaborativos.

O VRPTW gira em torno de uma quantidade de veículos, alvos distribuídos e conhecidos que são os locais onde os veículos devem passar dentro de uma janela de tempo, além da restrição de capacidade de percurso de cada veículo visitar, e um depósito. Este

depósito é considerado o ponto inicial e final de todas as unidades que percorrerão os alvos. Cada alvo possui sua própria posição euclidiana, uma janela de tempo de visita e um caminho para o cada próximo salto (alvo). O objetivo do problema é desenvolver uma série de rotas para cada veículo, de forma que todos os alvos sejam visitados dentro de uma determinada janela de tempo, e todos os veículos retornem ao depósito a tempo. Cada veículo no problema possui sua capacidade de visitação que não deve ser excedida durante a visitação de seus respectivos alvos (ADAM, 2008).

Segundo a autora Amanda et al. (WEINSTEIN, 2007), em sua publicação ela afirma que independente da tarefa a ser concluída, é importante que este pequeno grupo de VANTs possam cooperar entre si de forma eficiente para que os alvos sejam percorridos sempre de forma eficiente.

De acordo com Tooth e Vigo (TOTH P., 2001), em praticamente todas as variações do VRP - *Vehicle Routing Problem*, ou mesmo o VRPTW assume-se que todos os veículos partem de um único local para diferentes alvos, e apenas um veículo visita cada alvo. Porém ao lidarmos com grupos de VANTs e vários alvos a serem percorridos, é altamente desejável que todos os VANTs percorram seus respectivos alvos da forma mais eficiente possível. Levando isso em consideração, podemos garantir maior eficiência no sistema ao trabalharmos com grupos dinâmicos de VANTs, ao invés de considerá-los unidades indivisíveis que apenas podem visitar um local por vez. As consequências deste parâmetro implicam na vantagem da divisibilidade do grupo e rotear tais subgrupos para diferentes alvos, reagrupando-os em outros locais caso necessário.

Os autores em (TOTH P., 2001) descrevem o SRP sendo como um grafo  $G = (V, A)$ , onde  $V = v_0, v_1, \dots, v_n$  e  $v_0$  sendo o depósito inicial de partida dos VANTs. O conjunto de vértices do grafo é definido por  $A = (v_i, v_j), \in V, i \neq j$ , onde cada vértice possui um custo associado  $c(v_i, v_j)$ , sendo este o custo o tempo de deslocamento do alvo  $i$  até o alvo  $j$ , levando em consideração uma velocidade média para os VANTs. Além disso, a janela de tempo existe para todos os alvos, onde  $E$  representa o início da abertura desta janela, e  $L$  a última janela disponível onde um VANT ainda pode chegar e completar o serviço, que é dado por  $S$ . Caso o veículo chegue a alvo antes de  $E$ , acarretará um tempo de espera até sua abertura  $W$ , que é dado pela diferença entre o tempo de chegada e  $E$ .

Sendo assim o tempo total de percurso do veículo por rota é dado pela somatória entre todos os custos, tempos de espera, e serviço por alvo, resultando assim na definição do tempo total de percurso por VANT ( $\sum c_{ij} + \sum W_i + \sum S_i$ ). Além disso, devemos considerar que o tempo total não deve exceder o tempo final de chegada ao depósito.

Em (WEINSTEIN, 2007) os autores assumem que existam  $N$  alvos a serem percorridos, e  $K$  VANTs, levando em consideração que  $K \leq N$ , todos os alvos são conhecidos, e cada alvo requer a passagem de apenas um único VANT. Além disso, eles destacam o tema MDVRP - *Multi Depot Vehicle Routing Problem*, trabalhando a questão de que os

VANTs não necessariamente precisam retornar para o ponto de origem, distribuindo assim vários depósitos, locais onde considera-se o pouso dos mesmos. Além disso, os autores ressaltam em suas conclusões que o custo computacional para realização do cálculo da minimização do tempo total é muito alto, portanto, os autores recomendam a utilização de restrições em suas teses.

Segundo os autores em (ADAM, 2008) definem que é vantajoso a minimização destes três itens: total do percurso, quantidade de VANTs e tempo de espera por alvo. Ao otimizar tais variáveis, serão apresentados melhores resultados para o problema, dados pelas equações 3.1, 3.2 e 3.3.

Abaixo é apresentada a equação referente à minimização do percurso total do VANT.

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \quad (3.1)$$

Abaixo é apresentada a equação referente à minimização da quantidade de VANTs utilizados para o percurso.

$$\min \sum_{k=0}^K k \quad (3.2)$$

Abaixo é apresentada a equação referente à minimização do tempo de espera sobre o alvo.

$$\min \sum_{k \in K} \sum_{(i,j) \in A} t_{ik} x_{ijk} \quad (3.3)$$

Para a equação 3.3, temos a definição do tempo de espera  $t$  definido na equação 3.3, sendo este o tempo de espera do VANT sobre o alvo. Para o SRP, o tempo de espera dado na Equação 3.3 é definido pela diferença dos tempos de chegada de um VANT e o tempo de chegada do último, caso o tempo de chegada do último VANT seja após o tempo de início de serviço do alvo.

Os autores (ADAM, 2008) concluem que o modelo SRP apresenta soluções não apenas teóricas, mas também aplicações práticas para diversos modelos e cenários reais.

### 3.3 *Middleware* para VANTs Colaborativos

Aplicações para VANTs colaborativos já foram discutidas e exemplificadas nessa proposta de projeto, sistemas autônomos ou semiautônomos que operam em conjunto de forma a alcançar um objetivo tem muito a oferecer na área de automação de serviços. Porém, construir aplicações sobre um conjunto de dispositivos móveis é um desafio que geralmente envolve uma grande quantidade de mão de obra para desenvolvimento de sistemas de comunicação, e testes desse complexo conjunto (MOHAMED J. AL-JAROODI, 2013).

É importante que haja uma plataforma de serviços que abstraia todos os recursos envolvidos aos VANTs colaborativos, que torne simples a adesão de um novo nó na rede, que lide com a heterogeneidade de cada equipamento, seus recursos, câmeras, sensores, motores, padrões de movimentação e crie um único ponto de controle para lidar com todos os recursos possíveis de um sistema colaborativo. O autor Mohamed et al. (MOHAMED J. AL-JAROODI, 2013) defendem o conceito da criação de um middleware que crie uma camada de abstração entre o equipamento e o desenvolvedor, em casos de sistemas heterogêneos ou não, de modo a atrair a atenção de desenvolvedores e engenheiros para que adaptem seus novos recursos à este middleware, facilitando assim a interoperabilidade de todo o conjunto e a absorção das novas tecnologias no sistema.

Ao desenvolver um sistema de grande escala, distribuído, sem a presença de um *middleware* é um trabalho extremamente complexo. O autor em (MOHAMED J. AL-JAROODI, 2013) cita que “... *tentar desenvolver uma aplicação distribuída sem um middleware, é como tentar desenvolver uma simples aplicação em um computador sem um sistema operacional ...*”. O autor ainda explica que um middleware para VANTs colaborativos pode oferecer uma série de vantagens:

- Oferecer ferramentas e funções que simplifiquem o desenvolvimento de aplicações colaborativas para VANTs.
- Oferecer alto nível de abstração e interfaces que facilitem a integração, reutilização e desenvolvimento de recursos para VANTs colaborativos.
- Abstrair a heterogeneidade entre hardware, software e sistemas de diferentes fabricantes, seja dos VANTs em si, ou de seus recursos sensoriais.
- Abstrair os detalhes específicos da comunicação colaborativa.
- Prover uma arquitetura comum para adicionar novos serviços para o grupo, sem a necessidade de alterar o código de cada VANT.
- Oferecer segurança, confiabilidade e qualidade de serviço na comunicação.
- Prover as ferramentas necessárias para a melhorar a comunicação, e estabilidade do sistema, garantindo sua segurança e escalabilidade para novas aplicações para VANTs colaborativos.

### 3.4 Simuladores de Mobilidade

Pesquisas sobre redes *ad-hoc* e redes tolerantes à atrasos e desconexões geralmente recaem sobre simuladores devido à sua escalabilidade, facilidade em lidar com novos padrões de mobilidade, protocolos de roteamento, além dos custos envolvidos em realizar

testes com VANTs em cenários realísticos, tais como mar aberto, falhas no sistema destes equipamentos podem levar a custos muito altos (BAUMANN F. LEGENDRE, 2008). Os simuladores e modelos de mobilidade evoluíram de simples modelos sintéticos para modelos cada vez mais realísticos, e para obtermos maior precisão nos resultados redes UAANET, é importante que tenhamos simuladores precisos e confiáveis.

Registros reais dos padrões de movimentação dos nós são as melhores representações da realidade, porém difíceis de se obter (FUJIMOTO K. S. PERUMALLA, 2007). Atualmente existem diversos simuladores e vários modelos de mobilidade, simulando modelos reais, e sintéticos.

### 3.4.1 NS2 - *Network Simulator 2*

O NS2 – *The Network Simulator 2*, é um simulador baseado em eventos discretos, código aberto baseado em Linux, orientado para projetos de pesquisa. Ele provê todas as funcionalidades e recursos necessários para os protocolos TCP, UDP e RTP sobre redes cabeadas e sem fio, além de suporte a roteamento, multicast e protocolo IP. Além de possuir todas estas ferramentas que o tornam muito útil em cenários de simulações, ele ainda oferece a possibilidade de análise local do tráfego em cada nó (NS2, 2018).

O NS2 iniciou-se como uma variação do projeto *REAL Network Simulator* em 1989, onde este era um simulador originalmente desenvolvido para estudos de comportamentos dinâmicos e controle congestionamentos em redes.

O NS2 tem sido utilizado por inúmeros pesquisadores em seus projetos (BAUMANN F. LEGENDRE, 2008) (TAMINE, 2012) (JIN, 2010) para lidar com modelos de mobilidade, e protocolos de roteamento sobre redes *ad-hoc*, além de fornecer todas as ferramentas necessárias através de seu código aberto em linguagem de programação C++ (C++, 2018) para todos os tipos de pesquisa.

### 3.4.2 MATLAB

O MATLAB, desenvolvido pela empresa MathWorks (MATLAB, 2018) é uma das principais desenvolvedoras de sistemas matemáticos computacionais, engenheiros, cientistas e pesquisadores utilizam essa ferramenta e seus produtos relacionados em vários países para acelerar seus projetos de pesquisa e estudos.

O MATLAB é um sistema possui vários módulos voltados para tecnologia de sistemas sem fio, o permite provar conceitos de algoritmos através de simulações, tanto em hardware quanto em software. Várias bibliotecas desenvolvidas em MATLAB para comunicação sem fio, tecnologias 4G e 5G, modelagem de antenas, além da criação de protótipos e validação dos mesmos.

Alguns autores em nossa base de pesquisa utilizaram o MATLAB para suas simulações, e através da comparação dos resultados com ambientes reais, foi possível comprovar seu modelo de busca (CHAVES, 2013) (BIOMO T. KUNZ, 2015) (TEICHMANN M. DORDA, 2014).

### 3.4.3 *The One*

O The ONE - *The Opportunistic Network Environment* é um simulador baseado em eventos discretos, utiliza nós, ou também chamados como agentes. Seus principais recursos são a modelagem do padrão de movimento dos nós, contato entre os nós, roteamento e tratamento de mensagens entre si. A coleta dos resultados, e análise é dada através de uma interface de visualização, relatórios e ferramentas de pós-processamento (THEONE, 2018).

Desenvolvido pela SINDTN e CATDTN, ambos projetos fomentados pela Nokia *Research Center* ele é voltado para pesquisas, simulações e estudos sobre redes DTN – *Delayed Tolerant Networks*.

Vários estudos sobre redes tolerantes a atrasos e desconexões já foram realizados utilizando essa ferramenta (DESTA ARI KERÄNEN, 2013) (KERÄNEN TEEMU KÄRKKÄLÄNEN, 2010) (KERÄNEN JÖRG OTT, 2009) e tratam sobre protocolos de roteamento baseado em redes/nós móveis que possuem a característica de possibilidade de desconexão do restante do grupo.

### 3.4.4 *OMNET++*

O simulador de redes OMNET++ - Objective Modular Network Testbed in C++, é um sistema modular, discreto, extensível, baseado em componentes criados através de bibliotecas em linguagem de programação C++ (C++, 2018) inicialmente desenvolvido para simulações de redes cabeadas e sem fio, aplicações embarcadas, simulações de congestionamento de redes, sistemas ad-hoc, protocolos de internet e roteamento (OMNET++, 2018).

O OMNET++ ganhou muita popularidade por não ser apenas uma poderosa plataforma de simulação, mas também um ambiente de desenvolvimento completo que é cada vez mais utilizado na comunidade científica e engenheiros em meio industrial para simulação de cenários específicos. Foi desenvolvido inicialmente para cenários acadêmicos e de pesquisa, criado por György Pongor. O sistema utiliza plataforma Windows, Linux e MAC OS. Diversos estudos sobre redes ad-hoc, modelos de mobilidade e protocolos de roteamento foram conduzidos utilizando o OMNET++ (KUIPER, 2006) (BUJARI C. T. CALAFATE, 2017) (MAXA M. SLIM, 2016).

### 3.4.5 Outros Trabalhos Relacionados

Em (KASHUBA V. I. NOVIKOV, 2015) os autores Kashuba et al. estudam uma diferente abordagem para redução do planejamento de percurso por parte de VANT utilizando a trajetória em linha reta referente ao ângulo bissetriz entre os dois próximos sensores em relação à posição atual do VANT. Os autores comparam resultados obtidos de seu percurso com o nominal, sendo que o VANT percorra cada posição de sensor, conseguindo uma redução de quase 25% do percurso do VANT.

Wang et al. (WANG F. MA, 2015) em sua pesquisa estudam a utilizam de VANTs para coleta de dados sensoriais utilizando uma abordagem de FPPWR – *Fast Path-Planning With Rules*, onde utilizam nós móveis (VANTs) para coleta dos dados da rede de sensores utilizando *Cluster Headers* (CHs), dividindo a área em questão em grids, reduzindo significativamente o percurso redundante do VANT na coleta dos dados sensoriais. Os autores desenvolveram seu próprio simulador para realização das simulações e tiveram resultados favoráveis em relação ao volume de dados coletados, redução de percurso por parte do VANT além do tempo necessário para coleta desses dados sensoriais.

Em (JO J. HEO, 2018) os autores trabalham em um novo modelo de *Rendezvous* baseando-se na trajetória do VANT e velocidade do VANT garantindo assim melhor taxa de aquisição de dados por parte das estações de coleta distribuídas na região de interesse. Os resultados apresentados pelos autores são focados em taxa de sucesso em relação à conexão do VANT com os sensores, em função da variação da velocidade do nó móvel. Os autores utilizam um simulador WSN, conhecido como SIDnet-SWANS (GHICA, 2008) para obtenção dos resultados apresentados.

Em (XIONG, 2018a) os autores Xiong et al. realizam estudos sobre o agendamento do horário de transmissão dos sensores em terra, com o percurso do VANT, justamente para redução do consumo de energia por parte dos sensores. Os autores focam seus resultados no planejamento de percurso do VANT, e no agendamento do horário de transmissão dos sensores, em função desse percurso planejado. Os autores limitam seus temas de estudo em uma trajetória retilínea, com curvas suaves tais como um rio, ou mesmo um oleoduto, criando uma série de *way-points* que serão os pontos onde o VANT fará suas correções de rota. Os autores citam um algoritmo chamado Water-tank que resolve o problema do agendamento do período de transmissão dos sensores de acordo com o percurso planejado do VANT, baseando-se em timeslots. Seus resultados são baseados em raio de comunicação dos sensores, velocidade do VANT, consumo de energia e coleta de dados por parte do VANT.

Em (ZHAN Y. ZENG, 2018) Zhan et al. estudam a utilização de VANTs para coleta de dados sensoriais, utilizando a programação do tempo de *wake-up* das estações de coleta de acordo com a passagem do VANT, porém levam em consideração questões como

consumo de energia, e um estudo aprofundado da modelagem do canal de comunicação entre VANT e EC em função da velocidade do VANT, tamanho da informação à ser transmitida e potência de transmissão/recepção.

Em (RASHED, 2017) Rashed et al. realizam estudos sobre um VANT percorrendo uma área de  $2000 \times 2000 \text{ m}^2$  para mapeamento de sensores terrestres, porém utilizando um sistema de beacon para mapeamento destes sensores, porém também trabalhando com *cluster headers* (CH) para centralização e agrupamento dos sensores. Os autores exploram diversos modelos de mobilidade para o nó móvel (VANT) e apresentam resultados gráficos médios sobre 10 simulações para cada cenário em seu trabalho. Em (OLIVIERI, 2017) Olivieri et al. exploram as vantagens em se trabalhar com VANTs colaborativos para coleta de dados sensoriais em redes tolerantes a atrasos e interrupções. Os autores citam diversas vantagens em utilização de veículos aéreos, em relação a veículos terrestres, e propõem diversos modelos de redução de percurso para problemas como TSP (*Travel Salesman Problem*) e VRP (*Vehicle Routing Problem*). Ainda assim, os autores apresentam diversos resultados gráficos, tais como a taxa de vigilância para 2, 4, 8 e 16 nós, aplicando conceitos de *rendezvous* para encontro do VANT com a rede WSN.

Os próprios autores afirmam que estudos nessa área são novos e desafiadores, e não há muito conteúdo ainda em torno do agendamento do período de transmissão dos sensores de acordo com a rota do VANT. Artigos específicos referente utilização de VANTs para coleta de dados sensoriais, programação de sensores através de *timeslots* para gerenciamento do tempo de transmissão/hibernação das ECs, que utilizam um sistema de restrição e permissão não foram encontrados para estas aplicações, porém trabalhos semelhantes, em partes foram aplicados para trabalhar a aplicação deste conceito.

## 4 UAV Simulator

O *UAV Simulator* (UAVSIMULATOR, 2019) foi desenvolvido para sistemas operacionais Windows, em linguagem C# (C-Sharp) através do IDE *Microsoft Visual Studio 2017 Enterprise* (MICROSOFT, 2018) versão 15.5.7 e possui diversas funcionalidades para planejamento de percurso. Além disso, conta com recursos tais como simulação em tempo real de deslocamento de VANTs Colaborativos sobre uma rede de sensores para coleta de dados sensoriais, e entrega dos mesmos ao depósito, que seriam o destino para processamento desses dados.

O motivo da escolha do desenvolvimento do *UAV Simulator* foi a maior flexibilidade para adaptação dos recursos propostos para o tema do projeto, tendo em vista que os demais simuladores pesquisados também demandariam adaptações através de códigos e bibliotecas para o nosso tema. Como já possuíamos experiência em desenvolvimento de simuladores, modelos de mobilidade e protocolo de roteamento sobre redes móveis através de projetos de pesquisa anteriores, optamos por desenvolver nossa própria ferramenta para a modelagem do nosso trabalho. A Figura 13 apresenta sua tela principal.

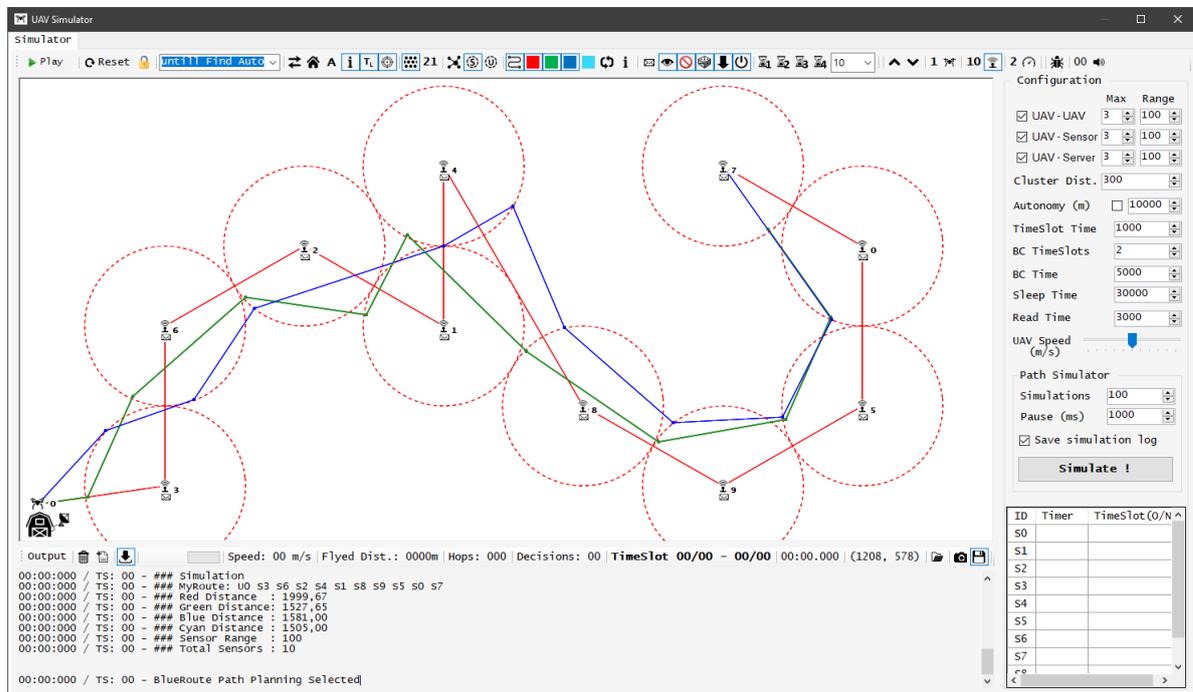


Figura 13 – Tela Inicial do *UAV Simulator*.

Na Figura 14 temos algumas das principais funcionalidades apresentadas em sua barra superior, responsável pelas variáveis do ambiente de simulação.

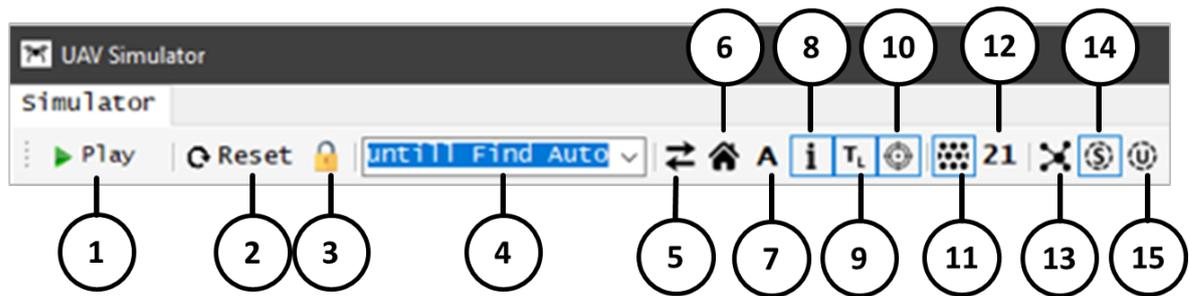


Figura 14 – Funcionalidades do *UAV Simulator*.

1. Inicia e Para a simulação em tempo real.
2. Exclui todos os objetos internos (VANT, EC) e recria o cenário com as mesmas quantidades.
3. Trava o *Reset*, ou seja, apenas reseta os atributos internos dos objetos e os reposiciona em suas posições originais, útil para repetição de cenários e simulações comparativas.
4. Seleção do modelo de mobilidade do VANT, *Random Direction*, *Random WayPoint*, *Wait Untill Find*, *uTimeslots*.
5. Alterna VANT titular com reserva.
6. Habilita o VANT retornar para o depósito quando finalizar sua rota padrão.
7. Alinha todos os VANTs junto ao depósito, além disso, reseta seus atributos internos.
8. Habilita a apresentação da ID dos objetos no simulador.
9. Habilita o tempo de leitura que o VANT aguarda sobre o objeto para ler os dados.
10. Habilita a apresentação do ponto de referência (*target*) do VANT em seu deslocamento.
11. Habilita o algoritmo para que as ECs não se sobreponham em relação a área de cobertura.
12. Apresenta a quantidade máxima de ECs que podem estar habilitadas para esta área de simulação e raio de comunicação das ECs.
13. Habilita o sistema de agrupamento em clusters por parte das ECs e VANTs.
14. Habilita o desenho do raio/limite de comunicação das ECs.
15. Habilita o desenho do raio/limite de comunicação dos VANTs.

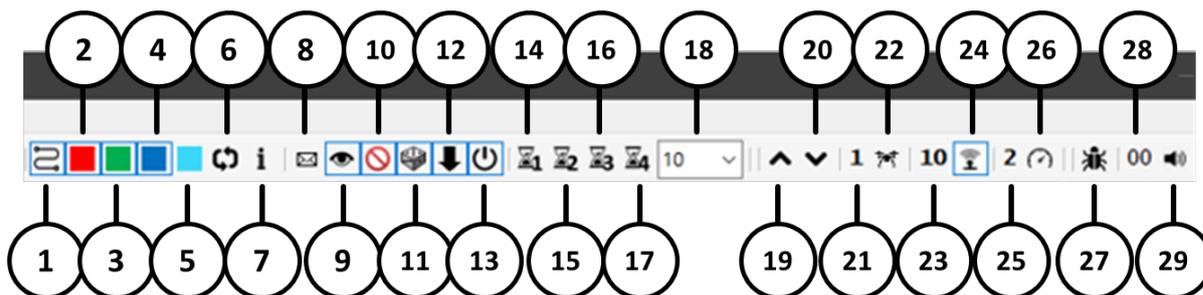


Figura 15 – Funcionalidades do *UAV Simulator*.

Continuando a barra superior do UAV Simulator, temos as demais funcionalidades apresentadas na Figura 15.

1. Habilita a apresentação das rotas do VANT (itens 2 à 5 da Figura 15)
2. Habilita a rota nominal do VANT, fazendo com que o VANT sobrevoe sobre cada sensor, e não sua região de cobertura
3. Habilita a rota *NearestPoint*
4. Habilita a rota Kashuba
5. Habilita a rota Híbrida (Kashuba x *NearestPoint*)
6. Habilita que a rota do último sensor até o ponto de partida
7. Habilita informações sobre a rota, tais como as principais coordenadas, além da distância percorrida pelo VANT até cada ponto.
8. Habilita que o VANT se desloque até a EC em timeslots já conhecidos, e não fique aguardando até o *timeslot* passar.
9. Habilita que o VANT esteja buscando continuamente por ECs, não apenas em coordenadas dos sensores.
10. Habilita o sistema de restrição de *timeslots*, o VANT consulta sua base de dados para tomada de decisões quando habilitado.
11. Habilita o uso de *timeslots* repetidos ou aleatórios para as ECs.
12. Habilita a reprogramação do timeslot da EC por parte do VANT.
13. Habilita o desligamento da EC por parte do VANT após a leitura dos dados.
14. Habilita todos os sensores para entrarem simultaneamente em modo de transmissão e hibernação de acordo com os valores informados.

15. Habilita que o VANT programe a EC de acordo com seu percurso, baseado nos valores informados pelo simulador.
16. Habilita que o VANT programe a EC de acordo com seu *timeslot*. Neste cenário, é considerado a quantidade de *timeslots* igual a quantidade de EC, sendo que cada EC é programada para seu próprio *timeslot*, sem repetição.
17. Habilita o sistema de *timeslots* do cenário baseando-se na quantidade informada pelo usuário no item 18 da Figura 15. Sendo assim, o VANT pode adicionar mais *timeslots* ao cenário de acordo com o cálculo da sua rota, e agendamento do tempo de transmissão e hibernação dos sensores.
18. Quantidade de *timeslots* informado para o item 17 da Figura 15.
19. Adiciona um VANT, EC ou aumenta a velocidade do VANT.
20. Remove um VANT, EC ou diminui a velocidade do VANT.
21. Quantidade de VANTs no cenário.
22. Seleciona VANT para utilização dos itens 19/20 da Figura 15.
23. Quantidade de EC no cenário
24. Seleciona EC para utilização dos itens 19/20 da Figura 15.
25. Velocidade dos VANTs
26. Seleciona Velocidade para utilização dos itens 19/20 da Figura 15.
27. Habilita o modo *DEBUG*, onde apresenta informações internas aos objetos do sistema para análise.
28. Quantidade de simulações já realizadas.
29. Habilita um BIP ao terminar a simulação em tempo real.

Continuando a barra lateral do UAV Simulator, temos as demais funcionalidades apresentadas na Figura 16.

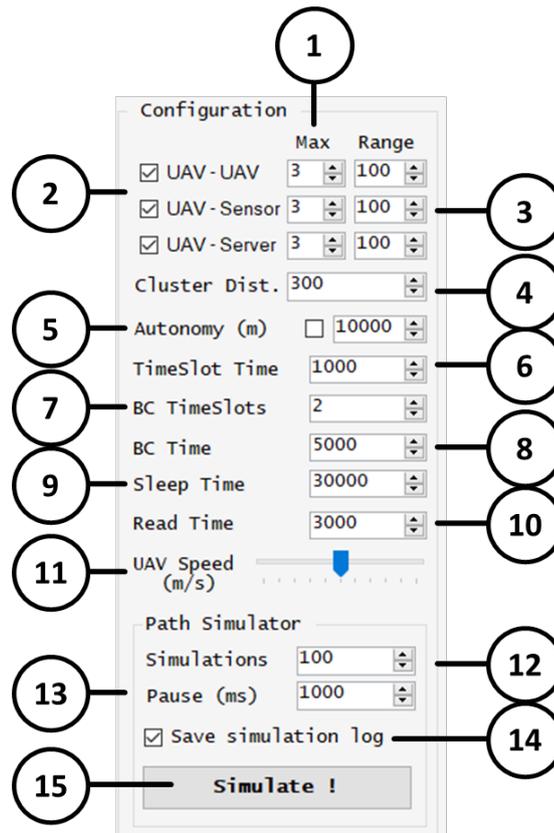


Figura 16 – Funcionalidades do *UAV Simulator*.

1. Quantidade máxima de conexões tipo VANT-VANT, VANT-EC e VANT-Depósito.
2. Habilita conexões tipo VANT-VANT, VANT-EC e VANT-Depósito.
3. Raio de comunicação VANT-VANT, VANT-EC e VANT-Depósito.
4. Distância utilizada como referência para criação dos agrupamentos dos *clusters* para ECs.
5. Autonomia do VANT em tempo e/ou distância informados pelo usuário.
6. Tempo do *timeslot*, informado pelo usuário, ou calculado pelo VANT.
7. Quantidade de timeslots utilizados como referência para *broadcast*.
8. Tempo de *broadcast* da EC.
9. Tempo de Hibernação da EC.
10. Tempo de leitura por parte do VANT sobre a EC.
11. Velocidade de deslocamento do VANT (5m/s – 80m/s)
12. Simulador automatizado – Quantidade de simulações automáticas

13. Pausa entre simulações
14. Salva os logs em arquivos extensão CSV (separado por vírgulas).
15. Inicia a simulação.

Continuando o painel inferior do *UAV Simulator*, temos as demais funcionalidades apresentadas na Figura 17.

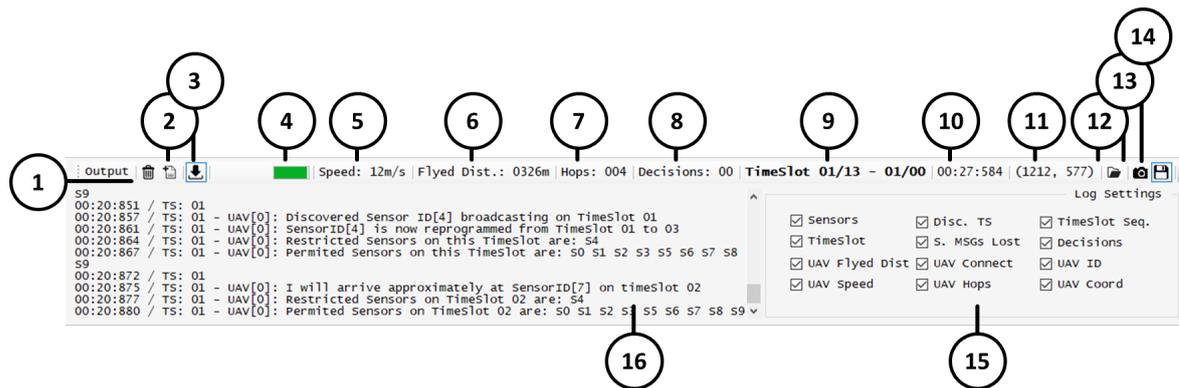


Figura 17 – Painel de saída do *UAV Simulator*.

1. Apaga o painel de visualização das saídas do UAV Simulator.
2. Copia para a área de transferência do Windows as informações de saída do UAV Simulator.
3. Habilita a visualização da última linha do painel de saída do UAV Simulator.
4. Visualização gráfica da autonomia/bateria do VANT titular que está em simulação.
5. Velocidade do VANT em m/s.
6. Distância percorrida pelo VANT em metros.
7. Quantidade de saltos realizados pelo VANT.
8. Quantidade de decisões tomadas pelo VANT.
9. Timeslot atual / Total de Timeslots - Timeslots sequencial / Quantidade de voltas (repetições de timeslots).
10. Tempo de simulação
11. Tamanho da área de simulação em metros
12. Abrir pasta de logs

13. Bate uma foto da área de simulação e salva na pasta de logs.
14. Habilita os logs de simulação.
15. Habilita as informações que serão salvas nos logs de simulação.
16. Saída de eventos do simulador.

Na Figura 18., são apresentadas as informações em tempo real referente às estações de coleta.

ID	Timer	TimeSlot(O/t)
S0	29174	14
S1	15383	11
S2	140680	07
S3	15463	13
S4	1611	12
S5	85173	05/03
S6	126946	06
S7	15728	09/11
S8	44141	00

Figura 18 – Painel de visualização das ECs.

1. ID da Estação de Coleta.
2. Temporizador de seu estado atual (Transmitindo / Hibernando). Caso o campo esteja destacado em verde, a EC está em modo ativo de transmissão. Em amarelo, seu estado está prestes a alterar.
3. *Timeslot* aleatório ao qual a EC foi atribuída automaticamente (O = *OLD*, N = *NEW*) e timeslot ao qual o VANT a programou. O campo destacado em azul significa que o VANT já a reprogramou. Em branco, a EC ainda não foi localizada pelo VANT.

Sobre a área de simulação do *UAV Simulator* apresentado pela Figura 19, são apresentadas as seguintes informações. Além das informações destacadas, o sistema permite funcionalidades de arrastar e soltar objetos para reposicioná-los de acordo com a escolha do usuário, facilitando a criação de cenários para testes.

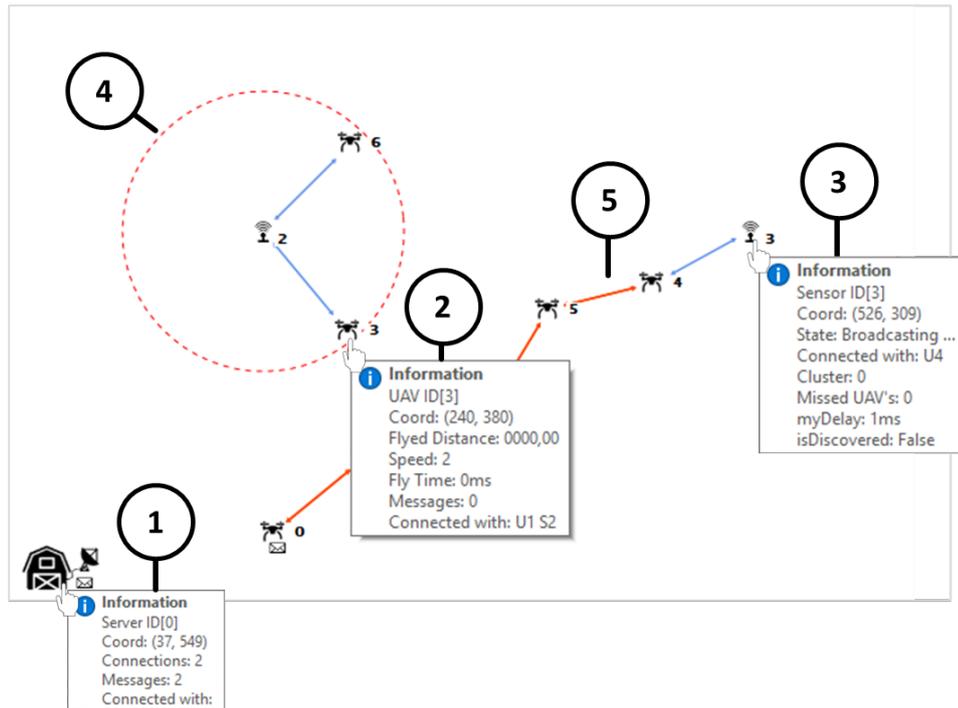


Figura 19 – Área de Simulação e Informações sobre Objetos.

1. Ao manter o mouse sobre o depósito (*mouse hover*), são apresentadas as informações internas referente ao depósito.
2. Ao manter o mouse sobre o VANT (*mouse hover*), são apresentadas informações adicionais internas referente ao objeto selecionado.
3. Ao manter o mouse sobre a EC (*mouse hover*), são apresentadas informações adicionais internas referente às estações de coleta.
4. Representação do raio de comunicação da EC quando habilitado para ser apresentado.
5. Representação da conexão VANT-VANT

# 5 OTIMIZAÇÃO DO PERCURSO PLANEJADO

Neste capítulo vamos apresentar os modelos de otimização de percurso estudados e implementados no UAV Simulador. O primeiro modelo apresentado será o de Kashuba et al. (KASHUBA V. I. NOVIKOV, 2015), o segundo será o *NearestPoint*, que trata-se do ponto mais próximo entre o VANT e o sensor de coleta, sendo que o VANT dirige-se apenas até a borda do raio de comunicação desta estação, e além disso, temos como referência um modelo Sem otimização, que é o percurso nominal do VANT, pairando diretamente sobre cada estação, ou seja, um modelo sem nenhum tipo de otimização de percurso.

## 5.1 Modelo de Kashuba

Através do desenvolvimento de um simulador próprio, seguimos a metodologia proposta por Kashuba et al. (KASHUBA V. I. NOVIKOV, 2015) para redução do percurso planejado para coleta de dados sensoriais utilizando VANTs – Veículos Aéreos Não Tripulados. Em seu trabalho, os autores propõem uma abordagem para a área de cobertura destes sensores utilizando o ângulo bissetriz formado entre o VANT, e seus dois próximos saltos (sensores), para realizar o cálculo do ponto de contato entre o VANT e a área de cobertura destes nós. Os autores levam em consideração que o VANT parte de um ponto inicial, aqui chamado de depósito, já conhecendo todas as coordenadas dos sensores ao qual ele deve percorrer.

Na Figura 20-a e Figura 20-b os autores apresentam um cenário onde o VANT se localiza no ponto A e deve se deslocar até o ponto C, passando antes pelo ponto B, traçando assim a rota A-B-C como o percurso não otimizado, sempre obrigando o VANT a sobrevoar diretamente a coordenada de onde o sensor encontra-se. Ao aplicar a solução proposta em (KASHUBA V. I. NOVIKOV, 2015), é realizado o cálculo do ângulo bissetriz de A, em relação aos pontos B e C, e utiliza-se este ângulo para gerar uma trajetória em linha reta até a intersecção (Ponto X) com o raio de comunicação do sensor B, gerando assim a trajetória A-X-C apresentada pelos autores como a rota otimizada para o VANT.

O resultado do modelo proposto em (KASHUBA V. I. NOVIKOV, 2015) é visualizado na Figura 20-c, onde o percurso não otimizado é dado pela trajetória onde o VANT é obrigado a sobrevoar cada sensor. Juntamente é apresentado o percurso otimizado, representado pelo trajeto destacado com círculos na mesma figura, identificando o ponto percorrido pelo VANT em cada salto, fazendo com que ele sobrevoe o limite de comunicação da área de abrangência de cada sensor, criando assim seu modelo otimização de percurso.

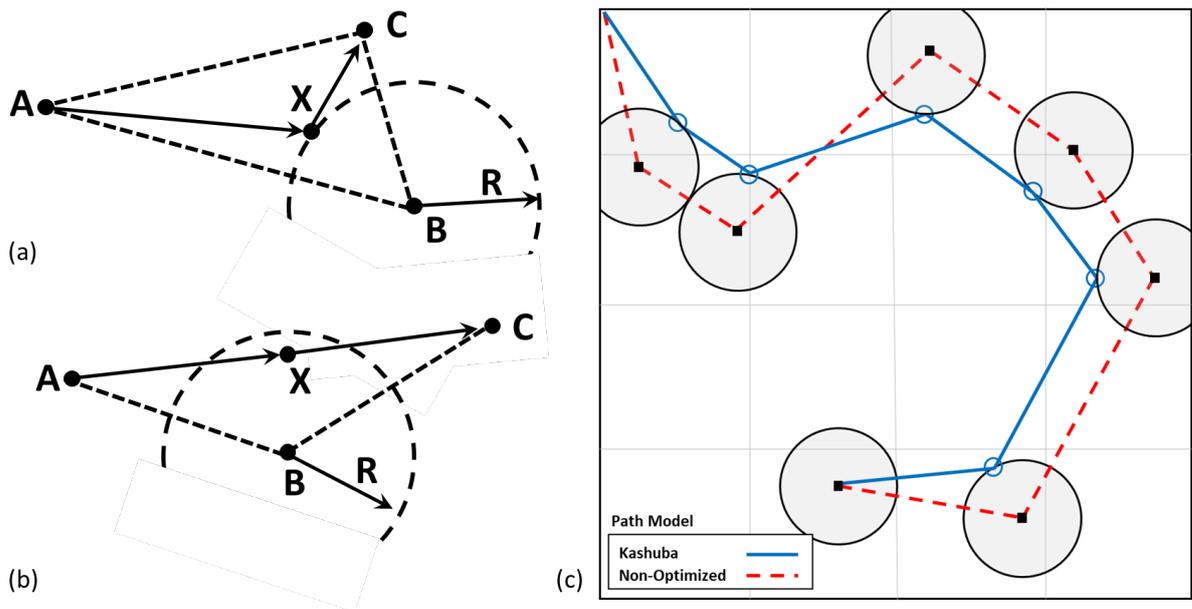


Figura 20 – Proposta de otimização de percurso dos autores.

Em seu artigo, os autores não discutem questões de confiabilidade na comunicação entre o VANT e o Sensor, uma vez que o VANT se encontra no limiar do raio de comunicação do seu destino.

A definição matemática que o ponto X deve obedecer é apresentada pela Equação 5.1 e 5.2, onde a distância dada pelos pontos B e X deverá ser menor que o raio de comunicação do sensor, além de a distância entre os pontos A-X, e X-C deverá ser sempre minimizada.

$$d(B, X) \leq R \tag{5.1}$$

$$d(A, X) + d(X, C) \rightarrow \min' \tag{5.2}$$

Utilizando um simulador próprio, os autores utilizaram 10 sensores variando seu raio de comunicação em 50, 100 e 200m. Cada valor obtido foi extraído à partir do resultado médio das simulações utilizando uma área de simulação de 2000m x 2000m. Os autores criaram diversos cenários aleatórios, variando as posições dos sensores dentro desta área, e calculam o percurso não otimizado, além do percurso seguindo as regras do modelo proposto por eles.

A Figura 21-a apresenta os resultados obtidos, comparando o percurso não otimizado com o modelo otimizado, em função do raio de comunicação dos sensores. Já a Figura 21-b apresenta o percentual de redução de percurso de seu modelo, também em função do raio de comunicação dos sensores.

Além disso, os autores apresentam na Figura 22 os resultados obtidos para o raio de comunicação de cada sensor fixados em 100 metros.

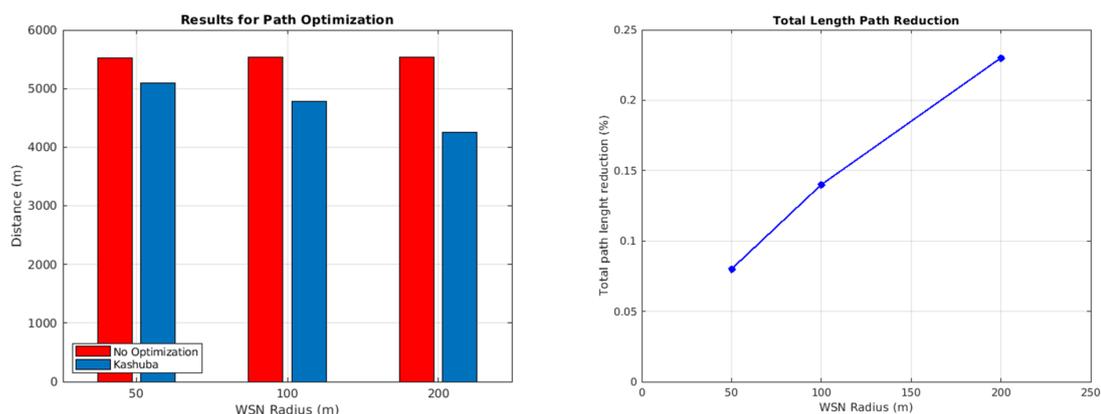


Figura 21 – Resultados para 10 sensores, em (KASHUBA V. I. NOVIKOV, 2015).

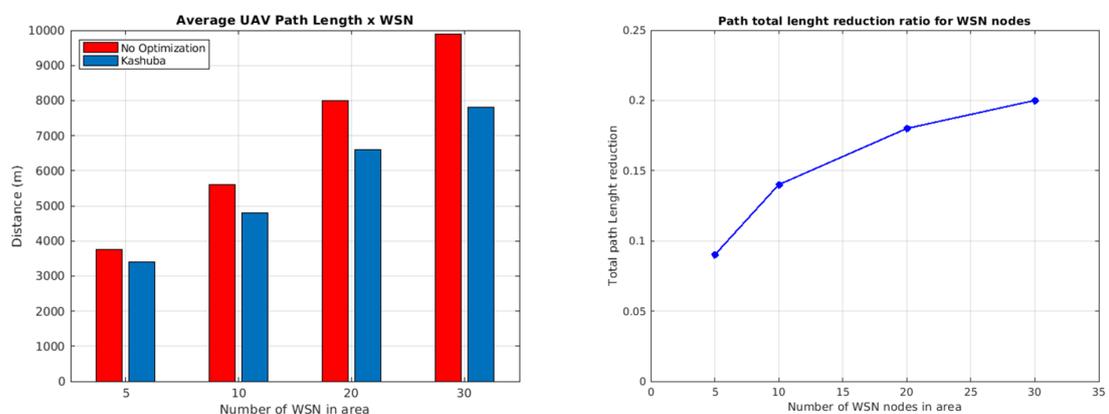


Figura 22 – Variação da Quantidade de Sensores em (KASHUBA V. I. NOVIKOV, 2015).

De acordo com os resultados apresentados na Figura 22-b, quanto maior a quantidade de sensores a serem percorridos pelo VANT, maior será a taxa de otimização de percurso de acordo com o modelo proposto pelos autores em seu artigo. Kashuba em seu artigo (KASHUBA V. I. NOVIKOV, 2015) alega que a taxa de otimização é eficiente, chegando a valores de até 20% de redução de percurso variando o raio de comunicação das estações de coleta, e quantidade de sensores como apresentado, além de possuir um baixo custo computacional para cálculo do percurso do VANT sobre sua rota otimizada.

## 5.2 Validação do Modelo

Utilizando o UAV Simulator (UAVSIMULATOR, 2019) foi possível implementar o modelo de otimização de percurso apresentado por Kashuba et al. (KASHUBA V. I. NOVIKOV, 2015). Na Figura 23 é apresentada a sobreposição gráfica do percurso gerado pelo modelo de Kashuba implantado como fundo de tela no ambiente de simulação do UAV Simulator. À esquerda da figura a imagem original do artigo apresentado, e à direita

a mesma imagem utilizada como fundo ao ambiente de simulação do UAV Simulator. Os sensores foram posicionados sobre cada elemento da figura original, e foi traçada a rota não otimizada (em vermelho), juntamente com o método de Kashuba aplicado sobre nosso simulador (em azul).

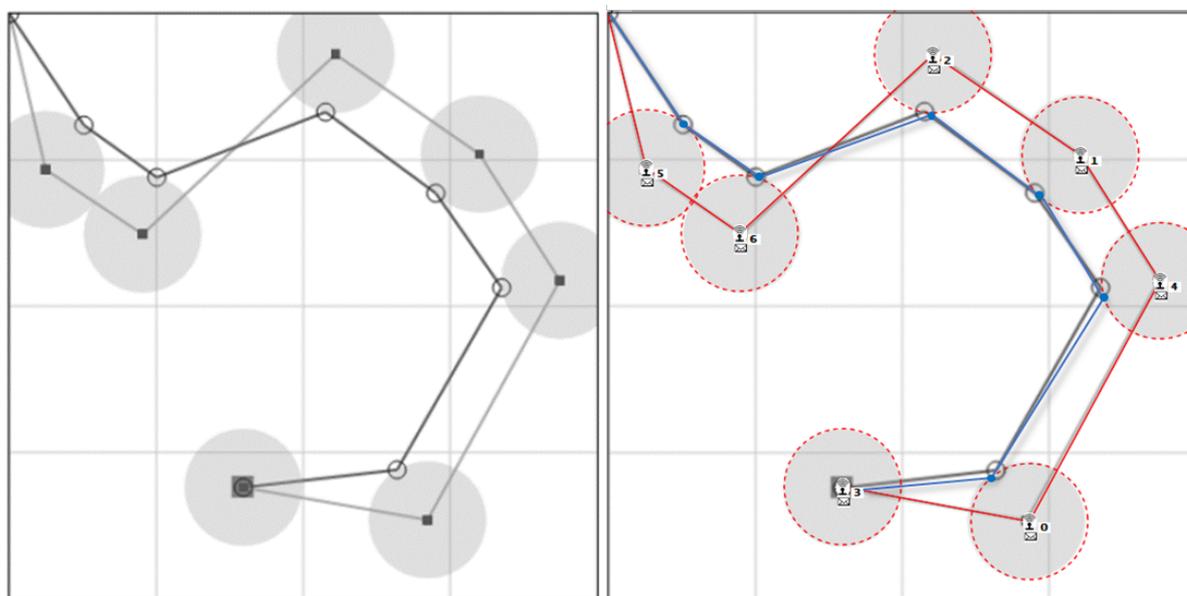


Figura 23 – Resultado do UAV Simulator sobreposto ao apresentado por Kashuba.

Apesar das dificuldades encontradas devido à falta de detalhes técnicos no artigo de Kashuba (KASHUBA V. I. NOVIKOV, 2015), como mencionamos anteriormente, os resultados (gráficos) obtidos em nosso simulador são muito próximos. Para cada resultado médio, realizamos 100 simulações e obtivemos a média aritmética do percurso total (não otimizado), além do percurso otimizado de acordo com o modelo proposto por Kashuba et al (KASHUBA V. I. NOVIKOV, 2015). Para os resultados que serão apresentados nas Figura 24 utilizamos área de simulação de 2000m x 2000m, 10 sensores, onde o raio de comunicação de cada sensor foi configurado para 50, 100 e 200 metros, exatamente como Kashuba realizou em seus experimentos.

Além disso, na Figura 25 obtivemos os valores para o raio dos sensores fixados em 100m, e variamos a quantidade de sensores dentro da área de simulação de 2000m x 2000m, dispostos de forma aleatória. Na Figura 25 são apresentados os resultados obtidos em (KASHUBA V. I. NOVIKOV, 2015), juntamente com os resultados obtidos em nosso simulador.

Através da análise das Figura 24 e 25, temos um indicativo de que a implementação do modelo proposto por Kashuba et al. (KASHUBA V. I. NOVIKOV, 2015) em nosso simulador está coerente com os resultados obtidos em seu artigo. As pequenas diferenças nos resultados gráficos se dão devido as adaptações já apresentadas que tivemos que realizar em nosso código, porém ainda assim, a divergência entre os resultados é mínima,

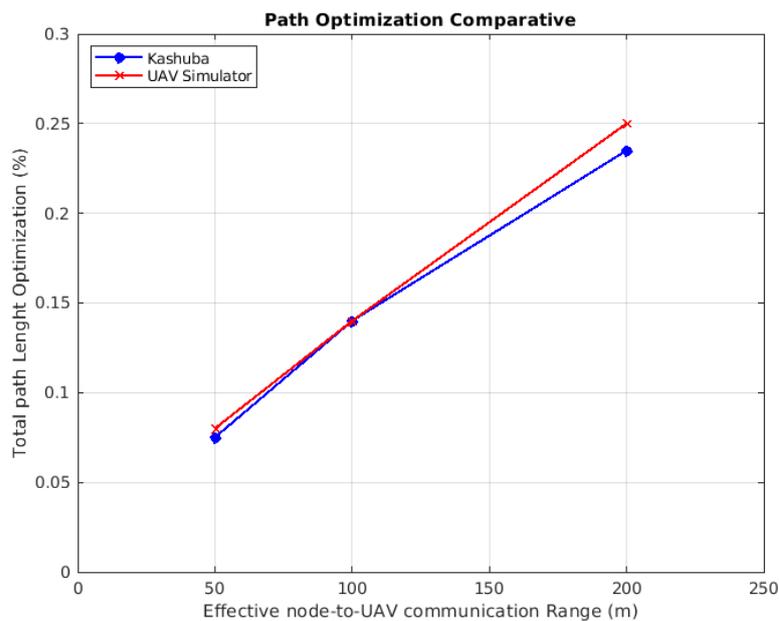


Figura 24 – Valores obtidos em (KASHUBA V. I. NOVIKOV, 2015), e em nosso simulador (UAVSIMULATOR, 2019).

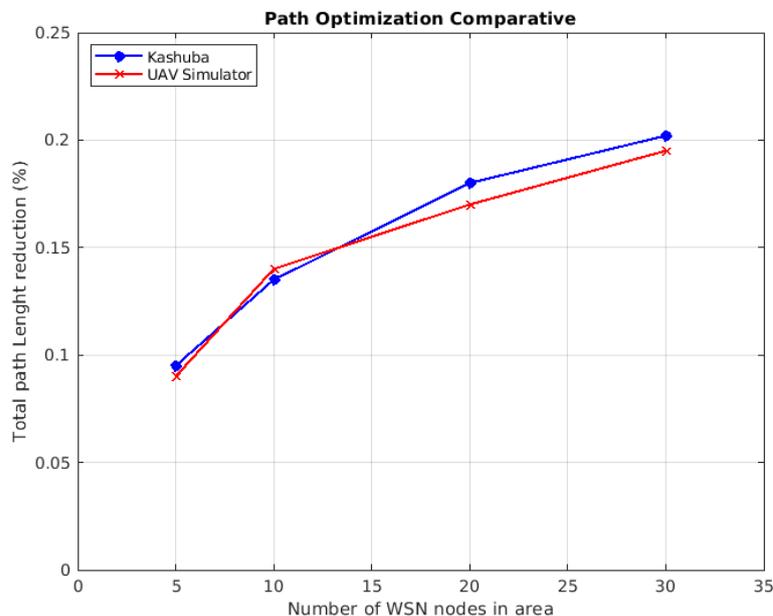


Figura 25 – Valores obtidos em (KASHUBA V. I. NOVIKOV, 2015), e em nosso simulador (UAVSIMULATOR, 2019).

validando assim o modelo de Kashuba em nosso simulador. Na Figura 26 é apresentado um Pseudocódigo referente a implementação do código dos autores em nosso simulador.

Como mencionado, o autor não detalha situações como apresentadas acima no código, onde a linha bisetritz formada entre o ângulo do VANT, juntamente com seus

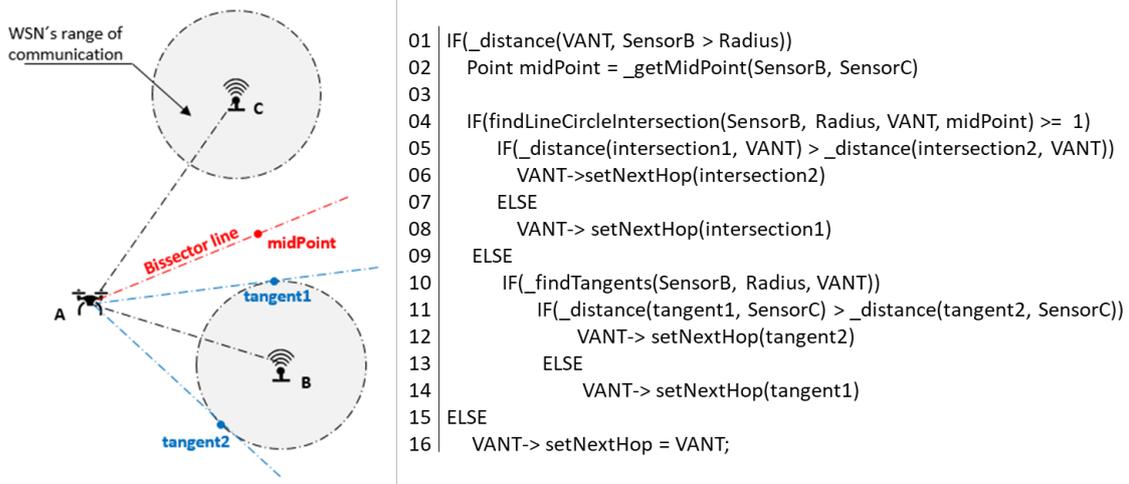


Figura 26 – Pseudocódigo referente ao percurso de Kashuba.

dois próximos sensores não cruza com nenhuma área de cobertura do seu próximo sensor, sendo assim necessária adaptação ao código para o VANT deslocar-se ao ponto tangente mais próximo do sensor subsequente (Sensor C), referente à Figura 26.

### 5.3 Modelo NearestPoint

Com a implementação do modelo de Kashuba validada com sucesso em nosso simulador, optamos por desenvolver um outro modelo de otimização de chegada à área de cobertura dos sensores em campo, denominado *NearestPoint*, também formalmente conhecido como DDNN – *The Direcional NN (Nearest Neighbour) Algorithm Directed to the Next Nearest Node* (ALEMAYEHU, 2017). O modelo é simples, pois realiza o cálculo da posição atual do VANT em linha reta até seu próximo sensor, sempre utilizando a intersecção entre esta linha reta e o raio de comunicação do sensor como seu ponto mais próximo, levando em consideração sempre o vizinho mais próximo em relação à sua posição atual.

Na Figura 27 temos a representação matemática da solução, dada por um triângulo retângulo onde o VANT que se encontra localizado no vértice A, calcula a distância euclidiana através da Equação 5.3, com o ponto B, sendo este o local onde o seu próximo salto está programado. O ponto mais próximo, é onde essa reta, formada pelos pontos A e B, é interseccionada pelo raio de comunicação do sensor no ponto B, sendo assim, este ponto é denominado NearestPoint  $N_P$ .

Matematicamente temos a seguinte situação:

$$d_{AB} = \sqrt{(B_X - A_X)^2 + (B_Y - A_Y)^2} \quad (5.3)$$

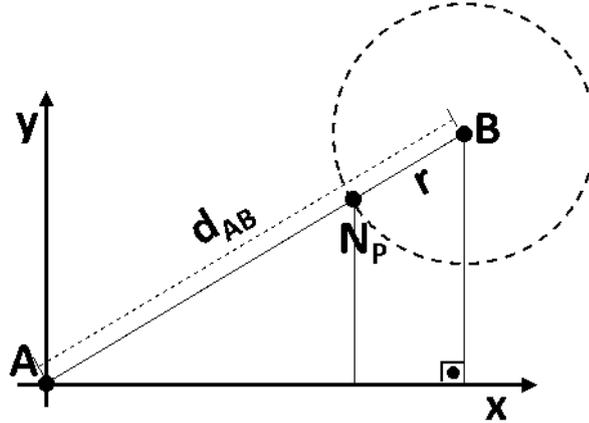


Figura 27 – Modelo matemático do *NearestPoint*.

Para o cálculo das coordenadas do *NearestPoint*  $N_P = (N_{P_X}, N_{P_Y})$ , através da Figura 27 obtemos matematicamente nosso ponto de interesse utilizando as equações 3 e 4 que serão utilizadas para calcular os pontos X e Y individualmente. Através de um simples cálculo trigonométrico, temos a equação 7 para cálculo da coordenada  $N_{P_X}$ .

$$\left[ \frac{(N_{P_X} - A_X)}{(d_{AB} - r)} = \frac{(B_X - A_X)}{d_{AB}} \rightarrow N_{P_X} = A_X + \frac{(B_X - A_X)(d_{AB} - r)}{d_{AB}} \right] \quad (5.4)$$

Prosseguindo com a dedução trigonométrica, apresentamos a equação 5.5 para o cálculo da coordenada  $N_{P_Y}$ .

$$\left[ \frac{(N_{P_Y} - A_Y)}{(d_{AB} - r)} = \frac{(B_Y - A_Y)}{d_{AB}} \rightarrow N_{P_Y} = A_Y + \frac{(B_Y - A_Y)(d_{AB} - r)}{d_{AB}} \right] \quad (5.5)$$

Sendo assim, a distância total  $D_T$  percorrida pelo VANT, desde sua origem até enésimo sensor S calculado no mapa será representado através da equação 5.6. A primeira parte da equação representa a distância *NearestPoint* da posição inicial do VANT, em relação ao seu primeiro destino  $S_0$ , e em seguida é aplicado o cálculo da distância *NearestPoint* para todas os sensores envolvidos na rota do VANT, se repetindo a cada salto.

$$D_T(N) = N_P(UAV_1, S_0) + \sum_1^{n-1} N_P(S_{i-1}, S_i) \quad (5.6)$$

Aplicamos este modelo matemático em nosso simulador para realizar as devidas comparações com o modelo sem otimização de percurso, e com o modelo apresentado por Kashuba em seu artigo. Em nosso ambiente, temos sensores distribuídos aleatoriamente por uma determinada área onde o VANT irá calcular a trajetória em linha reta (menor distância entre dois pontos), de sua posição atual até o ponto mais próximo da intersecção desta linha reta até seu destino, criando assim o seguinte modelo apresentado pela Figura 28.

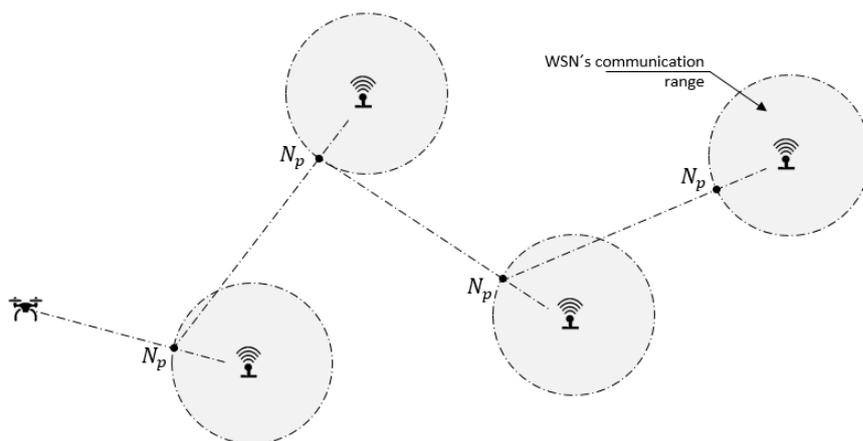


Figura 28 – Exemplo de rota gerada pelo *NearestPoint*.

Na Figura 29 temos uma imagem do UAV Simulator ilustrando o trajeto *Nearest-Point*, destacado pelo trajeto com quadrados representando cada salto calculado como *NearestPoint*, e para comparação, o trajeto não otimizado, que visa fazer com que o VANT sobrevoe por cima de cada sensor.

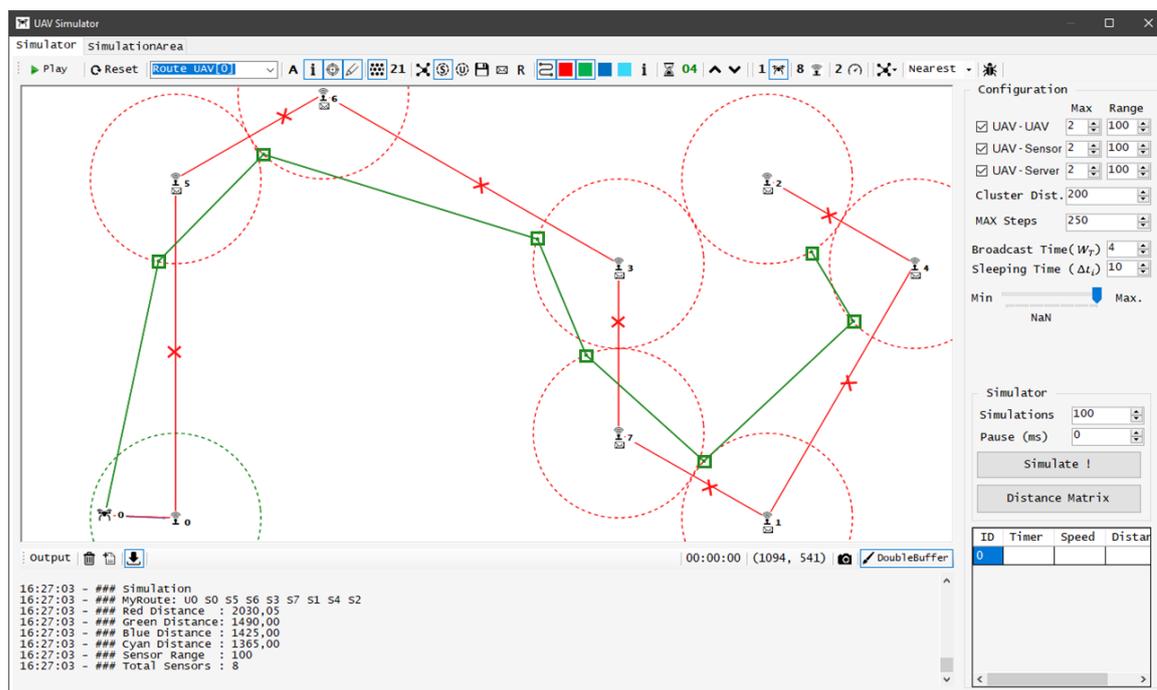


Figura 29 – *UAV Simulator* (UAVSIMULATOR, 2019) com o modelo *NearestPoint* aplicado.

Além disso, como podemos ver na Figura 29, o VANT localizado em sua posição inicial já se encontra dentro do raio de cobertura de um sensor, utilizando suas próprias coordenadas como o *NearestPoint*.

## 5.4 Resultados Comparativos

Através da implementação dos modelos *NearestPoint*, e de Kashuba, em nosso simulador foram realizados os testes seguindo a metodologia apresentada para comparação entre os modelos de percurso mencionados neste documento. Na Figura 30 podemos ver a implementação de gráfica de ambos modelos de otimização de percurso, juntamente com modelo sem otimização.

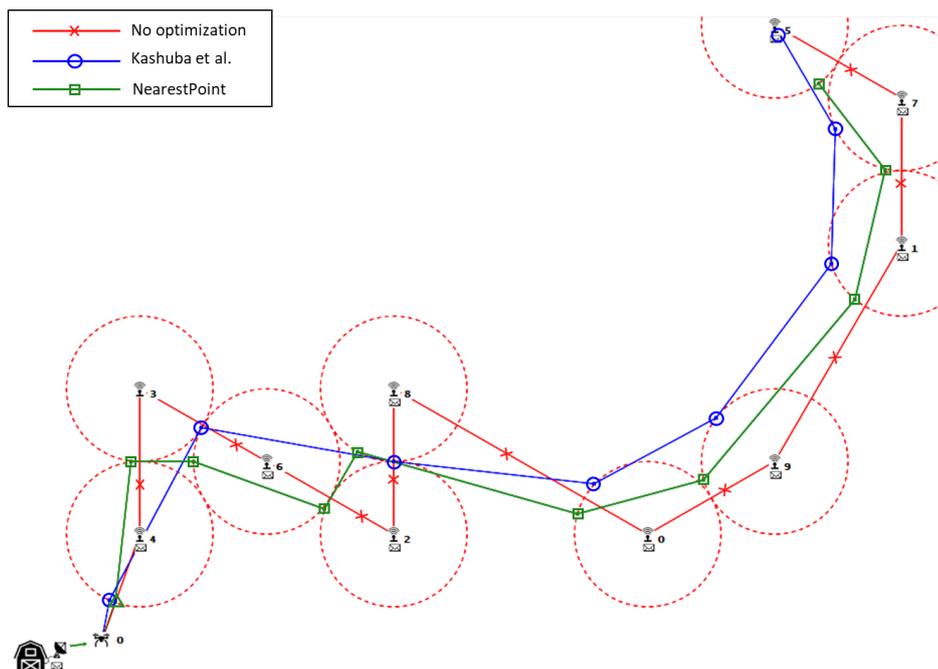


Figura 30 – Simulador apresentado os três modelos de percurso.

Ao analisarmos a Figura 31-b, que compara os modelos de Kashuba com o *NearestPoint*, comparando o percentual de otimização de percurso em função da quantidade de sensores inseridos aleatoriamente dentro da área de simulação, percebemos que existe um aprimoramento em relação aos resultados comparados com o *NearestPoint*. Essa melhora fica mais evidente quando trabalhamos com baixa densidade de sensores. Além disso, trabalhamos com algumas variações de cenários onde os aprimoramentos são mais significativos. Como apresentado na Figura 31-a/b, ao trabalharmos com uma baixa densidade de sensores na área de simulação, variando o raio de comunicação dos mesmos em 50, 100 e 200 metros, de um total de 100 simulações para cada cenário, obtemos um ganho médio de aproximadamente 10% de redução de percurso do modelo *NearestPoint* em relação ao modelo de Kashuba, considerando o cenário disposto com cinco sensores e raio de comunicação de 200 metros.

Podemos observar através da Figura 31-a que ao aumentar a área de comunicação da rede de sensores, o modelo *NearestPoint* apresenta maior vantagem em relação a otimização de percurso para o VANT, do que o modelo apresentado por Kashuba. Além

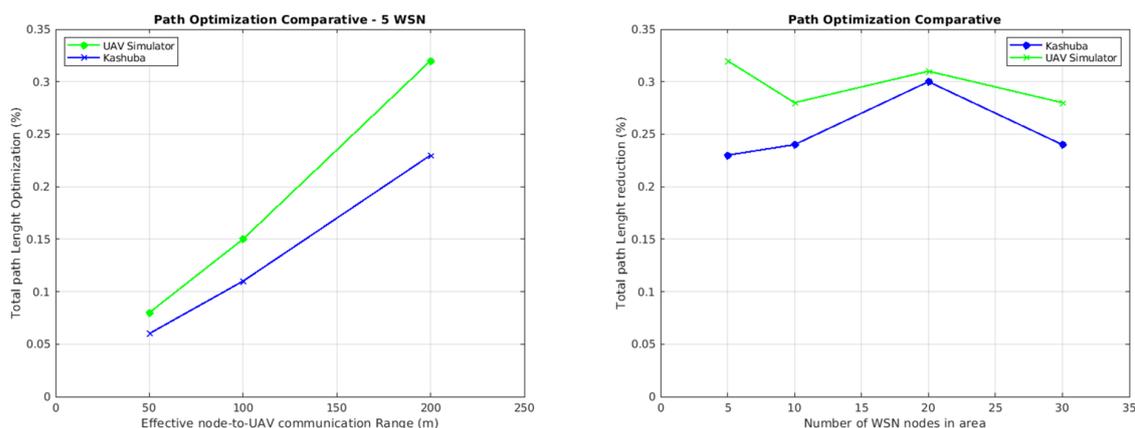


Figura 31 – Resultados comparativos.

disso, na Figura 31-b é perceptível que na medida em que a densidade de sensores aumenta, a diferença do percentual de redução de percurso diminui. Além disso, a otimização de percurso média dada pelo *NearestPoint* apresenta melhor desempenho em relação ao modelo de Kashuba, de acordo com a Figura 34, nossos resultados foram superiores em 95% dos cenários.

Através da Figura 32, é apresentado a diferença dos valores percentuais entre *NearestPoint* e Kashuba para o percentual de redução efetiva de percurso, em função da densidade de sensores na área em questão.

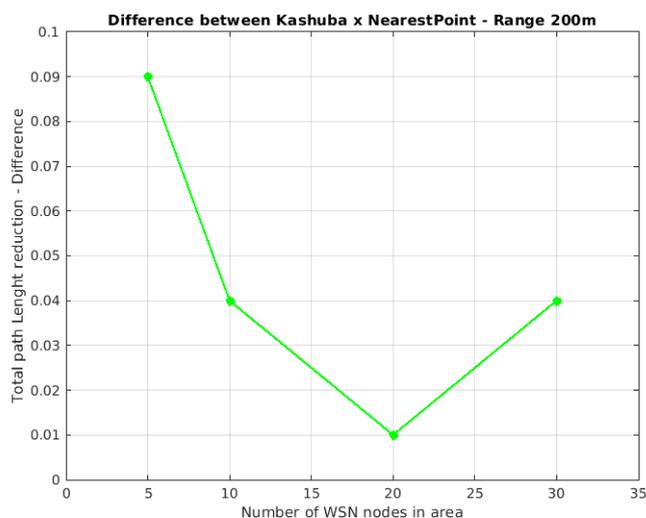


Figura 32 – Diferença entre os modelos *NearestPoint* e Kashuba.

O modelo de Kashuba, apresentou bons resultados em relação ao percurso não otimizado, como já vimos, e o modelo *NearestPoint* apresenta melhores resultados em relação aos de Kashuba, porém tais resultados são médios. Na Figura 33 é apresentado o gráfico de intervalo de valores, máximos, mínimos e médios comparando os três modelos discutidos.

Para a geração de tais valores, utilizamos 5 sensores espalhados aleatoriamente pela área de simulação com raio de comunicação dos sensores fixados em 200m, não consideramos sobreposição entre eles, sendo assim, realizamos um total de 100 simulações, armazenando a distância total de percurso em metros de cada modelo, e também armazenamos qual algoritmo/modelo apresentou melhor resultado em relação a otimização de percurso, sendo esse resultado favorável contabilizado como uma vitória ou derrota em caso de menor otimização de percurso em comparação com o outro algoritmo. Como podemos observar, o modelo *NearestPoint* apresenta resultados médios significativamente melhores em relação aos demais modelos.

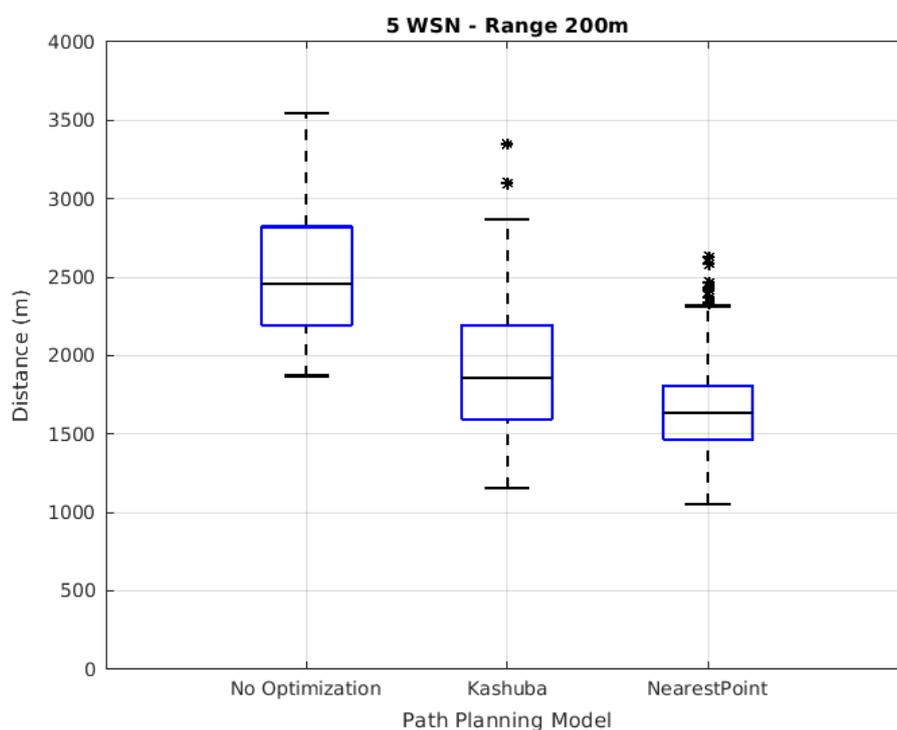


Figura 33 – Variação dos resultados obtidos em cada modelo para o percurso definido para o VANT.

Na Figura 34 temos o resultado gráfico do total de vitórias para cada modelo de otimização de percurso. No gráfico da base, temos o total de vitórias do *NearestPoint*, e sobre este o total de vitórias do modelo de Kashuba para um raio de comunicação dos sensores fixados em 200 metros. Como podemos ver, mesmo com um percentual baixo, o índice de vitórias do modelo de Kashuba aumenta conforme a densidade de sensores na área é elevado.

Através de uma análise realizada nas Figura 31 à 35, que conforme a densidade de sensores na área em questão aumenta, o modelo de Kashuba se torna mais eficiente em relação ao *NearestPoint*.

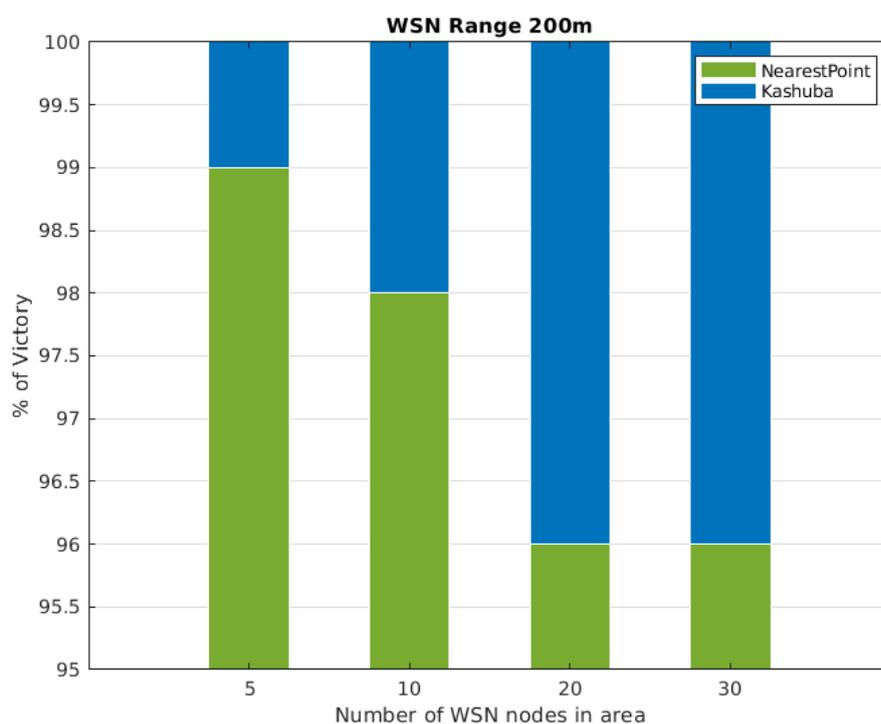


Figura 34 – Comparativo de vitórias entre *NearestPoint* e Kashuba.

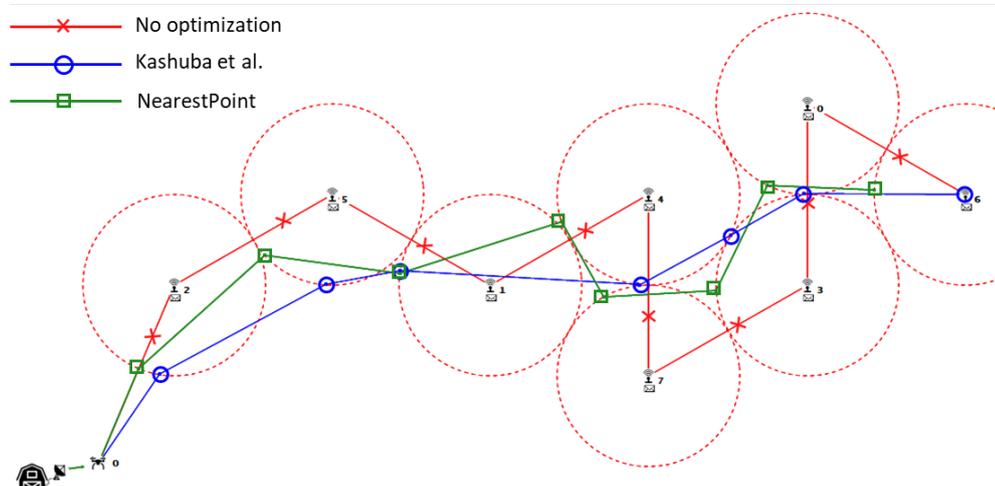


Figura 35 – Cenário com alta densidade de sensores.

Em ambientes mais densos em termos de sensores, podemos observar através da Figura 35 que o *NearestPoint* apresenta um comportamento em ziguezague, encaminhando o VANT de um sensor à outro, fazendo com que o VANT percorra um percurso maior. Já o modelo de Kashuba, pelo fato de trabalhar com o ângulo bissetriz entre seus próximos sensores apresenta na média um comportamento retilíneo ao criar o trajeto entre as áreas de coberturas dos sensores, sendo assim, uma grande vantagem em ambientes mais populosos em termos de sensores e área de cobertura.

Já ao analisar o cenário oposto, em ambientes menos densos em termos de sensores e área de cobertura, o modelo *NearestPoint* apresenta uma vantagem pois ele se torna mais eficiente indo direto ao encontro do seu próximo sensor, parando no momento exato em que entra na sua área de cobertura, partindo já para seu próximo destino. Como os sensores estão mais espalhados através da área, a trajetória torna-se mais retilínea de um ponto ao outro, independentemente do modo de aproximação da área de cobertura do sensor, portanto a busca pelo ponto mais próximo se torna vantajosa nesse cenário.

Na Figura 36, ao analisarmos o modelo *NearestPoint*, o VANT traça uma linha reta de sua posição atual para o sensor, utilizando o ponto em linha que intersecciona o raio de comunicação do sensor como seu destino, e no modelo de Kashuba, o primeiro ponto calculado seria o ângulo formado pela bissetriz entre os sensores 2 e 3, porém este ângulo não existe, sendo necessário o cálculo do ponto tangente, causando assim o custo maior do deslocamento do VANT em relação ao seu próximo destino. O processo se repete de sensor em sensor, sempre viajando pelo ponto tangente, portanto, cenários menos densos o modelo de Kashuba apresenta tal desvantagem.

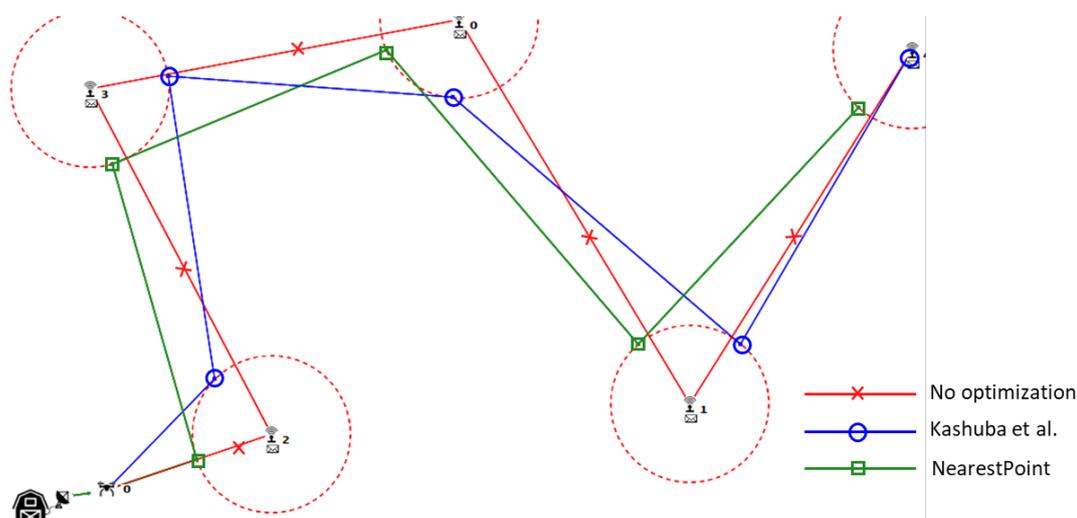


Figura 36 – Cenário com baixa densidade de sensores.

## 5.5 Discussão dos Resultados

No artigo (KASHUBA V. I. NOVIKOV, 2015) os autores explicam a concepção do uso do ângulo bissetriz para o cálculo da trajetória do VANT até o próximo sensor, gerando assim a otimização já apresentada, porém não explicam qual o método utilizado no caso de não haver um ângulo cujo a bissetriz entre seus dois próximos saltos não interseccione o raio de cobertura de seu próximo sensor. Ao analisarmos a Figura 37 temos o VANT localizado no ponto A à uma certa distância de seus próximos saltos, pontos B e C. A bissetriz calculada à partir do ponto A com os pontos B e C não gera uma

intersecção válida sobre o raio de comunicação do sensor localizado no ponto B, portanto o modelo apresentado pelos autores não pode ser aplicado para todos os cenários possíveis. Foi, portanto, necessário adotar novos métodos para complementar o modelo proposto pelos autores no caso de não houver intersecção da bissetriz com a área de cobertura do próximo sensor.

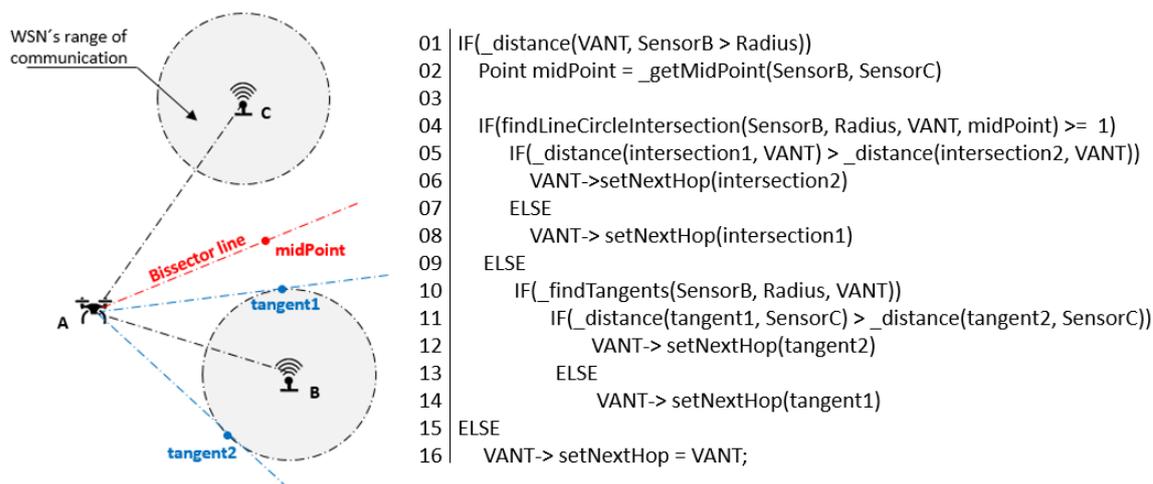


Figura 37 – Algoritmo criado para representar o trabalho de Kashuba.

O problema teve de ser contornado com a seguinte solução, primeiramente é verificado se o VANT se encontra dentro da área de cobertura de algum sensor no cenário inicial, para isso, a linha 01 do código apresentado realiza essa verificação, analisando se o VANT se encontra dentro do raio de comunicação do Sensor B, sendo seu primeiro destino dada a proximidade. Caso afirmativo, seu primeiro salto torna-se as próprias coordenadas do VANT, caso contrário é calculado um ponto médio entre os sensores B e C (chamado de “*midPoint*”) para localização do ângulo bissetriz entre os três pontos. Em seguida, é verificada se a reta gerada entre o VANT e o *midPoint* intersecciona o raio de comunicação do seu próximo salto, no caso SensorB. Essa reta, denominada acima na figura de “*Bisector Line*” pode interseccionar o raio de comunicação do Sensor B em até 2 pontos (*intersection1* e *intersection2*), caso seja este o cenário, o ponto mais próximo do VANT é escolhido como próximo salto para o VANT. Caso a reta bissetriz não interseccione nenhum ponto do raio de SensorB, conforme ilustrado na Figura 37, são calculados os dois pontos tangentes entre o VANT, e a área de comunicação do SensorB. Finalmente é escolhida a distância menor entre *tangent1* e *tangent2* do Sensor C como atribuição ao próximo salto do VANT. O processo se repete à cada salto.

Outro ponto que os autores não abordam é a questão do último salto. Como não existem mais pontos para serem calculadas a bissetriz do penúltimo para o último salto, o autor através da ilustração da Figura 20 simplesmente calcula o ponto final da trajetória do VANT como o ponto do último sensor, e não do raio de comunicação do sensor, fazendo

assim com que o VANT se desloque diretamente para as coordenadas do último sensor, perdendo assim qualquer possibilidade de otimização no último percurso.

Além disso, os autores limitam os resultados gerados no artigo para apenas 10 sensores em função do raio de comunicação, sendo estes 50, 100 e 200 metros conforme apresentado na Figura 21. Além disso, na Figura 22 os autores limitam seus resultados para 5, 10, 20 e 30 sensores com o raio de comunicação fixado apenas em 100m. Demais resultados não são apresentados, tais como, 5 sensores com raio de comunicação de 200 metros, ou 30 sensores com raio de comunicação de 50 metros, e assim por diante.

Outro ponto não abordado pelos autores é a questão da sobreposição das áreas de cobertura dos sensores. Levando em consideração a Figura 38, onde a área de simulação é um quadrado com arestas de 2000m, ou sendo este preenchido com 30 sensores, sendo que, cada sensor possui raio de cobertura de 200 metros. Ao utilizarmos uma disposição quadrada (WOLFRAMMATHWORLD, 2019) entre as células, não teremos área suficiente no simulador para acomodar integralmente todos os sensores, como podemos ver na Figura 38-a. Já na Figura 38-b temos o método de distribuição hexagonal (WOLFRAMMATHWORLD, 2019) que gera um melhor aproveitamento entre as áreas de cobertura das antenas, reduzindo assim áreas de sombra também, porém, o problema desse modelo, acomodando todas os sensores mesmo que não integralmente dentro da área de simulação, é a quantidade limitada de cenários possíveis para a geração das rotas entre os sensores, ou seja, a variação de cenários será pequena, sendo que todos apresentarão rotas semelhantes. Ao realizarmos o cálculo da razão da área total do simulador, pela somatória das áreas de cobertura dos trinta sensores, teremos aproximadamente 94% da área total da área de simulação coberta.

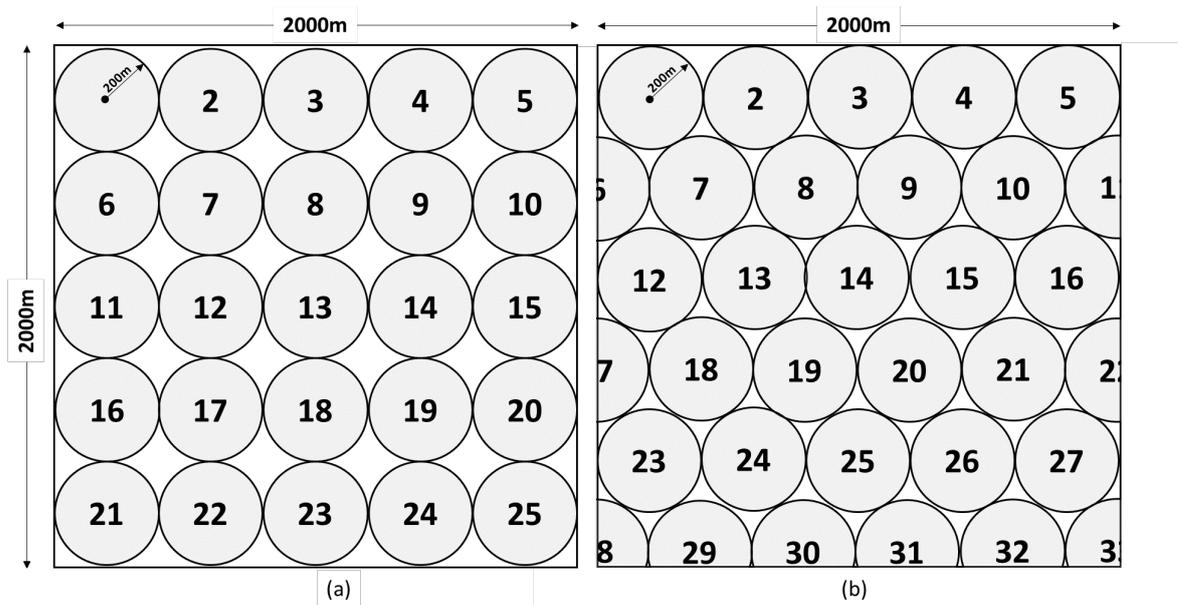


Figura 38 – (a) Disposição quadrada e hexagonal (b) de sensores no simulador.

Através da análise do cenário geral, podemos concluir que o modelo proposto com Kashuba foi implementado e validado no *UAV Simulator*. A pequena diferença em termos percentuais entre os resultados, se dá aos ajustes necessários para validar o modelo e falta de detalhamento técnico apresentado no artigo em (MAXA MOHAMED SLIM BEM MAHMOUD, 2017).

O modelo *NearestPoint* apresenta melhorias em relação aos resultados obtidos em nossas simulações chegando à 10% de diferença de ganho de otimização em relação aos valores obtidos por Kashuba, e se comparados diretamente ao cenário sem otimização, estes valores chegam a 30% de ganho de otimização.

Conforme podemos perceber através das Figura 31 a 36, a densidade de sensores está ligada ao resultado, favorecendo um modelo ou outro. Conforme a densidade de sensores diminui, o modelo *NearestPoint* apresenta melhores resultados que o modelo de Kashuba. Já no cenário oposto, com maior densidade de sensores na área de simulação o modelo proposto por Kashuba apresenta melhor performance, pois a rota calculada através do ângulo bissetriz formado entre o VANT e seus próximos saltos gera um percurso mais retilíneo em relação ao *NearestPoint*.

Sendo assim, optamos por utilizar o modelo *NearestPoint* para as simulações devido ao seu maior desempenho em relação aos demais modelos estudados e implementados.

## 6 PROGRAMAÇÃO DOS SENSORES *TIMESLOTS*

Através do desenvolvimento do próprio simulador, com o objetivo de que o VANT realize a programação das Estações de Coleta – ECs, que são sensores posicionados em uma determinada área, que se alternam entre estados de transmissão/recepção e hibernação de acordo com o percurso do VANT (YUE, 2018) (ZHAN Y. ZENG, 2018) (XIONG, 2018b) (ZENG, 2017). De acordo com os autores em [82], estudos na área de programação do agendamento do período de transmissão e hibernação dos sensores em função do percurso programado do VANT são novos e desafiadores.

O VANT utiliza como referência uma unidade de tempo chamada *timeslot*  $TS$  para organização e gerenciamento do tempo dos estados de hibernação e comunicação de cada EC. Para o cálculo do tempo de cada *imeslot*  $T_{TS}$ , o VANT utiliza a Equação 6.1, que leva em consideração o tempo total de seu percurso por entre todas as ECs, calculado a partir da sua velocidade média  $v_m$ , distância euclidiana entre todos os sensores ( $d_v(e_i, e_{(i+1)})$ ) da sua rota além do seu tempo de leitura  $t_{Li}$  sobre cada EC, e divide esse tempo total por uma determinada quantidade de *timeslots* para aquele cenário. A quantidade de *timeslots*  $Q_{TS}$  é informada através do simulador. De acordo com Xiong et al. (XIONG, 2018b), e Zheng et al. (ZENG, 2017) e Polo et al. (POLO G. HORNERO, 2015) em suas simulações, foi adotado uma velocidade média constante para os VANTs variando entre 5 e 25m/s, a qual também utilizaremos em nossas simulações. Além disso, ainda de acordo com Polo et al. (POLO G. HORNERO, 2015), um VANT convencional possui autonomia de tempo de voo de até aproximadamente 10 minutos, o que resulta, nessa velocidade média em uma autonomia aproximada de 10 km de percurso.

$$T_{TS} = \left( \sum_{i=1}^{n-1} ((d_v(e_i, e_{i+1})/v_m) + t_{Li}) \right) / Q_{TS} \quad (6.1)$$

Ao iniciarmos a simulação, cada EC é programada para entrar em modo de transmissão em um *timeslot*(tempo) aleatório. Nos demais *timeslots*, estas ECs encontram-se em modo de hibernação. Esse *timeslot* em que a EC se encontra em modo ativo de transmissão, pode ser compartilhado ou individual. Portanto, cabe ao VANT, durante seu percurso encontrar as ECs em modo ativo de transmissão e programar seu período de hibernação de acordo com sua própria rota, previamente definida, fazendo com que o resultado dessas reprogramações seja que as ECs entrem em modo de transmissão quando o VANT estiver sobre elas.

O problema que temos que resolver, além do planejamento de percurso do VANT

como já discutido no primeiro capítulo deste trabalho, é essa organização do timeslot ao qual cada EC se encontra quando o VANT inicia seu percurso. Não há como identificar em qual timeslot cada EC se encontra de forma prévia já que elas estão em locais remotos, e tecnicamente isoladas, sem comunicação alguma. O VANT, por sua vez, é utilizado como meio de transporte entre tais ECs, e o destino das informações geradas por elas. Portanto, cabe ao VANT sobrevoar cada EC, identificar o momento (*timeslot*) em que cada uma está em modo de transmissão, e reprogramá-la de acordo com o seu percurso planejado (JO J. HEO, 2018). Ao iniciar a simulação, cada EC recebe um timeslot de referência ao qual deverá alternar seu estado de hibernação para estado de comunicação ativada.

O tempo de duração de um *timeslot* é definido para cada cenário, pois varia de acordo com a disposição das ECs pelo cenário, impactando diretamente na distância total a ser percorrida pelo VANT, além da própria velocidade média aplicada ao VANT. Ao iniciarmos a simulação, cada EC é programada com essa unidade de medida (*timeslot*), e programada para que acorde em um determinado *timeslot* durante a simulação. A quantidade de timeslots que a EC irá permanecer ativada é definida como critério de simulação através da interface do sistema.

Na Figura 39, é possível ver o resultado final, idealizado em que o sistema se encontrará após a programação de todas as ECs. O VANT que se encontra em sua posição inicial, deverá percorrer sua rota pré-definida entre os quatro sensores na representação de um cenário abaixo, respeitando o intervalo de tempo necessário entre cada EC, fazendo com que cada uma entre em modo de transmissão no seu determinado *timeslot*.

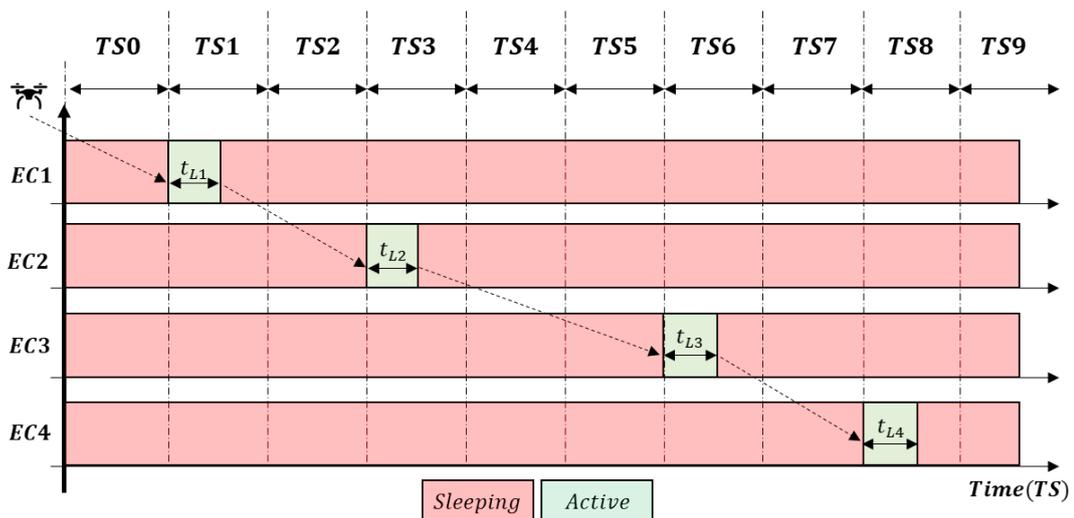


Figura 39 – Representação do sistema programado de acordo com o percurso do VANT.

Perceba na Figura 39 que cada EC possui seu próprio período de transmissão em seu respectivo *timeslot* - TS para transmissão e recepção (em verde), portanto, o VANT deverá se deslocar até a EC1 e o tempo para esse deslocamento será de um *timeslot*. Sendo assim, a EC1 deverá entrar em modo de transmissão no TS1. Em seguida, após a conexão

realizada com sucesso e leitura dos dados da EC, o VANT programa a mesma para que entre em modo de hibernação por um determinado tempo em *timeslots*, para que possa se dirigir ao seu próximo destino. Em seguida, o VANT terá pouco mais que um *timeslot* de tempo para se deslocar da EC1 à EC2 que estará em modo ativo de transmissão no TS3. Realizada a leitura dos dados, o VANT parte para seu próximo destino, a EC3 que se encontra um pouco mais distante da EC2, conseqüentemente o VANT necessitará de dois *timeslots* para se deslocar ao seu destino, sendo assim a EC3 deverá estar programada para acordar e transmitir no TS6. O processo se assim se repete entre EC3 e EC4 que permitem um *timeslot* de deslocamento para o VANT, sendo que o VANT encontrará EC4 transmitindo no TS8.

É importante ressaltar que o tempo de leitura  $t_{Li}$  é ajustável para as estações de coleta, e que após este tempo, o VANT coloca a estação para hibernar novamente, de forma que ela acorde e transmita no mesmo *timeslot* já programado.

O cálculo realizado pelo sistema, para sincronizar os eventos em *timeslots* se dá através do modelo apresentado na Figura 40, evitando assim que o VANT programado seu início de comunicação no *timeslot* 9, por exemplo, inicie exatamente no timeslot configurado, e sem variações (positiva ou negativa). Além disso, a finalização do evento de *broadcast* por parte da EC no exato momento em que o *timeslot* termina deve ser levado em consideração.

Inicialmente é calculada a diferença entre os *timeslots* atual e futuro a ser programado, resultando na diferença em *timeslots*, ao qual chamamos de *timeSlotDifference*. Somado a isso, é necessário levar em consideração o tempo já transmitido da EC, denominado de BCT (*BroadCast Time*), o tempo restante de *broadcast* (BCLT – *Broadcast LeftTime*), e finalmente o tempo de leitura  $T_L$  do VANT, para que seja possível realizar o cálculo exato do período em que a EC deverá hibernar.

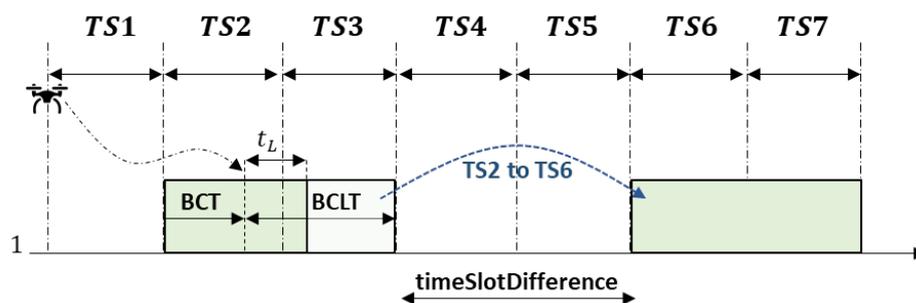


Figura 40 – Processo de reprogramação de *Timeslots*.

O algoritmo apresentado na Figura 41, realiza o cálculo da diferença entre o *timeslot* atual, e o novo ao qual a EC foi programada. É importante ressaltar que, o algoritmo calcula a diferença de tempo em *timeslots*, e em relação ao tempo de *broadcast* ativo da

EC, ou seja, o intervalo fracionado de um *timeslot* também é levado em consideração para o perfeito alinhamento dos estados de transmissão da EC, com o período de *timeslots* no relógio do VANT.

```

1 | timeSlotDifference = ((newTS - oldTS - BCLT) * timeSlotTime;
2 |   IF(BCLT > TL)
3 |     sensorSleepTime += (BCLT - TL)
4 |   ELSE
5 |     TL = BCLT
6 |
7 | WSNTimer = TL

```

Figura 41 – Pseudo algoritmo que descreve o cálculo do *Timeslot*.

No caso apresentado na Figura 40, o VANT chega à EC pouco mais da metade do TS2, ao qual ela entrou em modo de transmissão. Porém, após o tempo de leitura necessário, o VANT deverá calcular o BCLT (tempo restante de *broadcast*) e somar este tempo com a diferença de tempo restante de *broadcast* (em *timeslots*) para que a EC entre em hibernação no TS3, mas acorde exatamente ao iniciar o *timeslot* 6.

Além disso, para que o VANT se desloque entre as ECs de forma eficiente e minimize as chances de se deslocar para uma estação que esteja em modo de hibernação naquele determinado *timeslot* já conhecido, em que o VANT chegará até ela, foi desenvolvido um sistema de listas de proibição e permissão de ECs para cada *timeslot* que o VANT vai alimentando conforme passa pelas estações e as mesmas estão desligadas/hibernando, e também ao descobrir sensores em modo de transmissão, conforme representado pela Figura 42. A imagem citada apresenta que no *Timeslot* 0 (TS0) o VANT já passou pelas EC S3 e S4, e os mesmos estavam hibernando durante este *timeslot*, e ele desconsidera estes possíveis sensores numa próxima tomada de decisão sobre o *timeslot* 0 em questão, já sabe que as ECs S3 e S4 não estão ativas naquele momento, e toma a decisão de deslocar para outra EC de sua rota, de forma que não cometa mais os mesmos erros. Ainda na mesma imagem, podemos ver que o *timeslot* 1 (TS1) encontra-se eliminado, isso se dá devido ao VANT já ter localizado uma EC transmitindo sobre este *timeslot*, portanto, automaticamente desconsidera todas as ECs que possam estar vinculadas àquele *timeslot*.

Na Figura 17, é apresentada a saída de informações do simulador desenvolvido (UAVSIMULATOR, 2019), onde podemos acompanhar os eventos em tempo real, além dos logs dos objetos internos dos VANTs, ECs e outras informações do sistema, de forma tal que podemos acompanhar como o sistema toma suas decisões, facilitando a análise da simulação.

Sendo assim, podemos observar na saída dos logs do simulador correspondente à Figura 43, que o VANT está sobrevoando a estação de coleta S2 o encontra em modo de

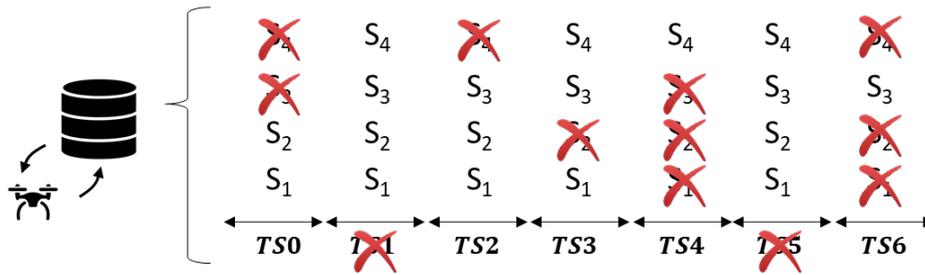


Figura 42 – Base de dados criada pelo VANT.

hibernação durante o período *timeslot* 0, ou seja, esse não é seu *timeslot* ativo de broadcast. Portanto, o VANT adiciona uma restrição à EC2 (SensorID[2]) associada ao *timeslot* atual. Essa restrição é utilizada como base de dados para futura referência, para mudanças de rota. Além disso, o sistema apresenta em sua saída os sensores permitidos e restritos para aquele *timeslot*, apenas para fins de acompanhamento da aprendizagem do VANT.

---

```
14:12:445 / TS: 00 - UAV[1]: Restriction set on SensorID[2] for timeSlot 00
14:12:467 / TS: 00 - UAV[1]: Restricted Sensors on this TimeSlot are: S4 S6 S8 S5 S9 S1 S0 S2
14:12:504 / TS: 00 - UAV[1]: Permitted Sensors on this TimeSlot are: S3 S7
```

---

Figura 43 – VANT alimentando a lista de permissão/restrrição.

Na Figura 44, um VANT ao encontrar uma EC ativa, automaticamente adiciona aquela EC à todas as listas de restrição de todos os *timeslots* das suas tabelas de restrição e permissão, exceto aquele ao qual ela pertence, reduzindo o tamanho das listas dos demais *timeslots* à cada descoberta, aumentando a probabilidade de tomadas de decisão corretas. Além disso, podemos ver também na mesma imagem que ao encontrar uma EC sobre o *timeslot* (10 no caso citado), o VANT reprograma a EC do *timeslot* atual, para seu novo *timeslot* (TS 07) usando as métricas apresentadas na Figura 40. Além disso, como é possível notar, o VANT apresenta a lista de sensores permitidos e restritos para o *timeslot* atual, para fins de acompanhamento.

---

```
12:38:015 / TS: 10 - UAV[1]: Discovered Sensor ID[0] broadcasting on TimeSlot 10
12:38:034 / TS: 10 - UAV[1]: SensorID[0] is now reprogrammed from TimeSlot 10 to 07
12:38:049 / TS: 10 - UAV[1]: Restricted Sensors on this TimeSlot are: S6 S8 S5 S9 S1 S3 S4 S0
12:38:065 / TS: 10 - UAV[1]: Permitted Sensors on this TimeSlot are: S2 S7
```

---

Figura 44 – VANT descobrindo um nova EC.

Através da Figura 45, podemos ver a consulta e utilização da base de dados criada pelo VANT antes de se deslocar para a EC3. Sempre ao chegar ao seu alvo planejado, e antes de se dirigir para seu próximo destino, o VANT realiza o cálculo do tempo em

*timeslots* necessário para chegar ao seu destino, além disso, consulta as tabelas de permissão e restrição para verificação da viabilidade daquele destino, para aquele *timeslot* se é válido ir até aquele local.

---

```
13:37:402 / TS: 13 - UAV[1]: I will arrive approximately at SensorID[3] on timeSlot 13
13:37:416 / TS: 13 - UAV[1]: Restricted Sensors on TimeSlot 13 are: S6 S8 S5 S9 S1 S2 S7 S4 S0
13:37:430 / TS: 13 - UAV[1]: Permitted Sensors on TimeSlot 13 are: S3
```

---

Figura 45 – VANT utilizando a consulta à sua base de dados.

Como um exemplo da aplicação desse banco de dados que o VANT cria e utiliza, a saída dos *logs* na Figura 46 indica uma tomada de decisão por parte do VANT sobre sua base de conhecimento. O VANT antes de se dirigir para a EC3, calculou o *timeslot* de chegada sobre o alvo onde este apontava para um *timeslot* proibido no momento de chegada do VANT, sendo assim, ele avalia seus possíveis outros destinos ainda não descobertos de sua rota, e calculou o tempo em *timeslots* para cada um destes, prevendo seu *timeslot* de chegada. Baseado nisso, ao cruzar com sua base de conhecimento para cada *timeslot*, optou por se dirigir para um outro sensor. No caso, seu destino viável foi a EC1 (S1), com previsão de chegada para o *timeslot* 1, como consequência da sua decisão, a EC foi descoberta em seu *timeslot* corretamente, e reprogramada.

---

```
29:08:759 / TS: 01 - UAV[2]: Sensor[3] is not permitted on timeSlot 03
29:08:805 / TS: 01 - UAV[2]: Checking undiscovered Targets are: S1 S5 S6 S2 S3 S11 S14 S16 S12
29:08:852 / TS: 01 - UAV[2]: Sensor[1] is permitted on timeSlot 01
29:08:895 / TS: 01 - UAV[2]: I made a decision ...
29:10:410 / TS: 02
29:10:461 / TS: 02 - UAV[2]: Discovered Sensor ID[1] broadcasting on TimeSlot 02
29:10:511 / TS: 02 - UAV[2]: SensorID[1] is now reprogrammed from TimeSlot 02 to 01
29:10:546 / TS: 02 - UAV[2]: Restricted Sensors on this TimeSlot are: S8 S2 S4 S0 S13 S17 S15 S7 S9 S10 S18
29:10:584 / TS: 02 - UAV[2]: Permitted Sensors on this TimeSlot are: S3 S5 S6 S11 S14
```

---

Figura 46 – Saída dos registros de tomada de decisão do VANT.

Além disso, existem outras tomadas de decisão que o VANT pode realizar, baseando em sua base de dados sobre os sensores proibidos e permitidos para cada *timeslot*. Apresentado na Figura 47, o VANT não tem para onde ir, ou seja, na sua lista de sensores ainda não descobertos, ao calcular o *timeslot* de chegada para todos seus destinos recaem sobre *timeslots* proibidos para os sensores, portanto, o VANT toma a decisão de permanecer no próprio lugar e aguardar o *timeslot* passar, para aí sim refazer o cálculo e se dirigir para um novo destino.

Todas as tomadas de decisão que os VANTs realizam durante a simulação são armazenadas para análise, mensurando o impacto que as listas de permissão e bloqueio

---

```

13:35:041 / TS: 12 - UAV[1]: Sensor[7] is not permitted on timeSlot 13
13:35:056 / TS: 12 - UAV[1]: Checking unDiscovered Targets are: S2 S7 S3
13:35:069 / TS: 12 - UAV[1]: Sensor[2] is restricted on timeSlot 13
13:35:085 / TS: 12 - UAV[1]: Sensor[7] is restricted on timeSlot 13
13:35:101 / TS: 12 - UAV[1]: Sensor[3] is restricted on timeSlot 12
13:35:119 / TS: 12 - UAV[1]: I have no place to go, ... I will wait here !

```

---

Figura 47 – Saída dos registros de tomada de decisão do VANT.

tem sobre o processo de descoberta dos sensores em solo, possibilitando também tomadas de decisões mais assertivas em relação à onde ir ou não. Na Figura 48 podemos ver graficamente, o VANT sobrevoando o sensor 2 (S2) no *timeslot* 1, onde não o encontrou em modo de transmissão ativo, portanto, é adicionado o S2 no TS1 à base de restrições, sabendo assim que no próximo TS1 o VANT não deverá sobrevoar àquele sensor mais. Adiante, o VANT passa pelo S4 no momento TS2, e novamente o adiciona à sua base de restrições. O VANT segue seu voo para o S1 onde chega no TS4, e o adiciona à sua base. Em seguida, o VANT calcula o TS de chegada para o S2, e confere em sua base de dados que este *timeslot* contém uma restrição para este sensor, portanto o VANT se dirige para o próximo sensor (S4) de sua rota, evitando assim de passar sobre um sensor inativo para aquele *timeslot*.

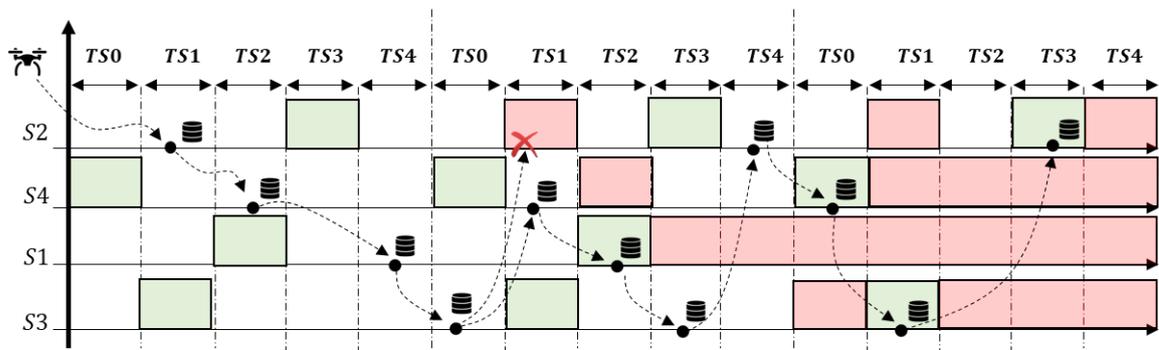


Figura 48 – Representação gráfica do sistema de restrições do VANT.

## 6.1 Metodologia

Utilizando o simulador propriamente desenvolvido e configurado, realizamos 2000 simulações para cada cenário  $C_1, \dots, C_7$  de acordo com a Tabela 1, com o intuito de determinar os fatores que interferem diretamente no processo de descoberta e reprogramação dos sensores, de acordo com a rota previamente definida pelo VANT.

Portanto, para cada simulação foram analisados os quesitos abaixo, além de coletados os valores sintéticos (médios) de todas as simulações, foram também coletados os

Característica	C1	C2	C3	C4	C5	C6	C7
Vel. Média	25 m/s	25m/s	15m/s	15m/s	Wait	25m/s	25m/s
Percurso	NearestPoint						
Raio de Comunic.	100 metros						
Área Simulação	1212 x 517m <sup>2</sup>						
Qtdade de ECs	5, 10, 15 e 20 Estações						
Tempo Broadcast	2 TS						1 TS
Autonomia do VANT	10km	10km	10km	10km	10min	10km	10km
Individual TS	Sim	Não	Sim	Não	Sim	Sim	Sim

Tabela 1 – Cenários configurados para Simulações.

dados analíticos, tais como máximos, e mínimos que também serão apresentados para uma melhor análise dos extremos ocorridos durante as simulações.

Os dados abaixo foram coletados e armazenados em arquivos extensão CSV (separado por vírgulas) para posterior análise, até que o VANT realizasse a descoberta de todos os sensores habilitados para o cenário.

1. Quantidade de VANTs
2. Distância percorrida por VANT
3. Tempo de voo por VANT
4. Tempo total de simulação
5. Quantidade de saltos por VANT
6. Quantidade de decisões tomadas por VANT
7. Distância da rota original do cenário
8. Quantidade de timeslots do cenário
9. Quantidade total de timeslots até final da simulação
10. Coordenadas do VANT em função do tempo
11. Estado das Estações de Coleta (Transmitindo/Hibernando)
12. Estado das Conexões do VANT com as ECs
13. Quantidade de voos perdidos pelas EC
14. Tempo de Broadcast das ECs (antes e depois da reprogramação)
15. Percentuais de redução de percurso e tempo para o VANT

## 16. Percentual de bateria/autonomia do VANT

**Cenário 01:** Para o primeiro conjunto de simulações, utilizamos os critérios abaixo.

- Velocidade média do VANT: 25m/s
- Planejamento de Percurso: NearestPoint
- Raio de Comunicação das ECs: 100m
- Área de Simulação: 1212m x 517m, aproximadamente  $700.000m^2$
- Quantidade de ECs: 5, 10, 15 e 20 estações
- Tempo de Broadcast das ECs: 2 timeslots
- Autonomia do VANT: 10km
- Sensores em timeslots individuais

Na Figura 49 são apresentados a quantidade de VANTs utilizados para identificar e reprogramar todas as ECs.

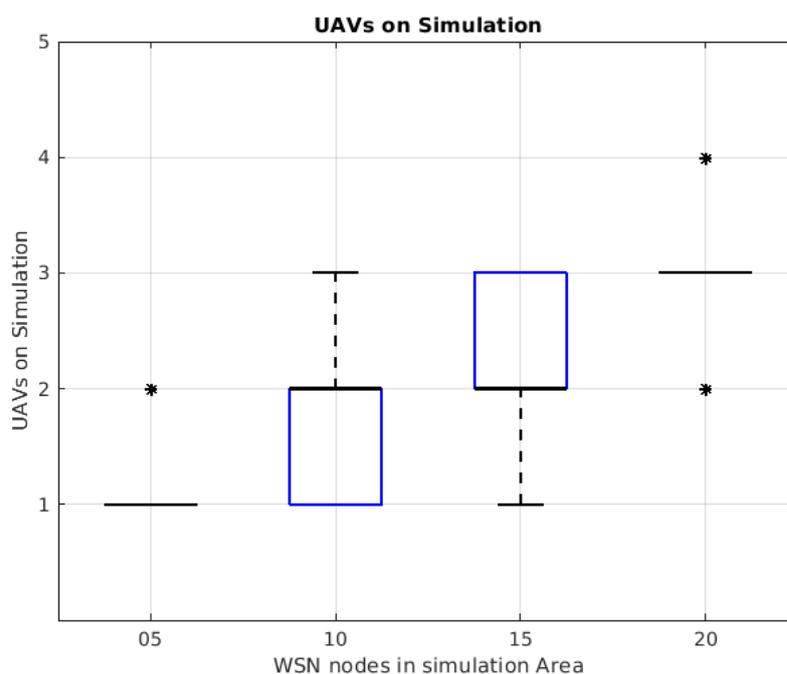


Figura 49 – Quantidade de VANTs – Cenário 01.

Na Figura 50, é apresentada a distância total percorrida pelos VANTs em função da quantidade de Sensores no cenário de simulação.

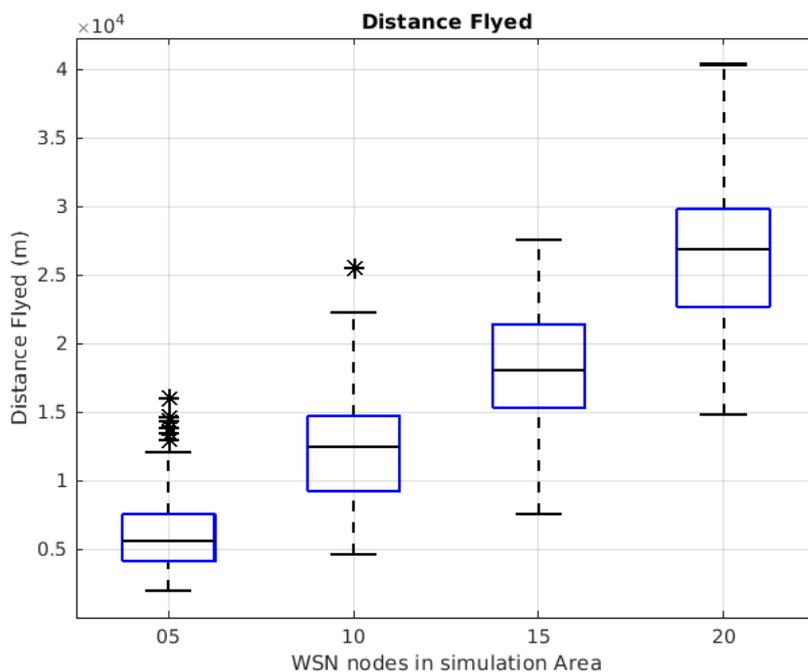


Figura 50 – Distância Percorrida – Cenário 01.

A Figura 51 apresenta o tempo de voo dos VANTs no ambiente de simulação até que todos os sensores sejam descobertos e reprogramados.

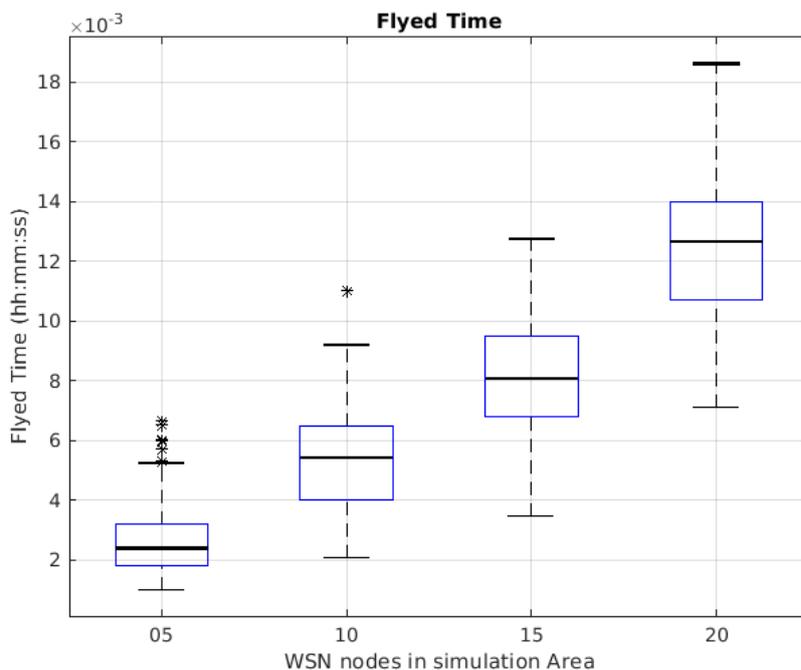


Figura 51 – Tempo de voo – Cenário 01.

A Figura 52 representa a quantidade de saltos realizados pelo VANTs no cenário

até a descoberta e reprogramação de todos os sensores.

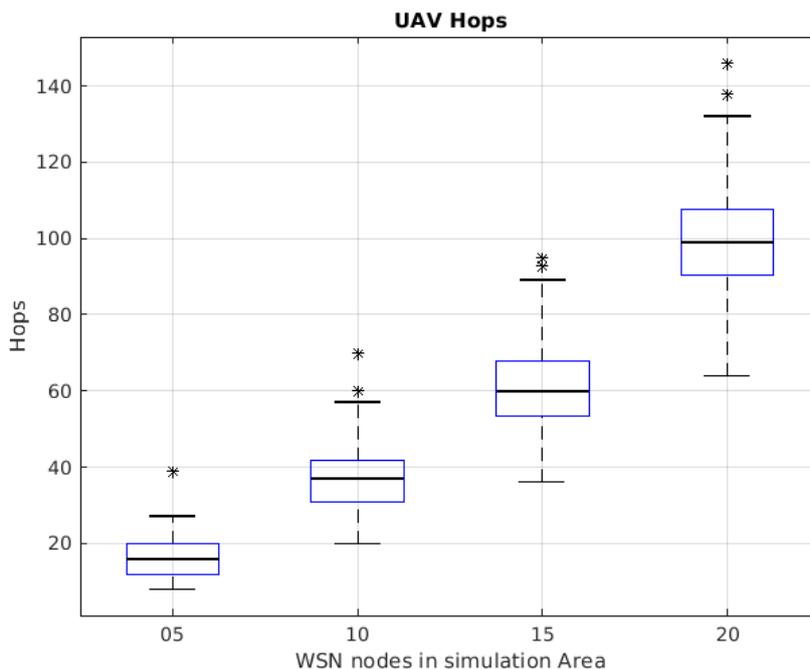


Figura 52 – Quantidade de Saltos – Cenário 01.

Na Figura 53, temos a apresentação da quantidade de decisões tomadas pelos VANTs durante a fase de descoberta e reprogramação dos sensores.

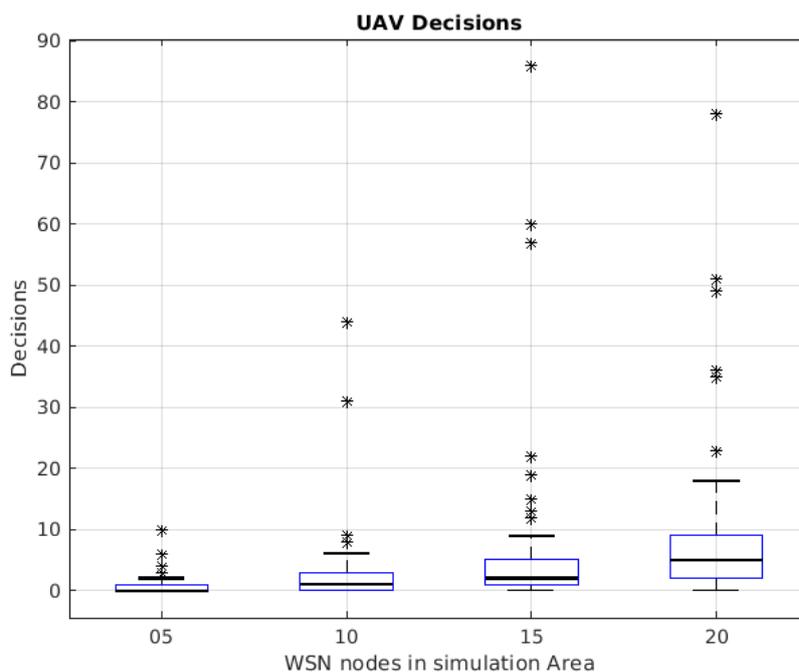


Figura 53 – Quantidade de Decisões – Cenário 01.

Na Figura 54, temos a quantidade total de *timeslots* contabilizados que se passaram até a descoberta e reprogramação de todos os sensores no ambiente de simulação.

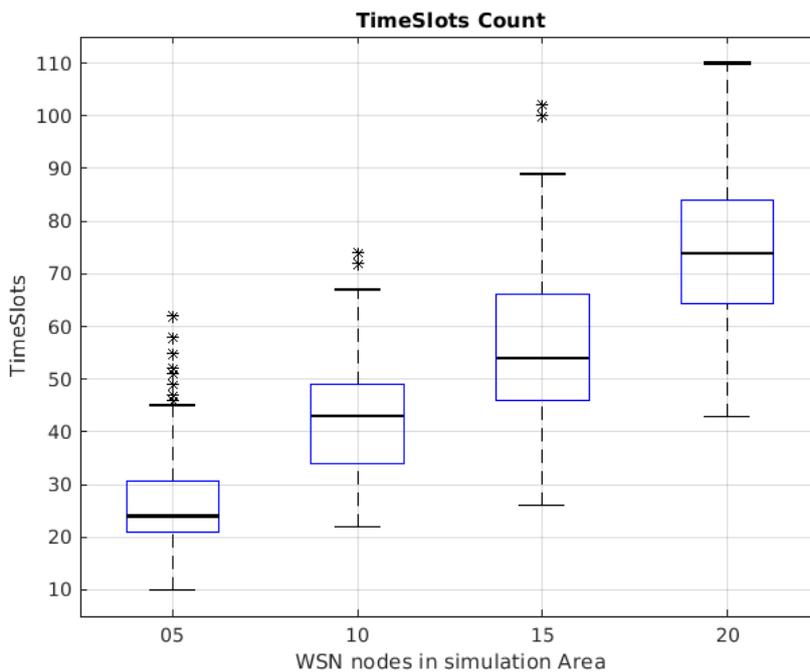


Figura 54 – Quantidade de *Timeslots* – Cenário 01.

Na Figura 55 temos a apresentação da média entre a distância total da rota NearestPoint definida pelo VANT para o percurso, e a distância total percorrida até a identificação e reprogramação de todos os sensores, para cada cenário.

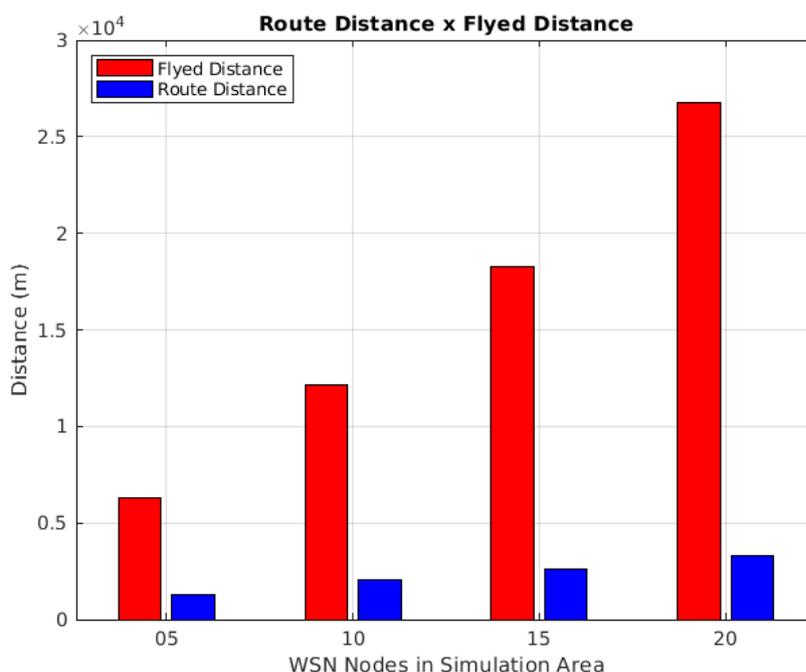


Figura 55 – Distância Percorrida e Distância da Rota – Cenário 01.

**Cenário 02:** Para o segundo conjunto de simulações, utilizamos os critérios abaixo, a diferença em relação ao cenário anterior, é que aqui os *timeslots* são compartilhados entre as Estações de Coleta, e não individuais.

- Velocidade média do VANT: 25m/s
- Planejamento de Percurso: NearestPoint
- Raio de Comunicação das ECs: 100m
- Área de Simulação: 1212m x 517m, aproximadamente  $700.000m^2$
- Quantidade de ECs: 5, 10, 15 e 20 estações
- Tempo de Broadcast das ECs: 2 *timeslots*
- Autonomia do VANT: 10km
- Sensores em timeslots compartilhados

Na Figura 56 são apresentados a quantidade de VANTs utilizados para identificar e reprogramar todas as ECs.

Na Figura 57, é apresentada a distância total percorrida pelos VANTs em função da quantidade de Sensores no cenário de simulação.

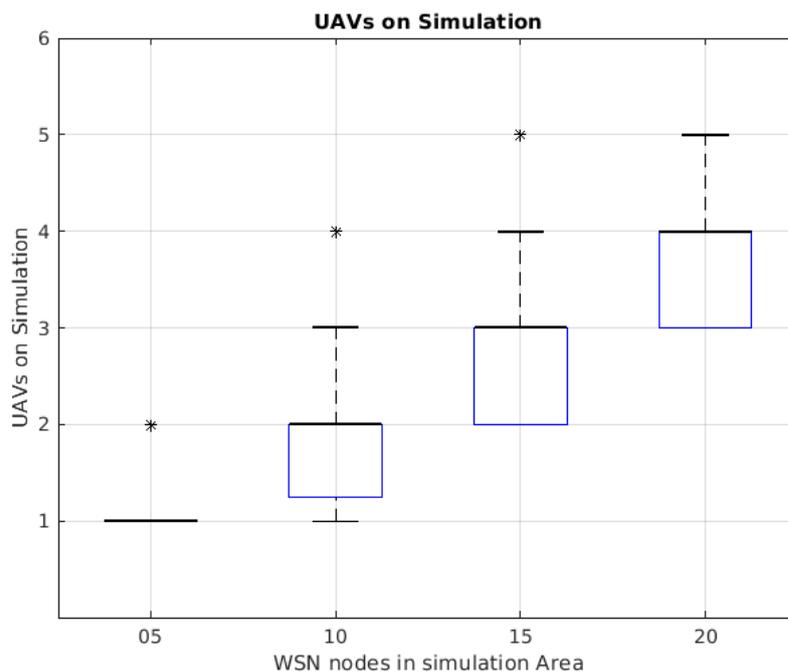


Figura 56 – Quantidade de VANTs – Cenário 02.

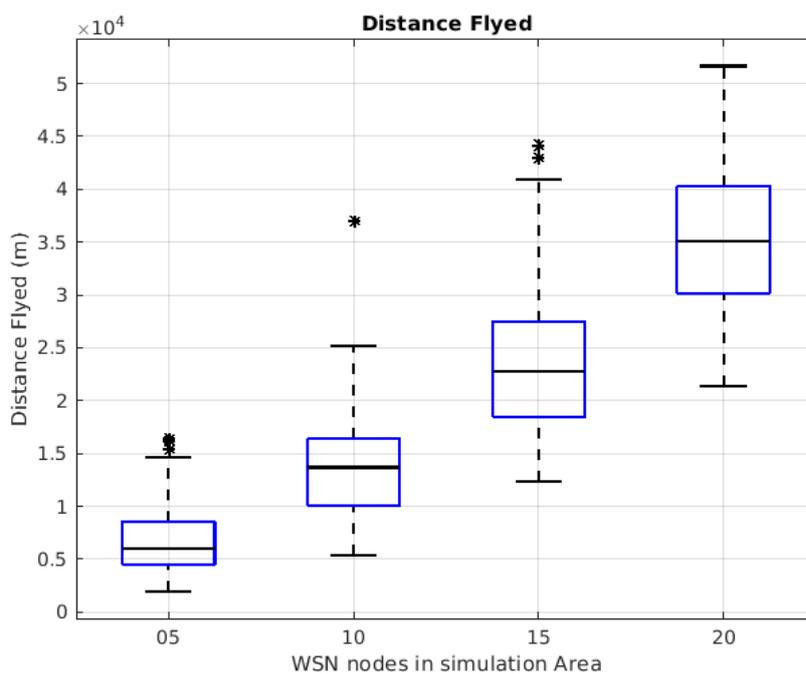


Figura 57 – Distância Percorrida – Cenário 02.

A Figura 58 apresenta o tempo de voo dos VANTs no ambiente de simulação até que todos os sensores sejam descobertos e reprogramados.

A Figura 59 representa a quantidade de saltos realizados pelo VANTs no cenário até

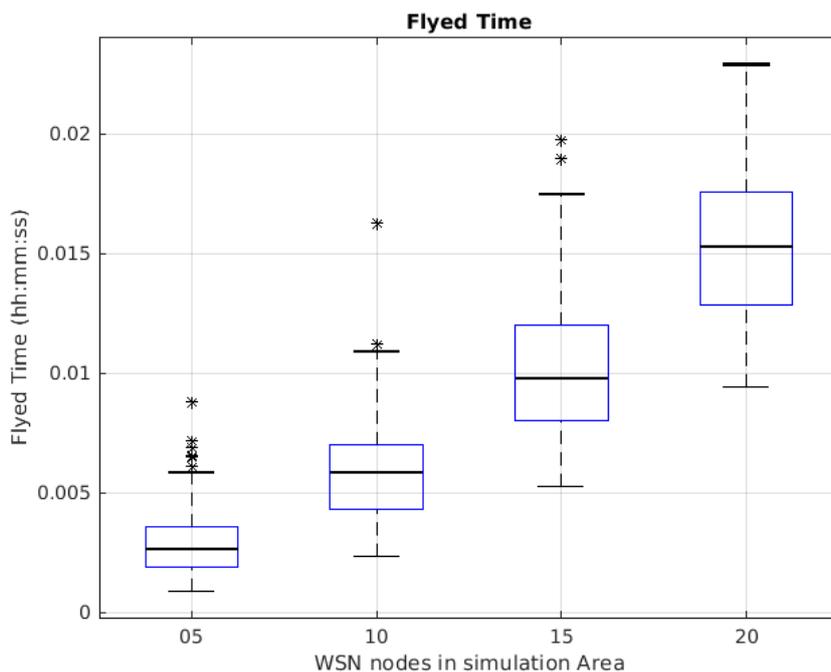


Figura 58 – Tempo de Voo – Cenário 02.

a descoberta e reprogramação de todos os sensores. A Figura 52 representa a quantidade de saltos realizados pelo VANTs no cenário até a descoberta e reprogramação de todos os sensores.

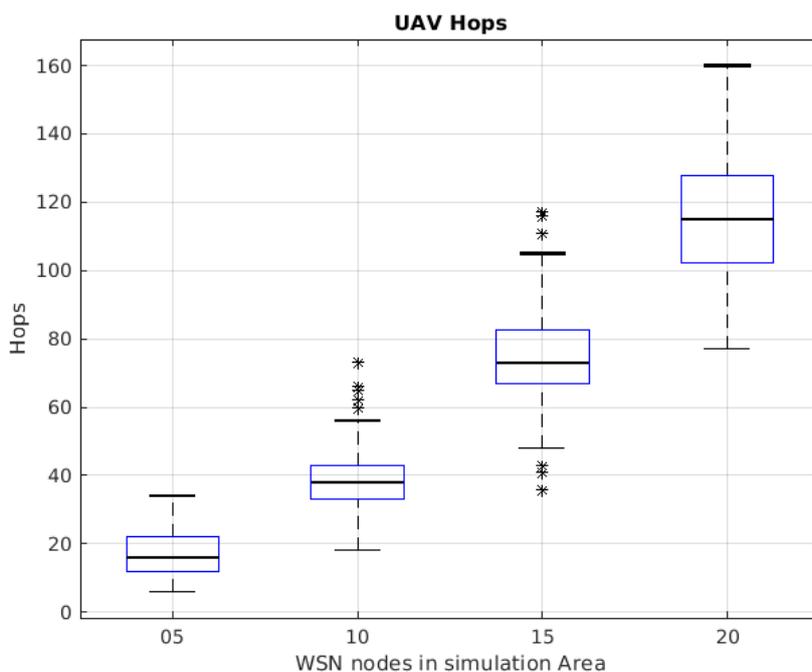


Figura 59 – Quantidade de Saltos – Cenário 02.

Na Figura 60 temos a apresentação da quantidade de decisões tomadas pelos VANTs durante a fase de descoberta e reprogramação dos sensores.

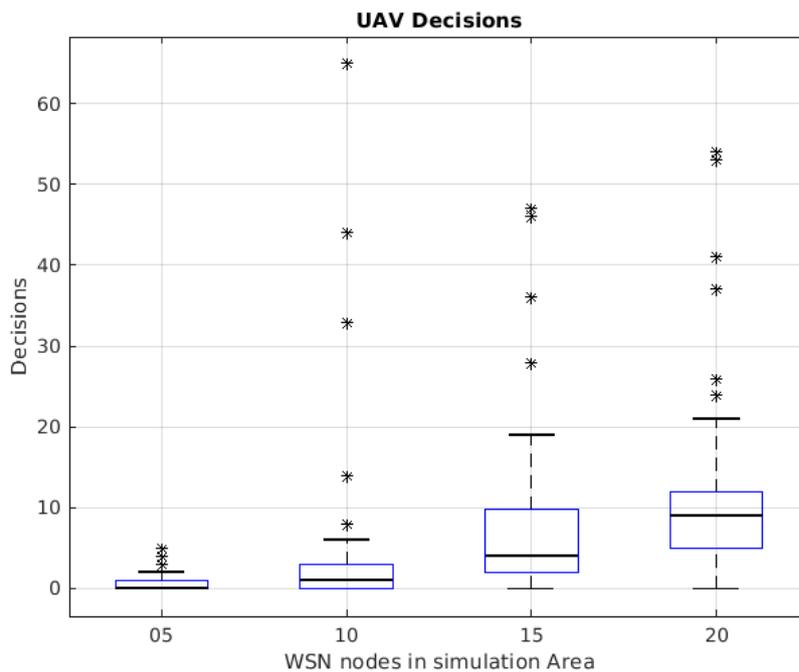


Figura 60 – Quantidade de decisões tomadas pelos VANTs – Cenário 02.

Na Figura 61, temos a quantidade total de *timeslots* contabilizados que se passaram até a descoberta e reprogramação de todos os sensores no ambiente de simulação.

Na Figura 62 temos a apresentação da média entre a distância total da rota NearestPoint, e a distância total percorrida pelos VANTs para cada cenário.

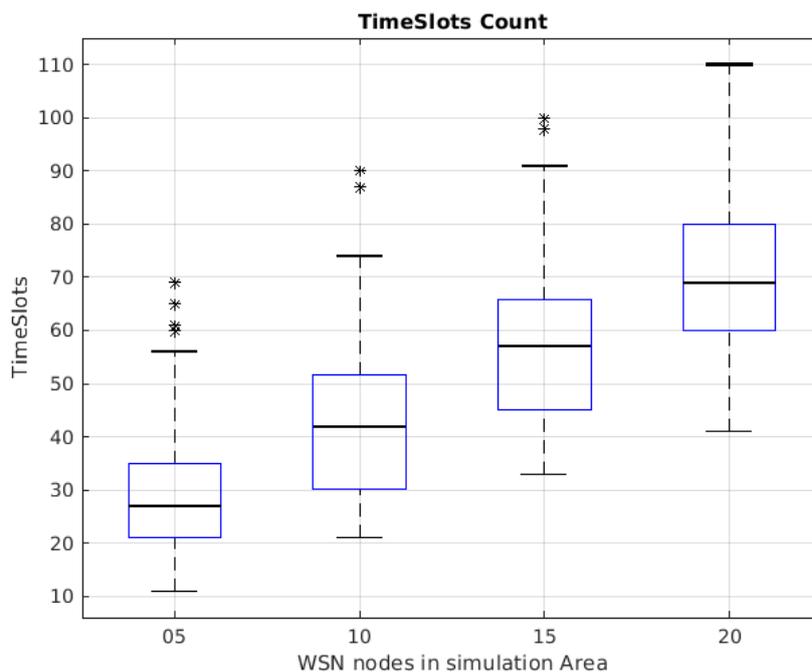


Figura 61 – Quantidade de Timeslots – Cenário 02.

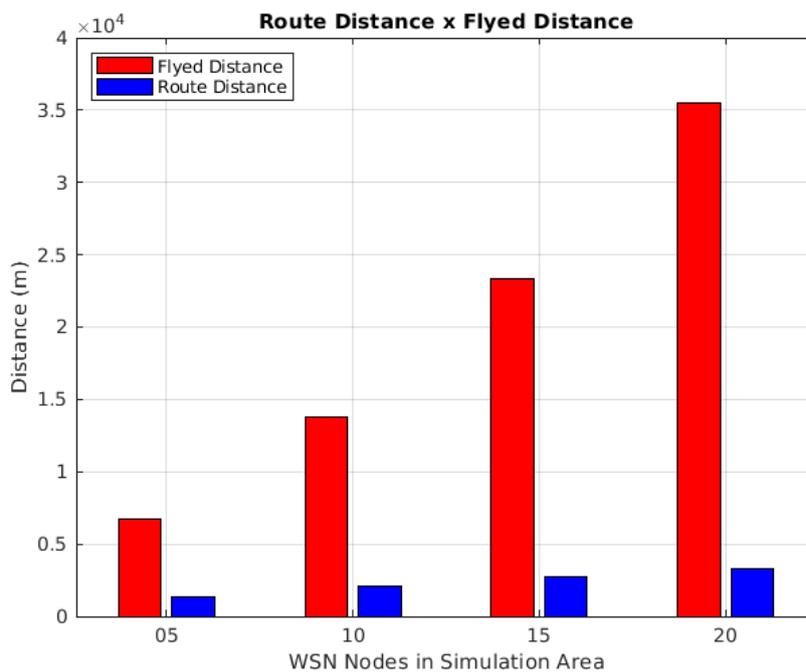


Figura 62 – Distância Percorrida e Distância da Rota – Cenário 02.

**Cenário 03:** Para a terceira simulação, utilizamos os critérios abaixo. Em relação ao primeiro cenário, variamos a velocidade do VANT.

- Velocidade média do VANT: 15m/s

- Planejamento de Percurso: NearestPoint
- Raio de Comunicação das ECs: 100m
- Área de Simulação: 1212m x 517m, aproximadamente  $700.000m^2$
- Quantidade de ECs: 5, 10, 15 e 20 estações
- Tempo de Broadcast das ECs: 2 *timeslots*
- Autonomia do VANT: 10km
- Sensores em timeslots individuais

Na Figura 63, é apresentada a quantidade de VANTs utilizados par identificar e reprogramar todas as ECs.

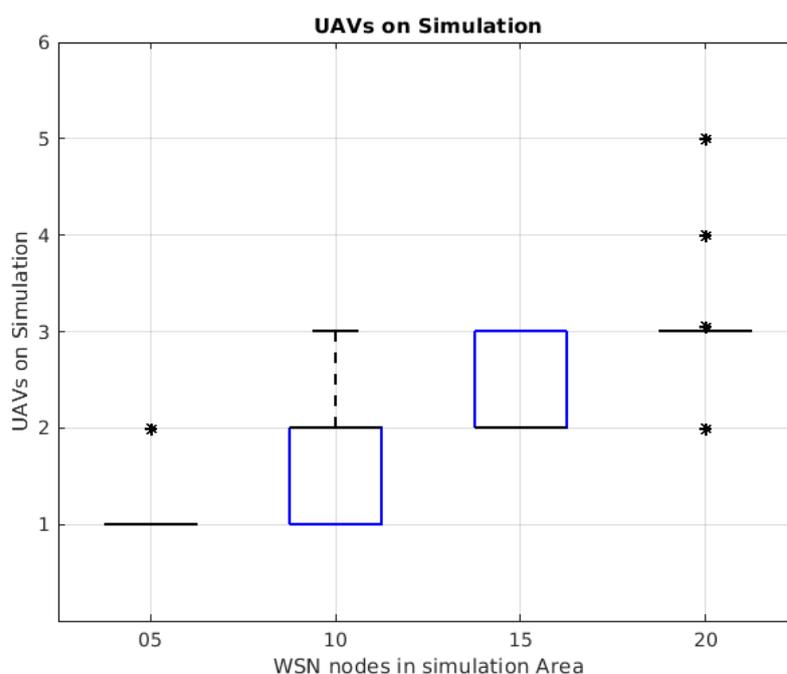


Figura 63 – Quantidade de VANTs – Cenário 03.

De acordo com a Figura 64, temos a distância total percorrida pelos VANTs em função da quantidade de Sensores no cenário de simulação.

A Figura 65 apresenta o tempo de voo dos VANTs no ambiente de simulação até que todos os sensores sejam descobertos e reprogramados.

A Figura 66 representa a quantidade de saltos realizados pelo VANTs no cenário até a descoberta e reprogramação de todos os sensores.

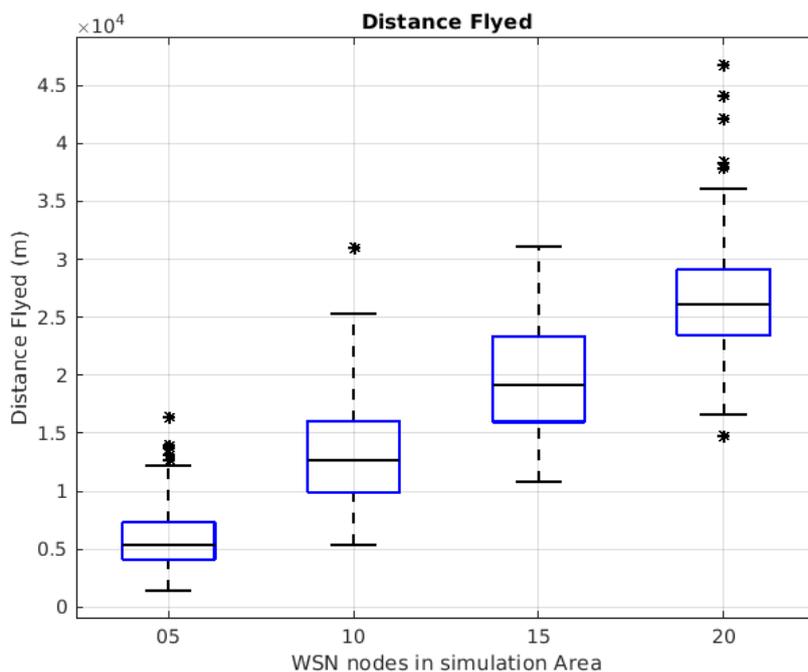


Figura 64 – Distância Percorrida – Cenário 03.

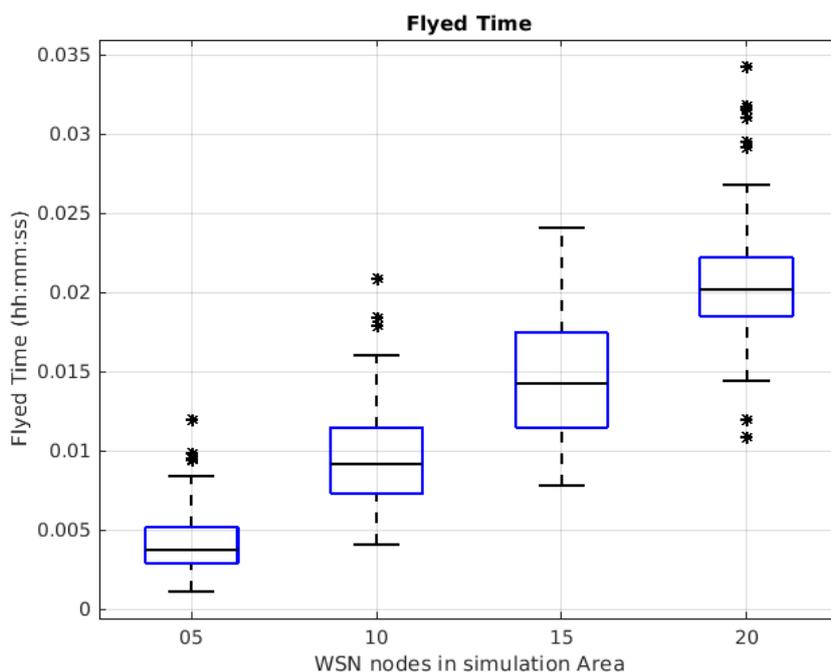


Figura 65 – Tempo de voo – Cenário 03.

Na Figura 67, temos a apresentação da quantidade de decisões tomadas pelos VANTs durante a fase de descoberta e reprogramação dos sensores.

Na Figura 68, temos a quantidade total de *timeslots* contabilizados que se passaram

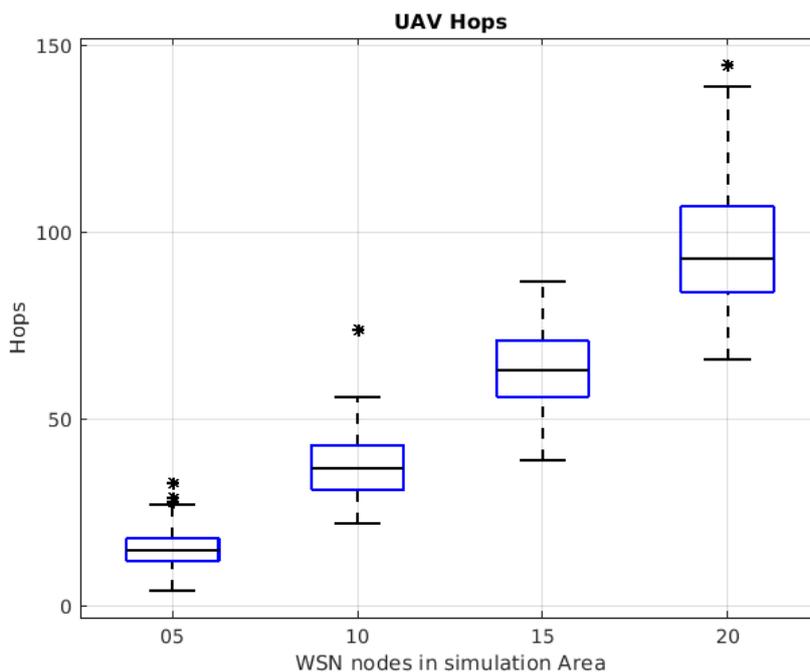


Figura 66 – Quantidade de Saltos – Cenário 03.

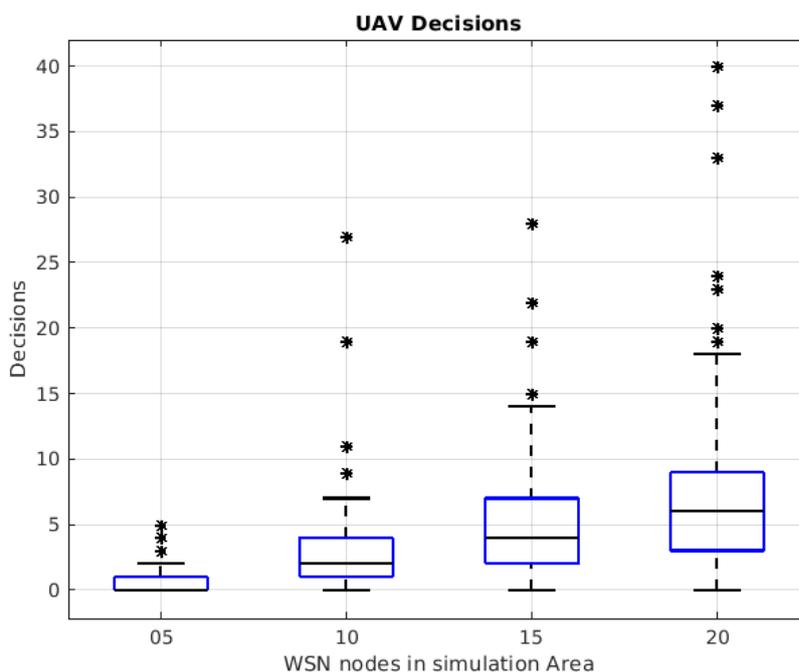


Figura 67 – Quantidade de Decisões – Cenário 03.

até a descoberta e reprogramação de todos os sensores no ambiente de simulação.

Na Figura 69 temos a apresentação da média entre a distância total da rota NearestPoint, e a distância total percorrida pelos VANTs para cada cenário.

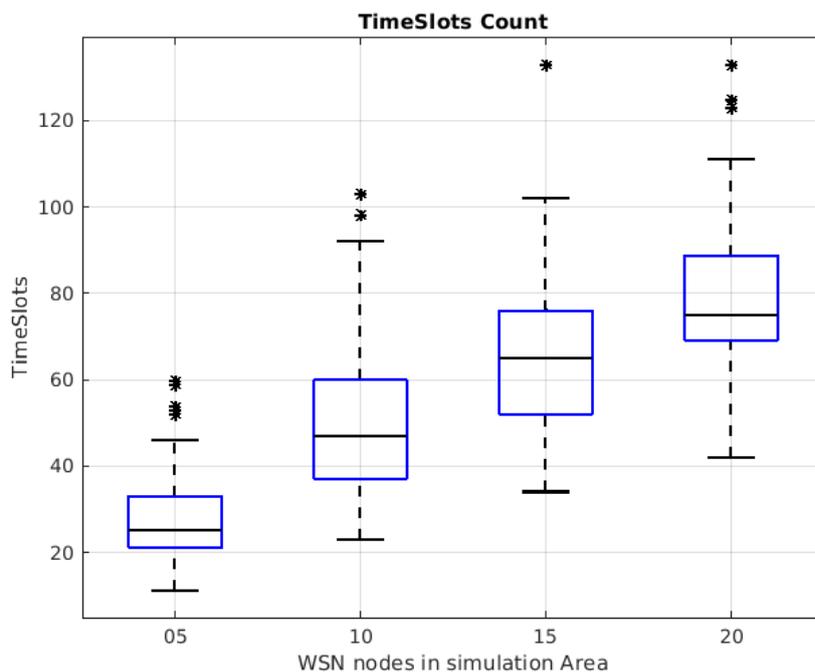


Figura 68 – Quantidade de Timeslots – Cenário 03.

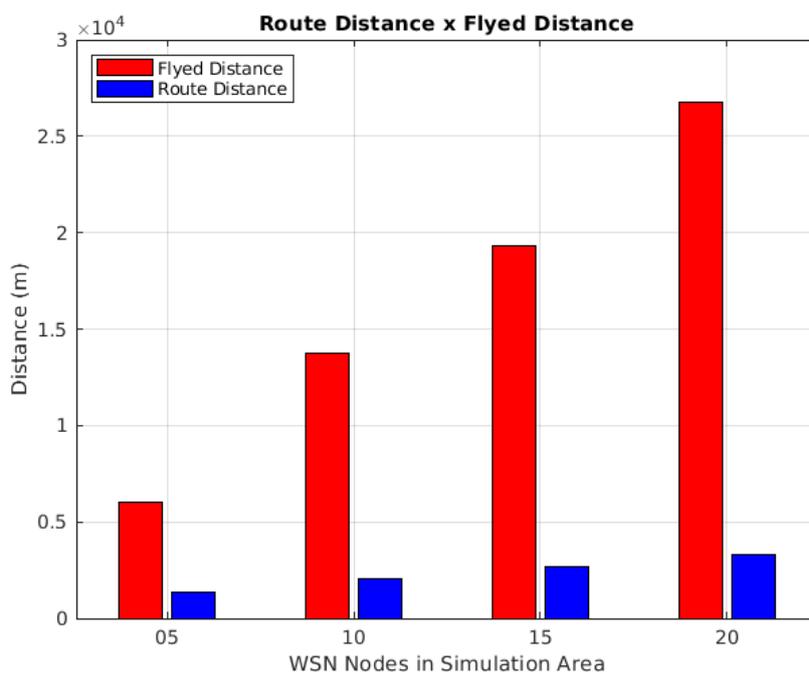


Figura 69 – Distância Percorrida e Distância da Rota – Cenário 03.

**Cenário 04:** Para o quarto cenário, variamos a velocidade média do VANT em relação ao primeiro cenário, além disso, utilizamos *timeslots* compartilhados, e não individuais para as ECs.

- Velocidade média do VANT: 15m/s
- Planejamento de Percurso: NearestPoint
- Raio de Comunicação das ECs: 100m
- Área de Simulação: 1212m x 517m, aproximadamente  $700.000m^2$
- Quantidade de ECs: 5, 10, 15 e 20 estações
- Tempo de Broadcast das ECs: 2 *timeslots*
- Autonomia do VANT: 10km
- Sensores em timeslots compartilhados

Na Figura 70, é apresentada a quantidade de VANTs utilizados para identificar e reprogramar todas as ECs.

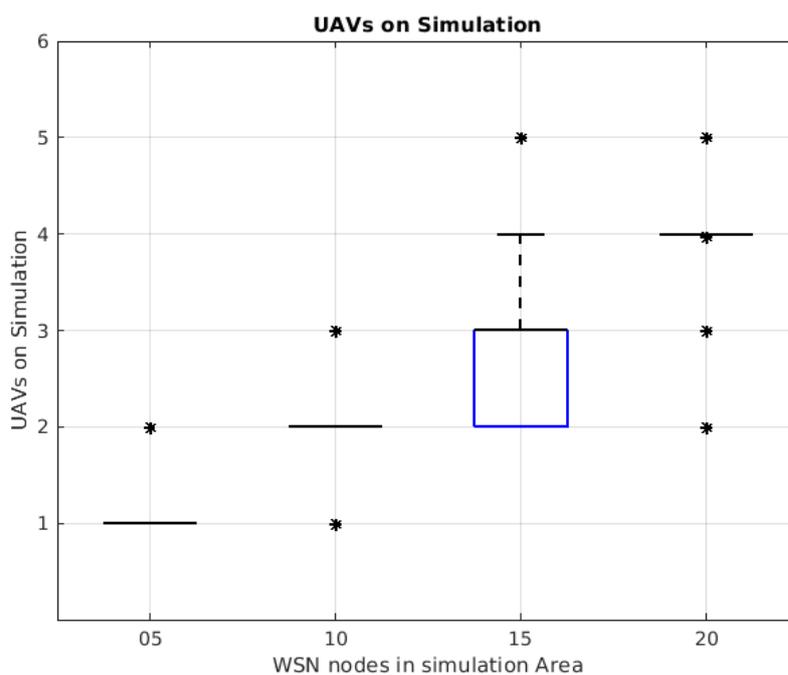


Figura 70 – Quantidade de VANTs – Cenário 04.

De acordo com a Figura 71, temos a distância total percorrida pelos VANTs em função da quantidade de Sensores no cenário de simulação.

A Figura 72 apresenta o tempo de voo dos VANTs no ambiente de simulação até que todos os sensores sejam descobertos e reprogramados.

A Figura 73 representa a quantidade de saltos realizados pelo VANTs no cenário até a descoberta e reprogramação de todos os sensores.

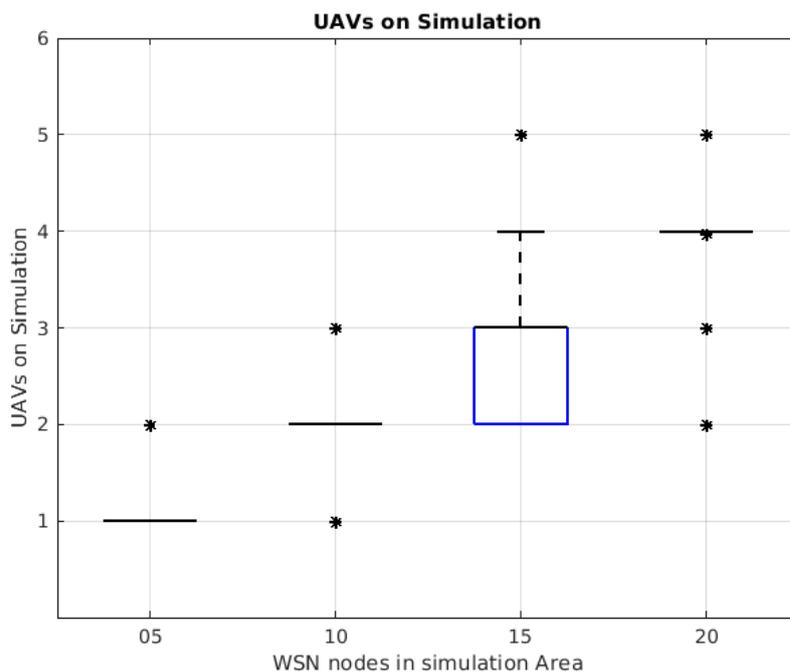


Figura 71 – Distância Percorrida – Cenário 04.

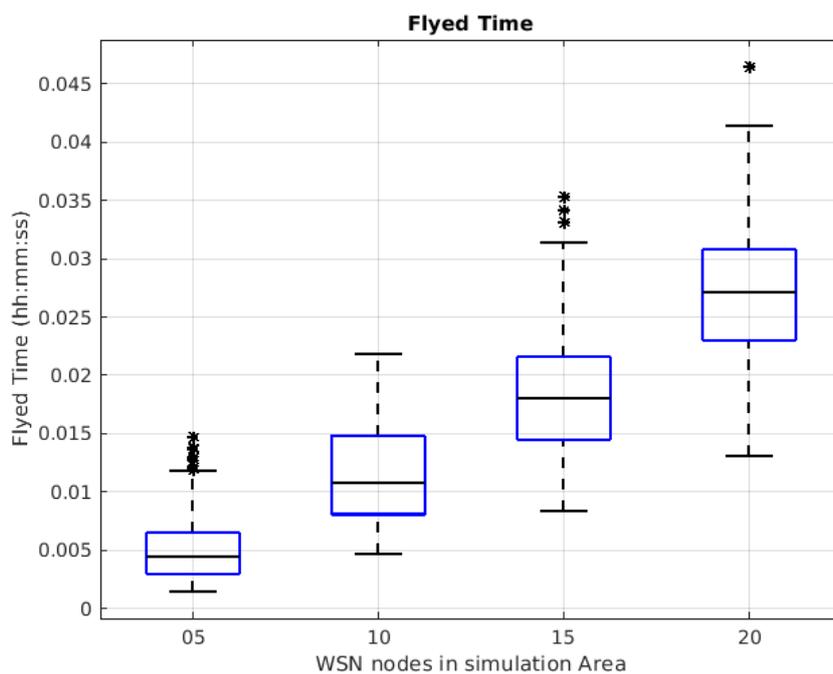


Figura 72 – Tempo de Voo – Cenário 04.

Na Figura 74, temos a apresentação da quantidade de decisões tomadas pelos VANTs durante a fase de descoberta e reprogramação dos sensores.

Na Figura 75, temos a quantidade total de *timeslots* contabilizados que se passaram

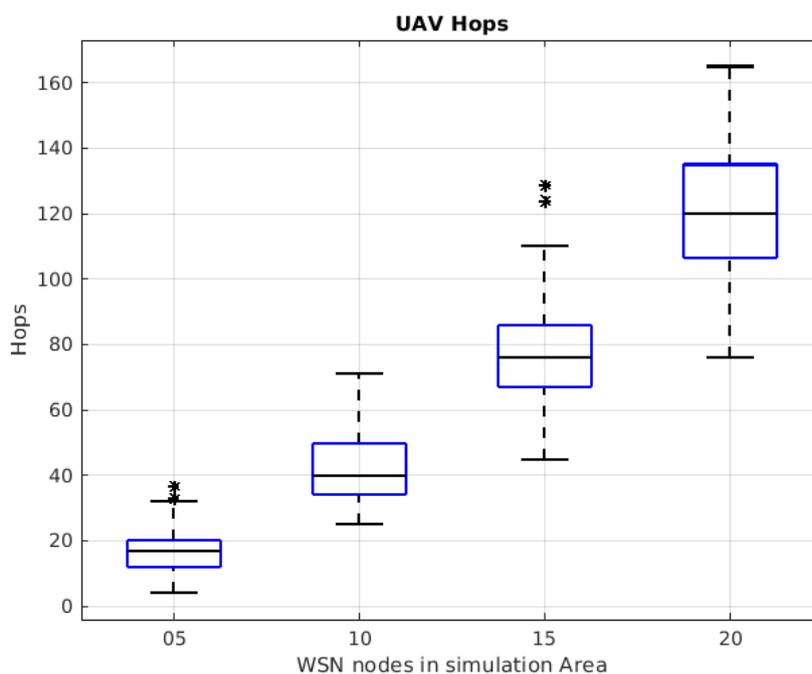


Figura 73 – Quantidade de Saltos – Cenário 04.

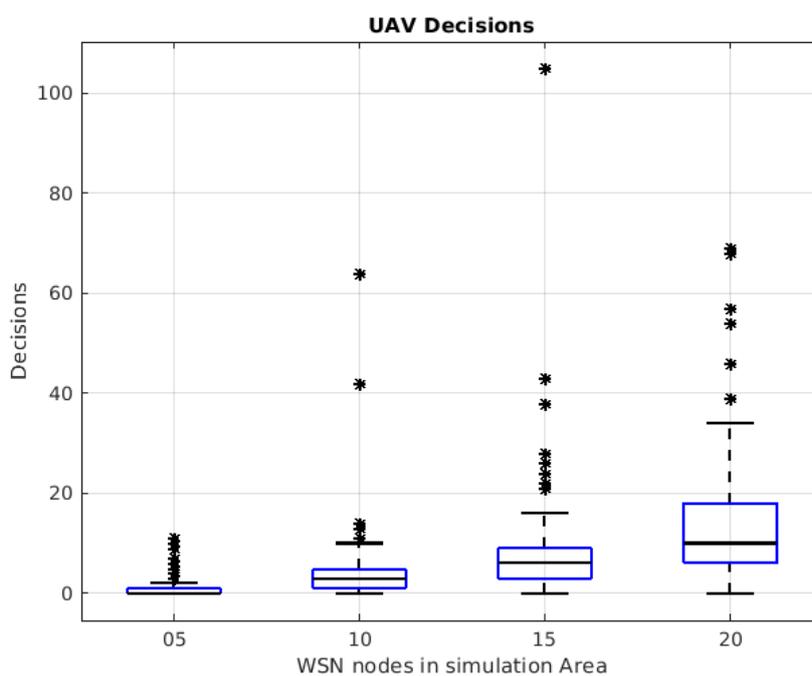


Figura 74 – Quantidade de Decisões – Cenário 04.

até a descoberta e reprogramação de todos os sensores no ambiente de simulação.

Na Figura 76 temos a apresentação da média entre a distância total da rota NearestPoint definida pelo VANT para o percurso, e a distância total percorrida até a

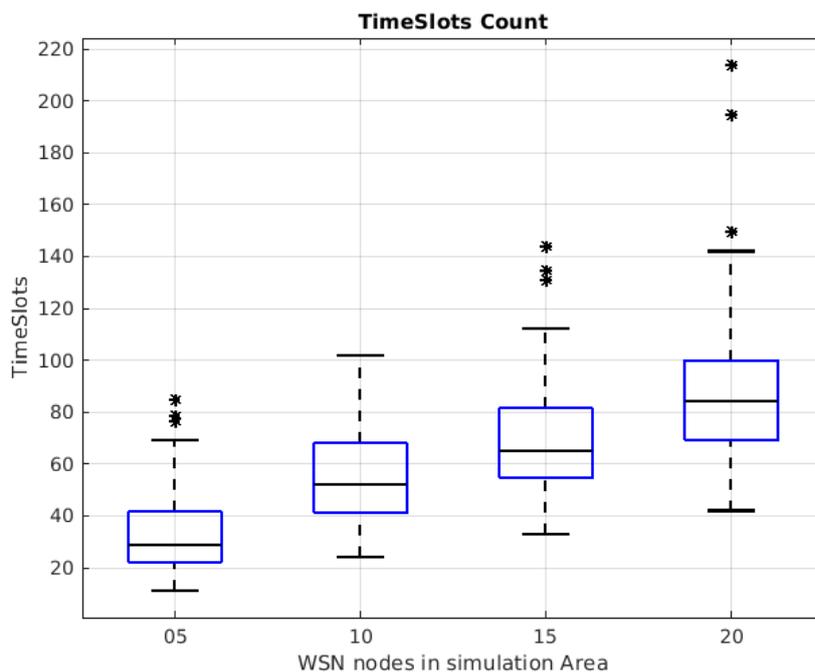


Figura 75 – Quantidade de Timeslots – Cenário 04.

identificação e reprogramação de todos os sensores, para cada cenário.

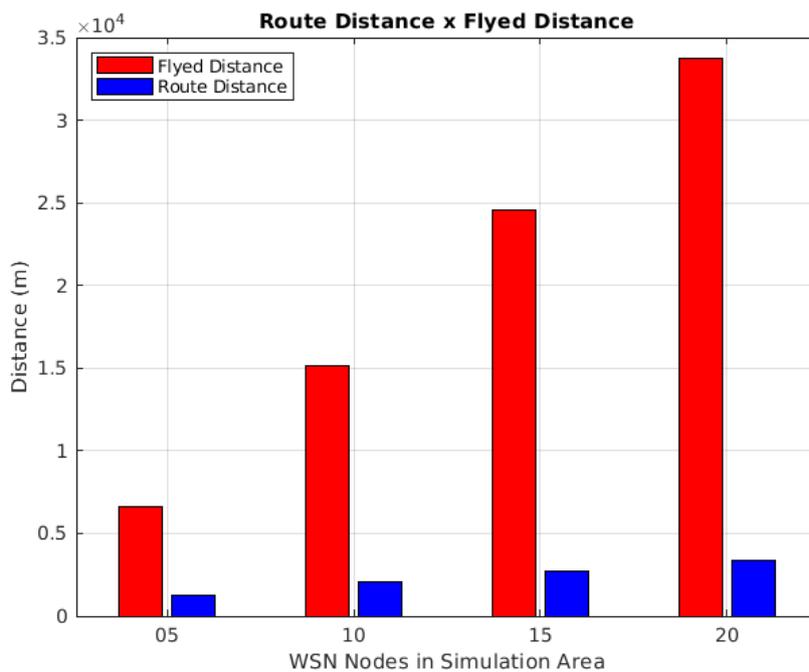


Figura 76 – Distância Percorrida e Distância da Rota – Cenário 04.

**Cenário 05:** Para a quinta simulação, utilizamos o modelo simplificado onde o VANT aguarda sobre a estação de coleta até que a mesma entre em modo de broadcast, sendo este o critério para que o VANT se dirija para o próximo sensor de sua rota, sendo este um modelo comparativo em relação aos demais.

- Velocidade média do VANT: 25m/s
- Planejamento de Percurso: NearestPoint
- Raio de Comunicação das ECs: 100m
- Área de Simulação: 1212m x 517m, aproximadamente  $700.000m^2$
- Quantidade de ECs: 5, 10, 15 e 20 estações
- Tempo de Broadcast das ECs: 2 timeslots
- Autonomia do VANT: 10 minutos
- Sensores em timeslots individuais

Na Figura 77, é apresentada a quantidade de VANTs utilizados para identificar e reprogramar todas as ECs.

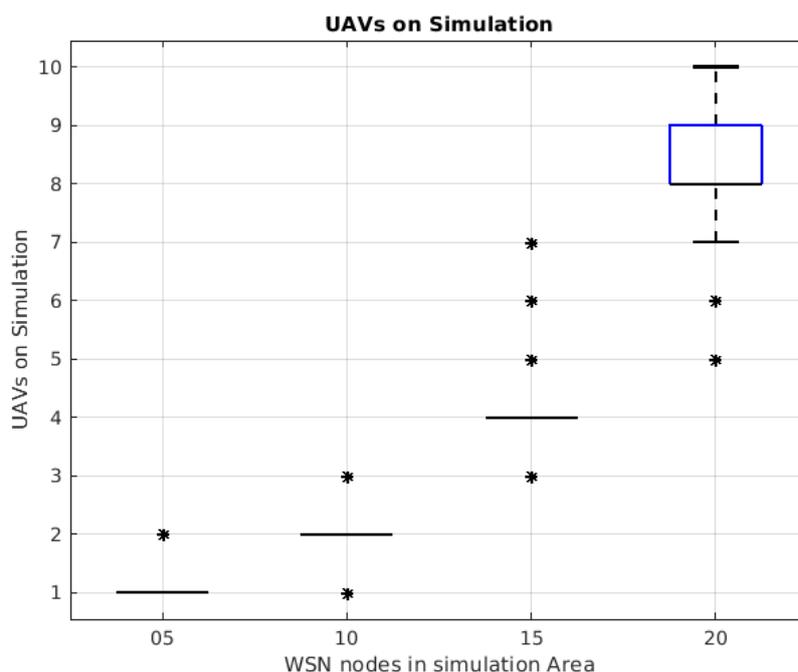


Figura 77 – Quantidade de VANTs – Cenário 05.

De acordo com a Figura 78, temos a distância total percorrida pelos VANTs em função da quantidade de Sensores no cenário de simulação.

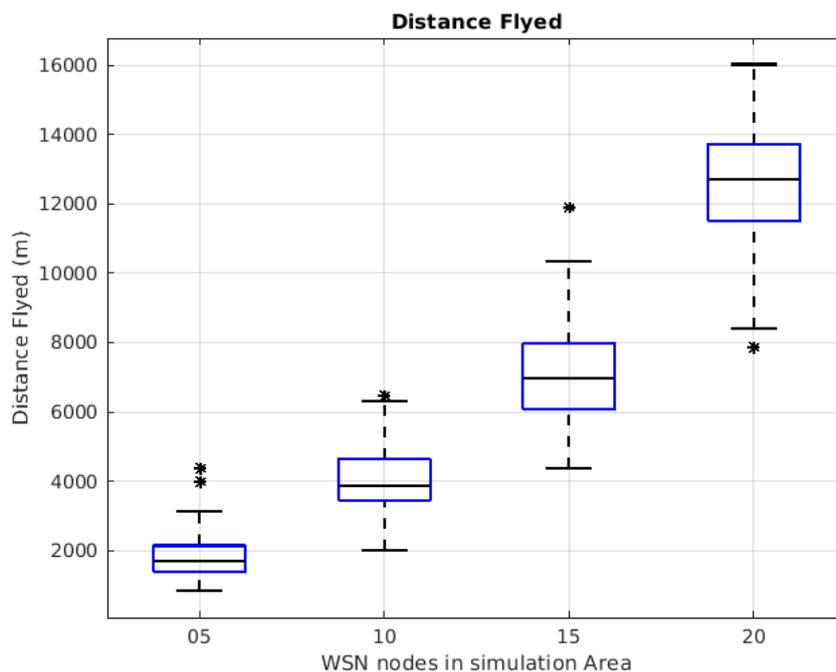


Figura 78 – Distância Percorrida – Cenário 05.

A Figura 79 apresenta o tempo de voo dos VANTs no ambiente de simulação até que todos os sensores sejam descobertos e reprogramados.

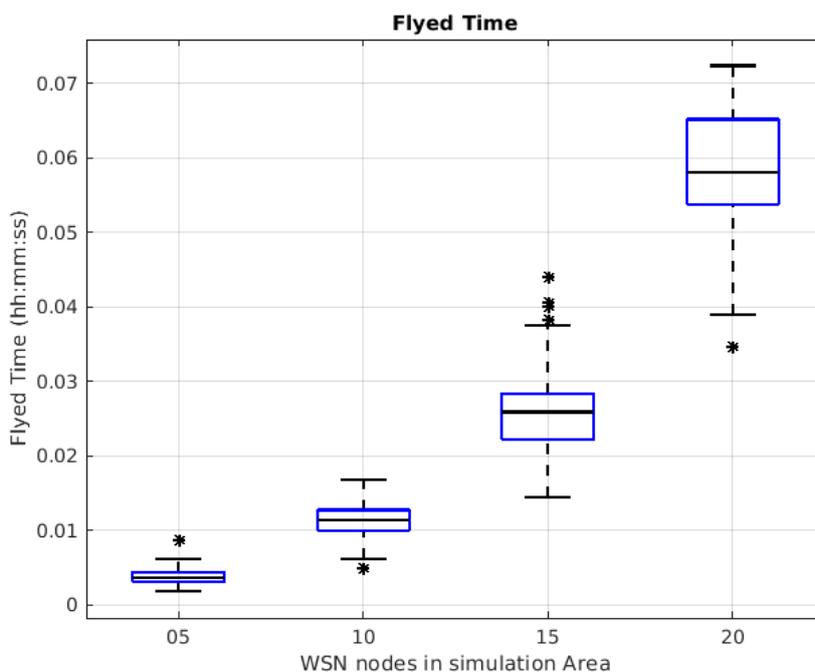


Figura 79 – Tempo de Voo – Cenário 05.

Na Figura 80, temos a apresentação da quantidade de decisões tomadas pelos

VANTs durante a fase de descoberta e reprogramação dos sensores.

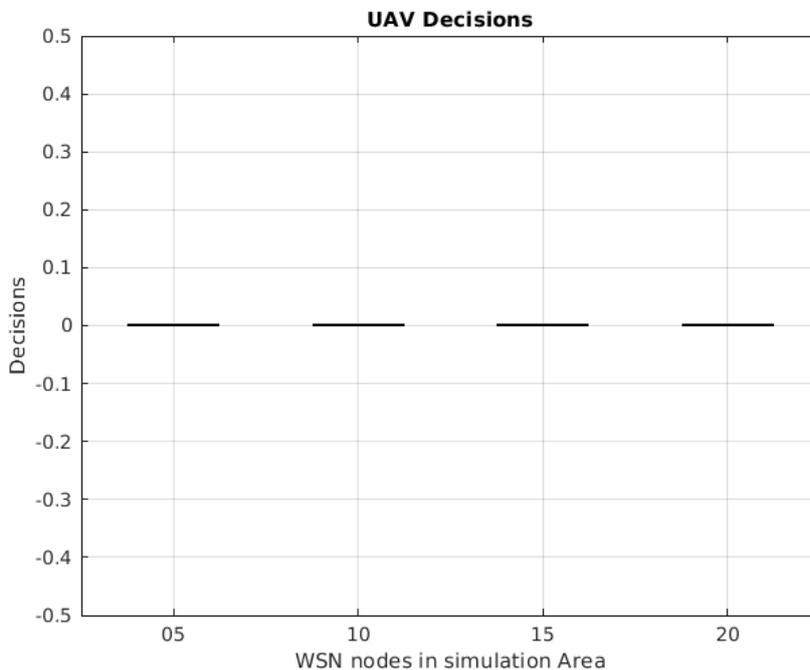


Figura 80 – Quantidade de Decisões – Cenário 05.

Na Figura 81, temos a quantidade total de *timeslots* contabilizados que se passaram até a descoberta e reprogramação de todos os sensores no ambiente de simulação.

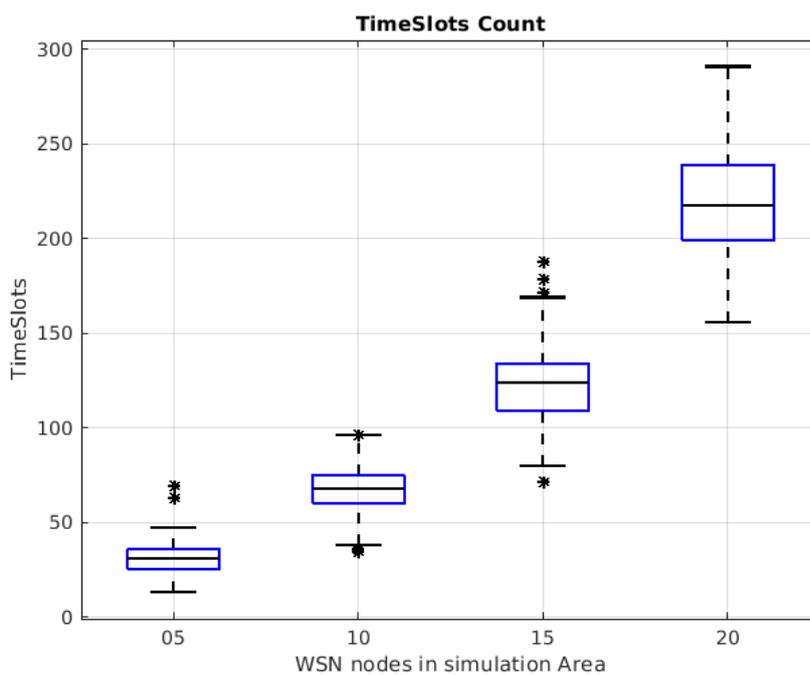


Figura 81 – Quantidade de Timeslots – Cenário 05.

**Cenário 06:** Para a sexta simulação, utilizaremos o primeiro cenário como referência, porém o VANT irá trabalhar com seu sistema de aprendizado desligado. O objetivo desse cenário é identificar o impacto que o sistema de decisão agrega sobre o modelo.

- Velocidade média do VANT: 25m/s
- Planejamento de Percurso: NearestPoint
- Raio de Comunicação das ECs: 100m
- Área de Simulação: 1212m x 517m, aproximadamente  $700.000m^2$
- Quantidade de ECs: 5, 10, 15 e 20 estações
- Tempo de Broadcast das ECs: 2 *timeslots*
- Autonomia do VANT: 10km
- Sensores em timeslots individuais
- Sistema de aprendizado (listas de permissão e restrição desativados)

Na Figura 82, é apresentada a quantidade de VANTs utilizados para identificar e reprogramar todas as ECs.

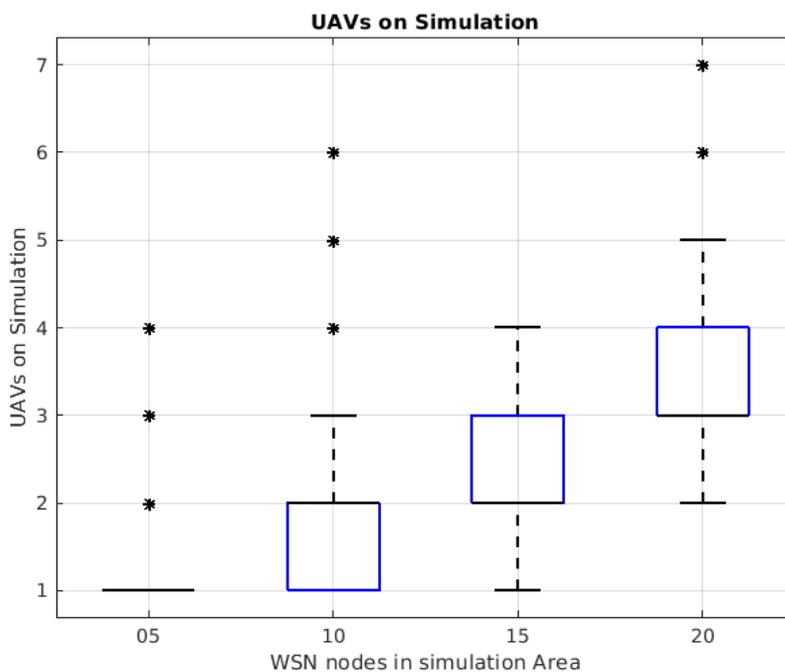


Figura 82 – Quantidade de VANTs – Cenário 06.

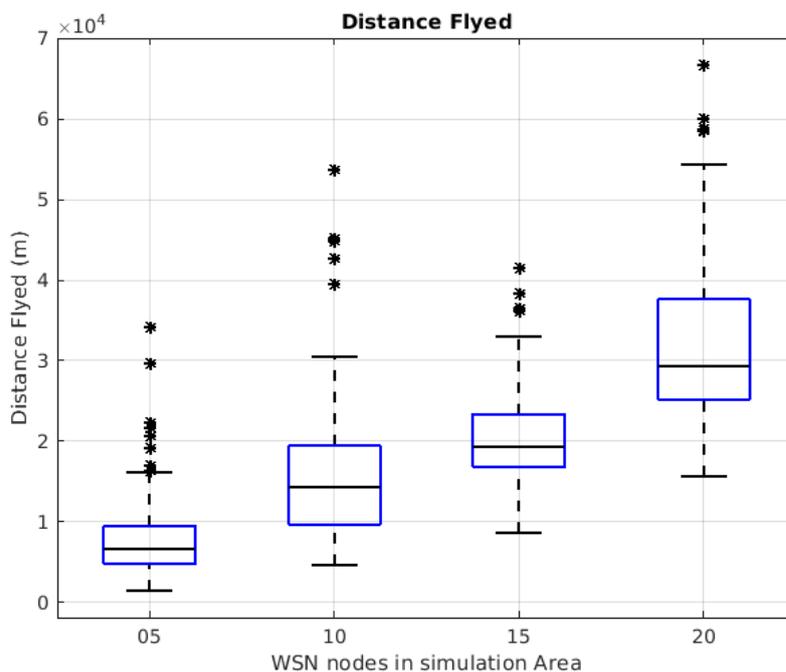


Figura 83 – Distância Percorrida – Cenário 06.

De acordo com a Figura 83, temos a distância total percorrida pelos VANTs em função da quantidade de Sensores no cenário de simulação.

A Figura 84 apresenta o tempo de voo dos VANTs no ambiente de simulação até que todos os sensores sejam descobertos e reprogramados.

A Figura 85 representa a quantidade de saltos realizados pelo VANTs no cenário até a descoberta e reprogramação de todos os sensores.

Na Figura 86, temos a quantidade total de *timeslots* contabilizados que se passaram até a descoberta e reprogramação de todos os sensores no ambiente de simulação.

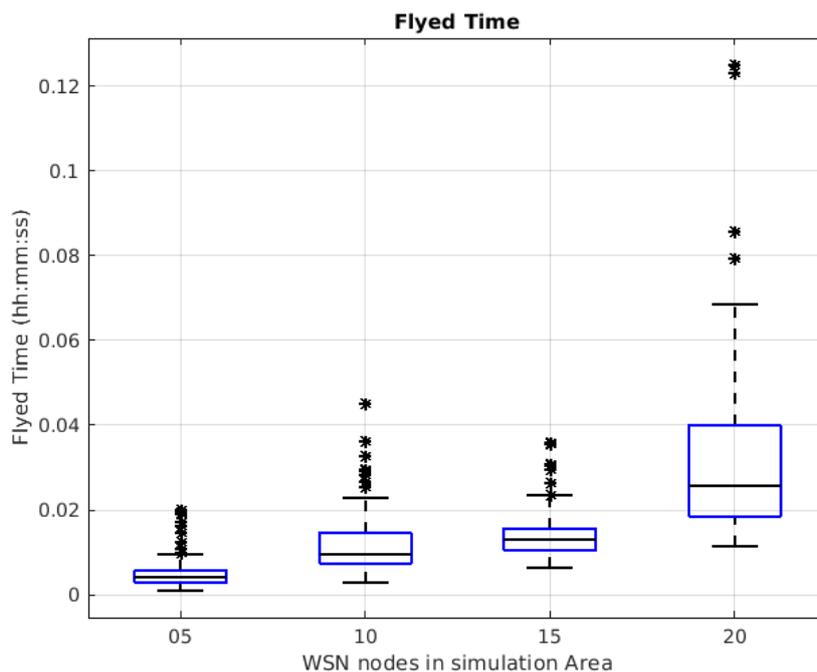


Figura 84 – Tempo de Voo – Cenário 06.

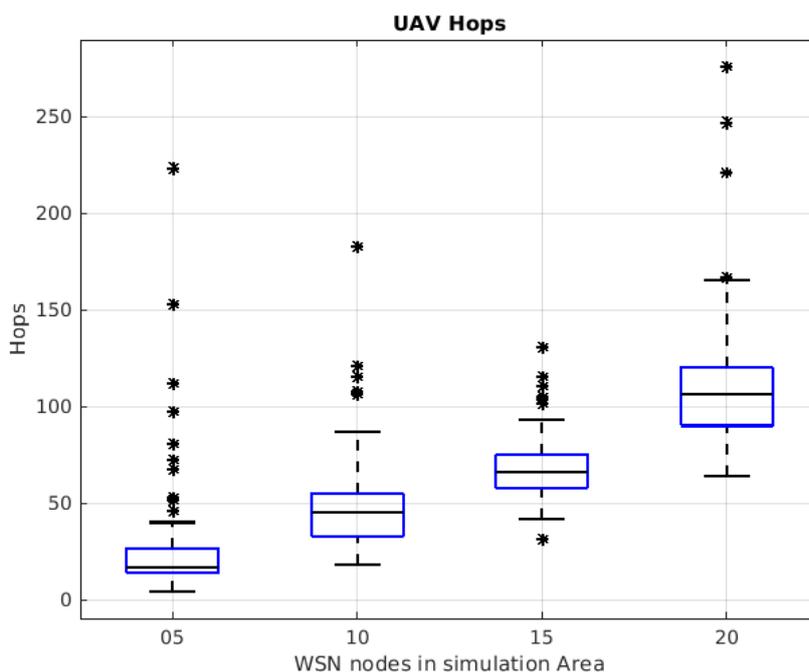


Figura 85 – Quantidade de Saltos – Cenário 06.

**Cenário 07:** Para a sétima simulação, utilizaremos o primeiro cenário como referência, porém as Estações de Coleta estarão transmitindo e disponíveis para identificação por apenas um timeslot, ao invés de dois como os demais cenários.

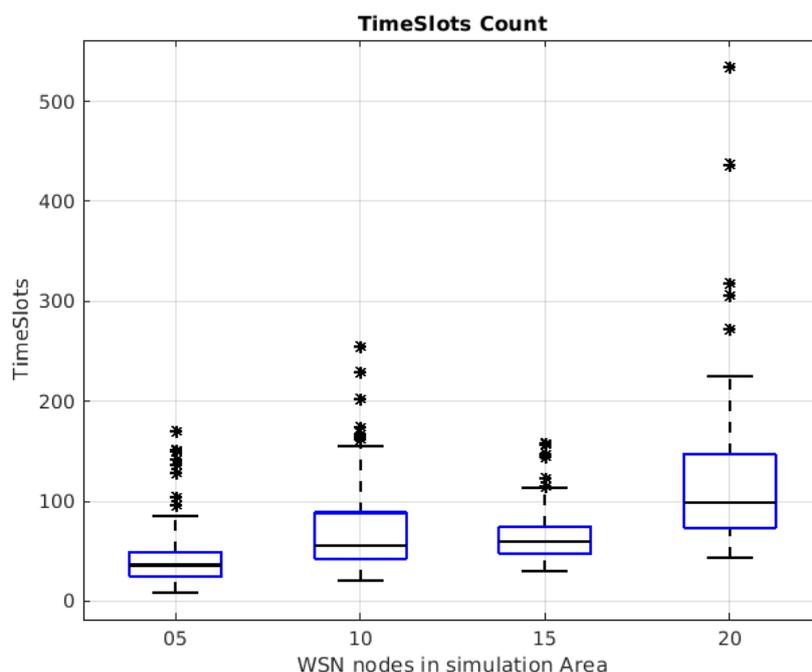


Figura 86 – Quantidade de Timeslots – Cenário 06.

- Velocidade média do VANT: 25m/s
- Planejamento de Percurso: NearestPoint
- Raio de Comunicação das ECs: 100m
- Área de Simulação: 1212m x 517m, aproximadamente  $700.000m^2$
- Quantidade de ECs: 5, 10, 15 e 20 estações
- Tempo de Broadcast das ECs: 1 *timeslots*
- Autonomia do VANT: 10km
- Sensores em timeslots individuais
- Sistema de aprendizado ativado

Na Figura 87 é apresentada a quantidade de VANTs utilizados para identificar e reprogramar todas as ECs.

De acordo com a Figura 88, temos a distância total percorrida pelos VANTs em função da quantidade de Sensores no cenário de simulação.

A Figura 89 apresenta o tempo de voo dos VANTs no ambiente de simulação até que todos os sensores sejam descobertos e reprogramados.

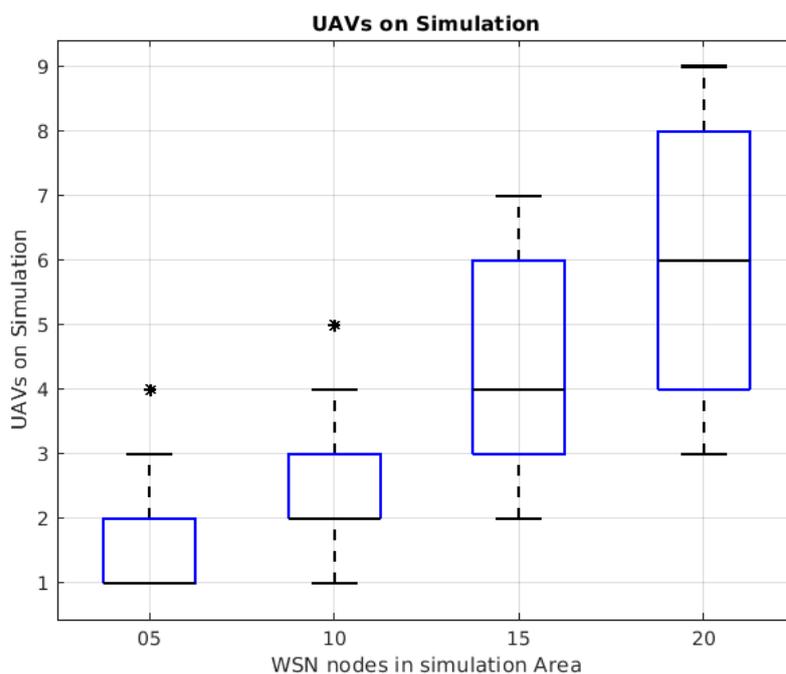


Figura 87 – Quantidade de VANTs – Cenário 07.

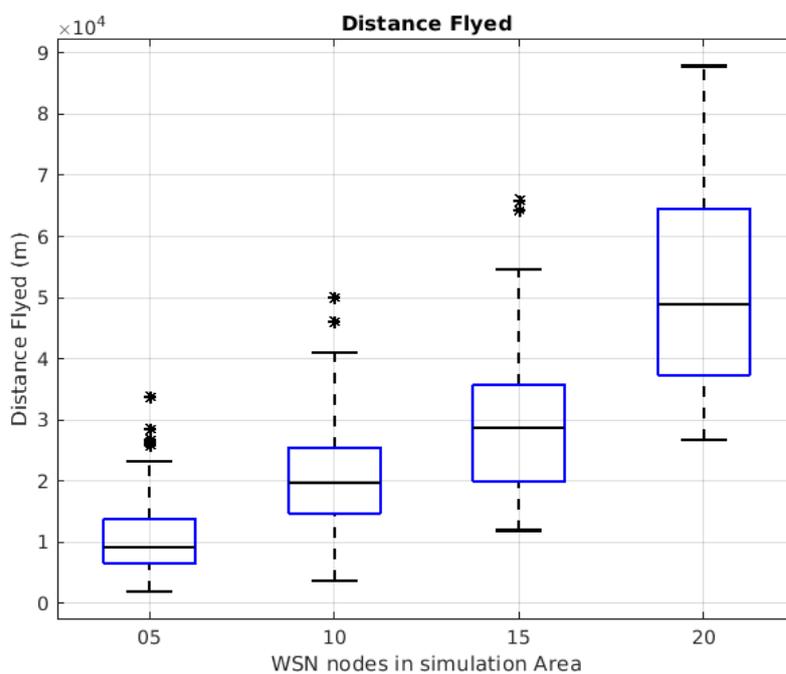


Figura 88 – Distância Percorrida – Cenário 07.

A Figura 90 representa a quantidade de saltos realizados pelo VANTs no cenário até a descoberta e reprogramação de todos os sensores.

Na Figura 91, temos a apresentação da quantidade de decisões tomadas pelos

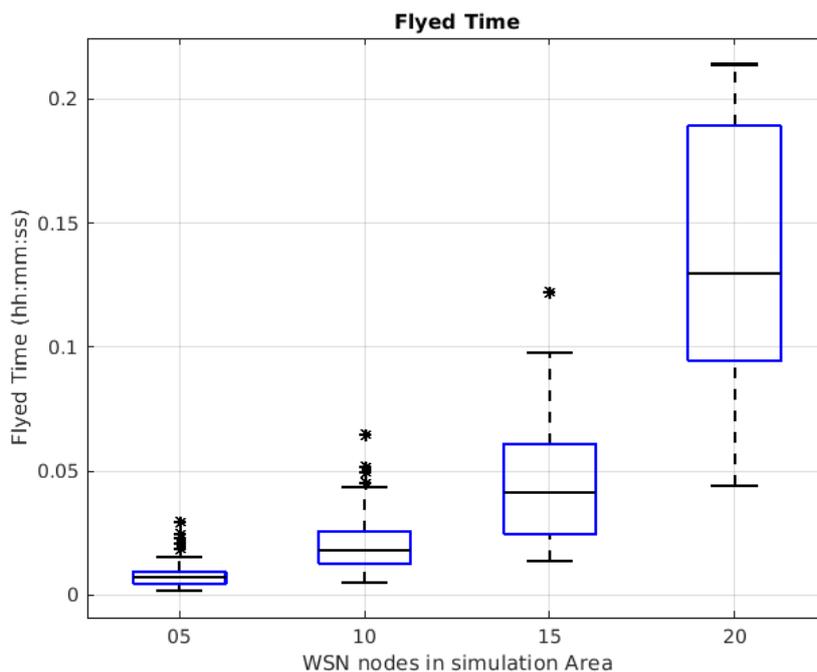


Figura 89 – Tempo de Voo – Cenário 07.

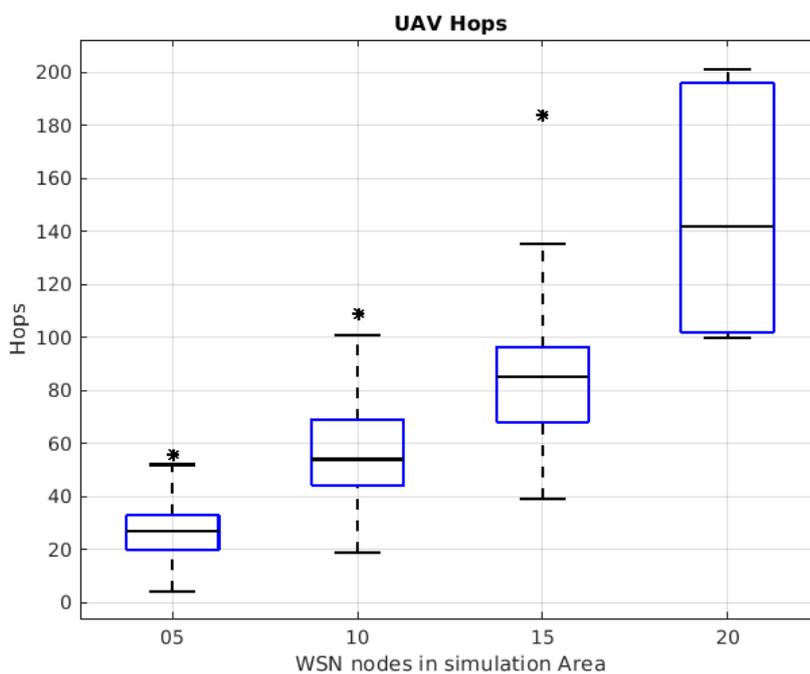


Figura 90 – Quantidade de Saltos – Cenário 07

VANTs durante a fase de descoberta e reprogramação dos sensores.

Na Figura 92, temos a quantidade total de *timeslots* contabilizados que se passaram até a descoberta e reprogramação de todos os sensores no ambiente de simulação.

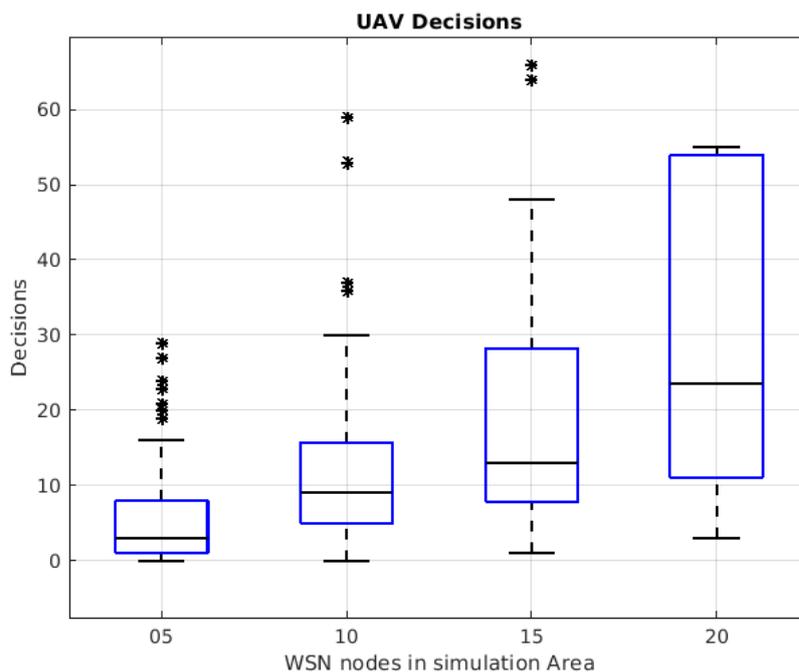


Figura 91 – Quantidade de Decisões - Cenário 07

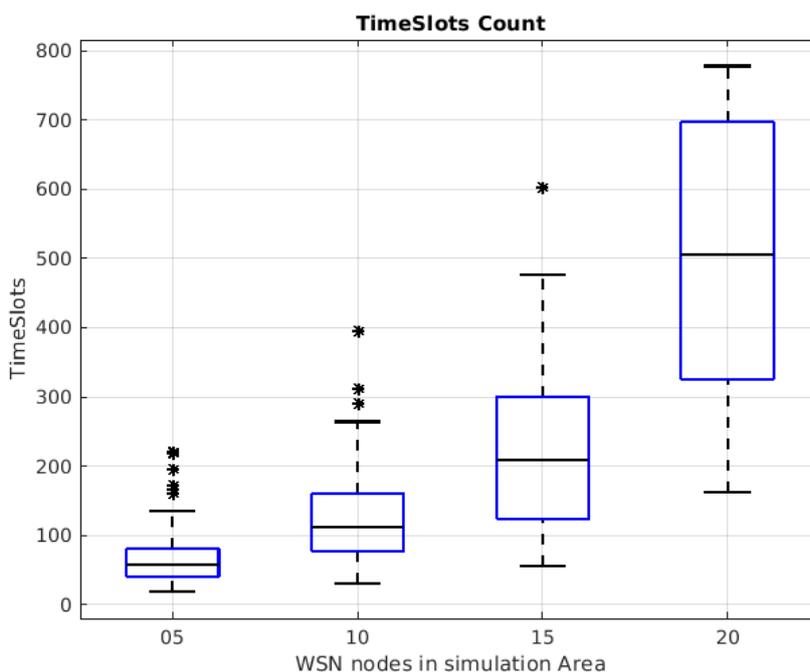


Figura 92 – Quantidade de Timeslots – Cenário 07

## 6.2 Resultados Comparativos

Comparação entre Resultados - Quantidade de VANTs utilizados nas simulações.

O gráfico abaixo compara graficamente a quantidade média de VANTs utilizados para as simulações em cada cenário.

A Figura 93 apresenta os resultados comparativos da quantidade de VANTs utilizados em cada cenário.

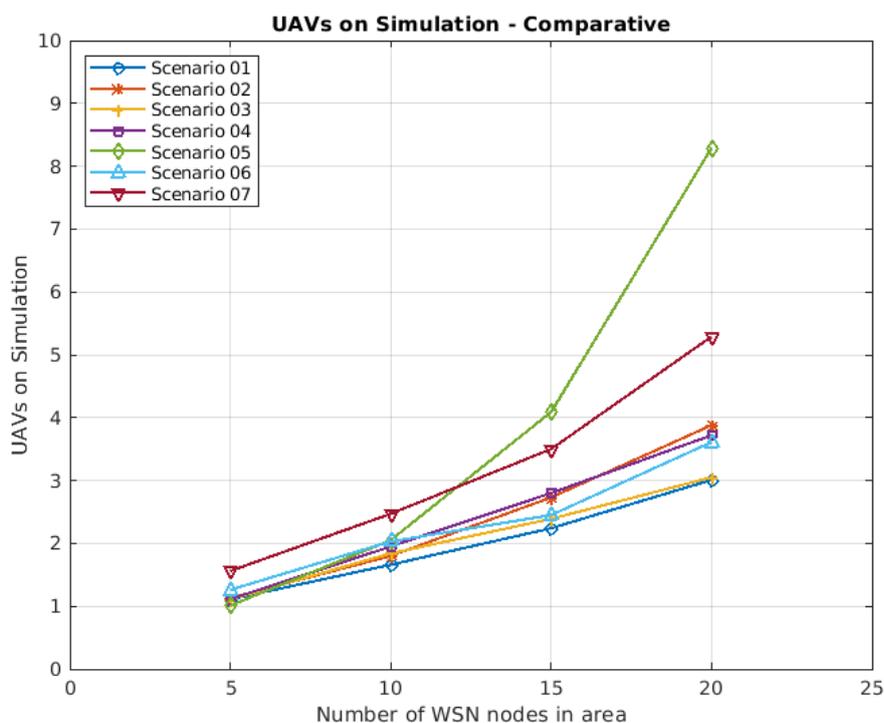


Figura 93 – Comparativo entre Quantidade de VANTs.

A Figura 94 apresenta os resultados comparativos para a média das distâncias percorridas pelos VANTs em cada cenário.

A Figura 95, apresenta os resultados comparativos para a média do tempo de voo pelos VANTs em cada cenário.

Na Figura 96, apresentamos o tempo médio de voo para os VANTs para cada cenário.

Na Figura 97, é apresentada a média da quantidade de decisões tomadas pelos VANTs para cada cenário.

Na Figura 98, é apresentado o comparativo entre a quantidade de tomadas de decisões por parte dos VANTs, para cada cenário.

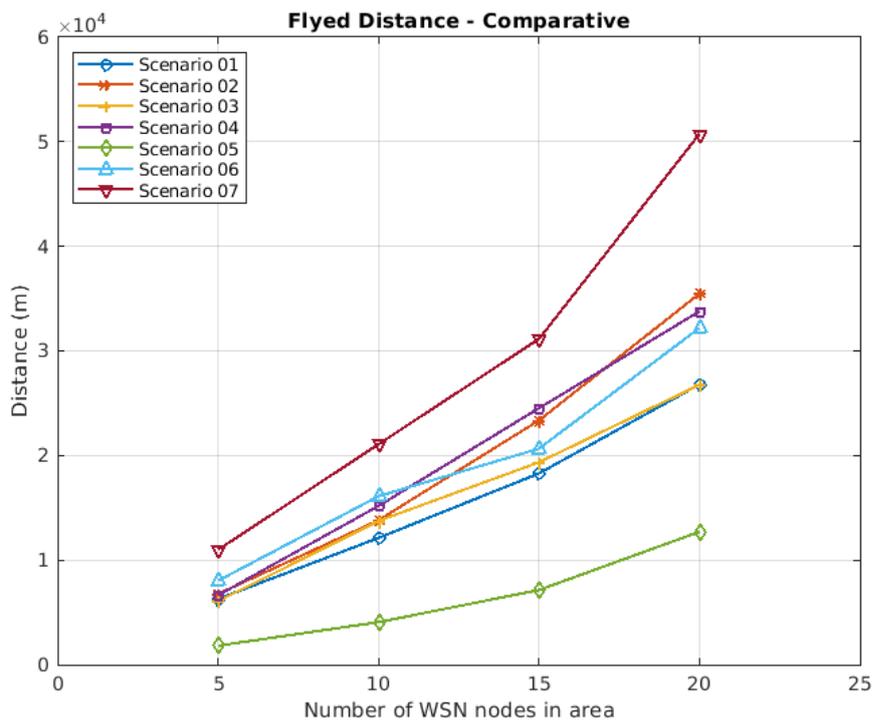


Figura 94 – Comparativo entre as Distância Percorrida pelos VANTs.

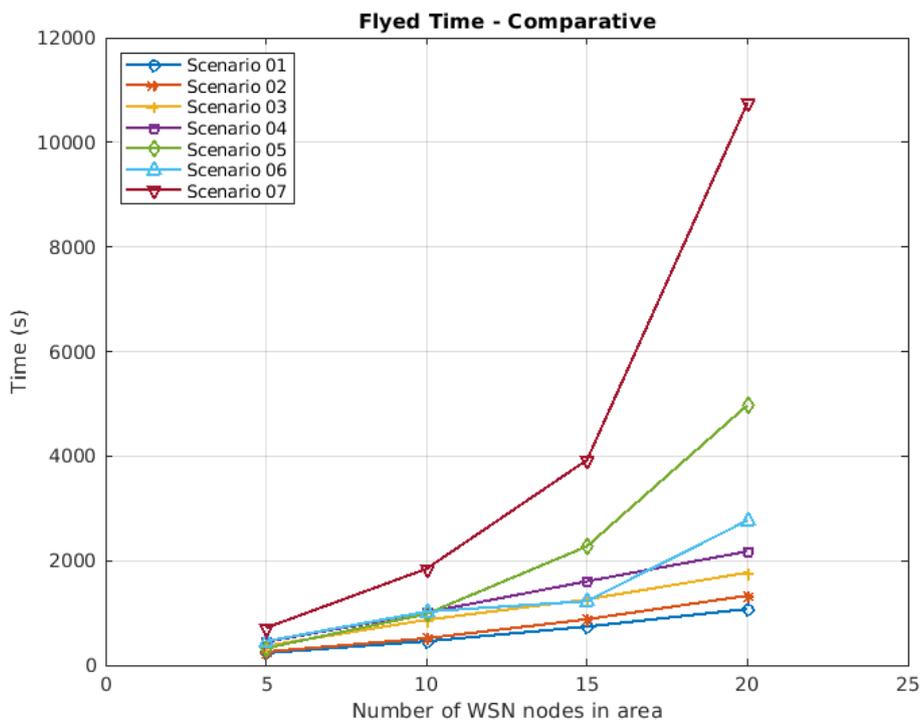


Figura 95 – Comparativo entre Tempo de Voo dos VANTs.

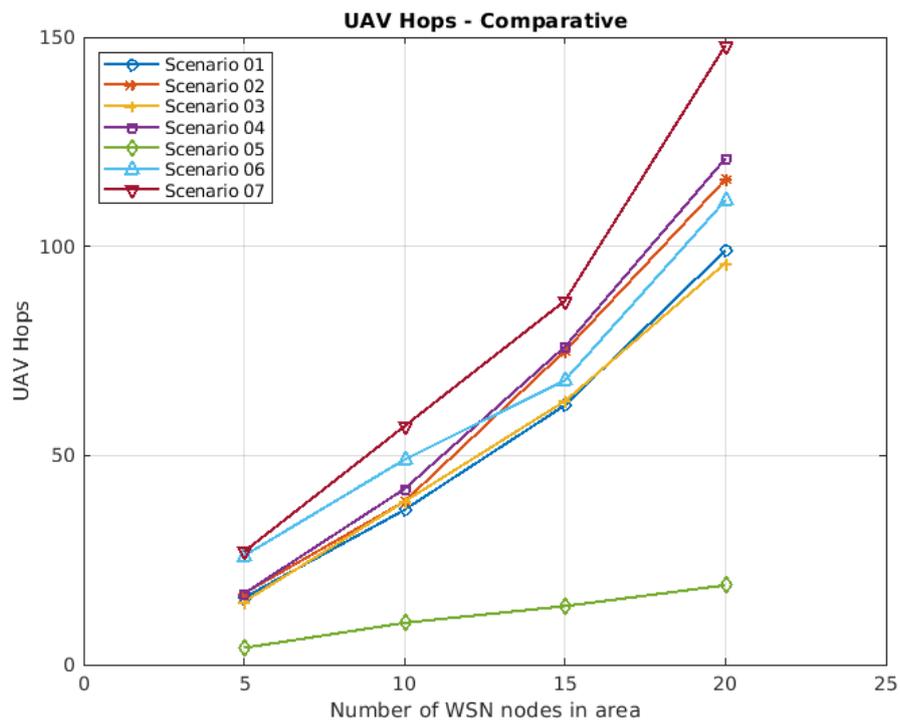


Figura 96 – Comparativo entre Quantidade de Saltos dos VANTs.

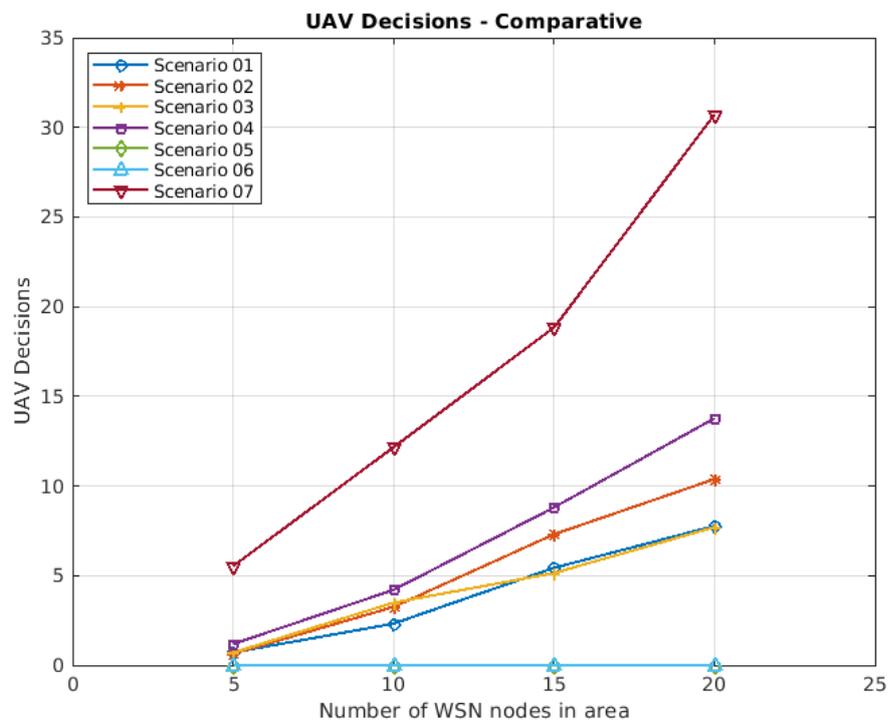


Figura 97 – Comparativo entre Tomadas de Decisões pelos VANTs.

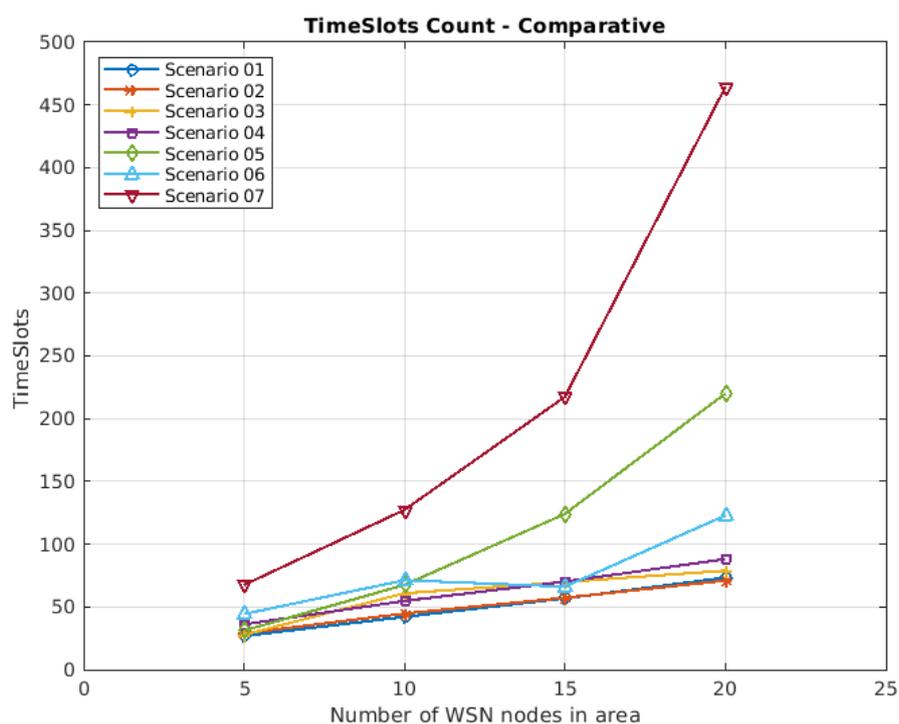


Figura 98 – Comparativo entre Quantidade de TimeSlots.

### 6.3 Discussão dos Resultados

De acordo com a primeira simulação, com velocidade constante do VANT de 25m/s, utilizando o percurso *NearestPoint* como rota padrão para visitar as ECs, sendo que estas não compartilham *timeslots* para sua fase de transmissão, temos um padrão para a quantidade de VANTs. De acordo com a Figura 49, para 5 ECs, temos ocupação média de apenas um VANT, para 10 ECs temos ocupação média de um à dois VANTs em 75% do casos, chegando até 3 VANTs para o restante dos cenários (25%). Para 15 ECs temos ocupação média de dois a três VANTs para 75% dos casos, e quando trabalhamos com 20 ECs, temos ocupação média de 3 VANTs para praticamente 100% dos cenários, com exceção, de alguns pontos fora da curva, onde foram necessários 2 e 4 VANTs para finalizar a simulação.

Ao realizarmos uma análise sobre os motivos em que ocorreram maior ocupação de VANTs para identificação dos cenários, maior distância percorrida, ou mesmo tempo de ocupação dados pelas Figura 49, Figura 51 e Figura 54, foi possível identificar os motivos dos pontos “fora da curva” que se sobressaíram em relação à média dos dados médios obtidos. De acordo com Figura 99 e Figura 100 podemos observar este padrão de voo, onde o VANT se move constantemente entre uma célula e outra, porém não é capaz de encontra-las em modo de transmissão.

Ainda de acordo com a Figura 100, temos os instantes de conexão do VANT com

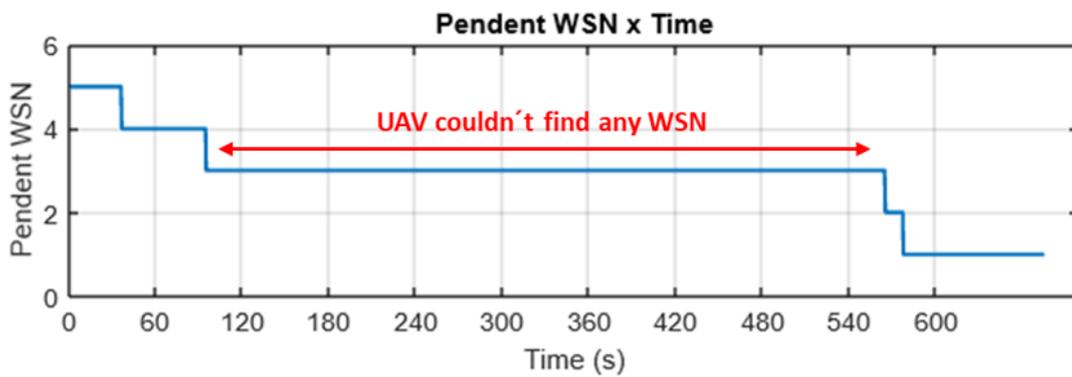


Figura 99 – Tempo de voo em relação à sensores não descobertos pelo VANT.

as ECs em função do tempo.

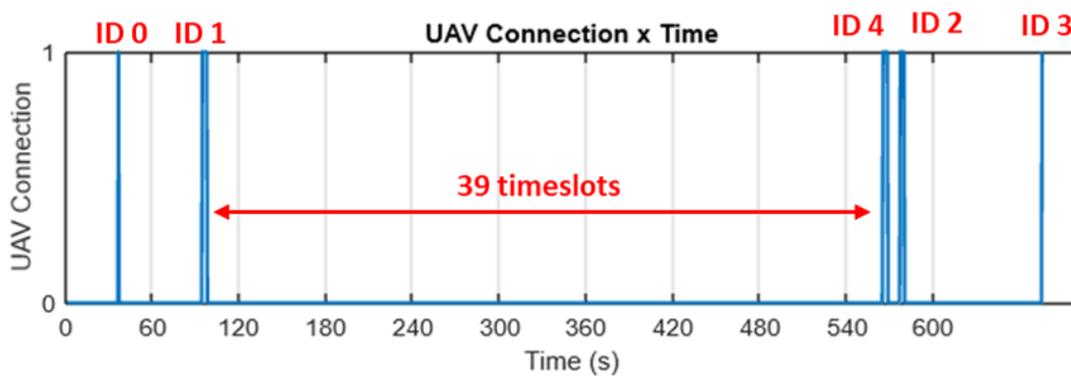


Figura 100 – Conexões do VANT com Estações de Coleta em função do tempo.

Sobre o eixo vertical, temos a conexão com a EC por parte do VANT, sendo assim podemos perceber que no primeiro minuto de simulação, o VANT identificou as ECs cujo ID é 0, durante *timeslot* 2 e ID 1 respectivamente no *timeslot* 7, de acordo com os *logs*. Em seguida, ficou sobrevoando as ECs restantes cujo IDs são 2, 4 e 3 até que finalmente encontrou a EC com ID 4 depois de quase 8 minutos de voo (*timeslot* 46), finalmente encontrou as EC 2 e 3 em seus próprios *timeslots*.

À medida em que o VANT se desloca entre uma estação de coleta e outra, ele às adiciona em suas listas de permissão e restrição, conforme representado pela Figura 48. Portanto, com base em seu banco de dados, o VANT em algum momento tomará uma decisão alterando o padrão de voo da sua rota padrão, mudando assim a dinâmica de sua passagem entre as estações.

Para este cenário, considerando que temos 5 ECs na área de simulação, sendo que para cada uma existe a possibilidade de estar transmitindo em um dos 13 *timeslots*

determinados para o cenário (sem repetição), existem um total de 65 (5x13) possíveis combinações do VANT localizar uma estação em modo de transmissão. À cada passagem do VANT sobre uma estação de coleta, ele vai retirando aquela estação, daquele *timeslot* de sua lista, reduzindo as possibilidades com o tempo. De acordo com a Figura 101, podemos analisar o momento em que o VANT começou a utilizar informações conhecidas de seu banco de dados para tomada de decisão sobre ir ou não ir para determinada estação, sobre aquele determinado *timeslot*. Esse processo de tomada de decisão apenas ocorre quando o VANT já possui conhecimento sobre o cenário sobrevoado.

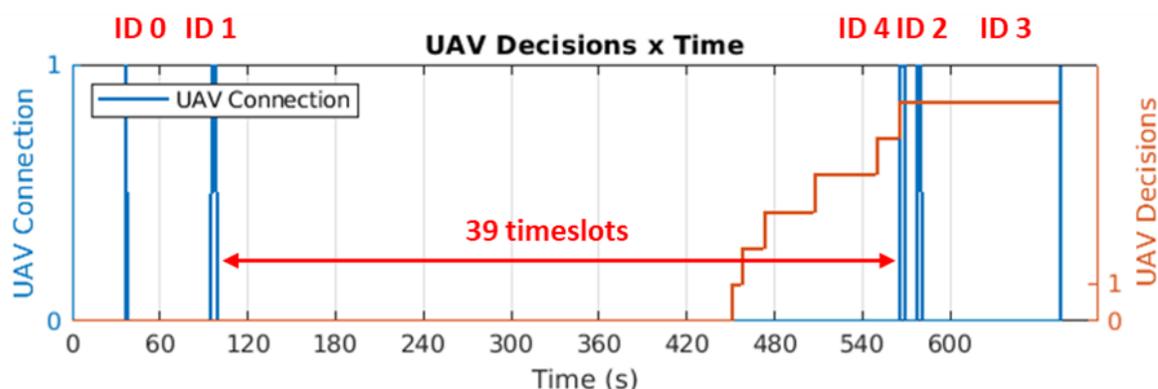


Figura 101 – Decisões tomadas pelo VANT.

De acordo com os *logs* gerados para essa simulação, o VANT possui 65 combinações diferentes em seu banco de dados interno, quando o VANT encontrou o 2º sensor (ID 1) no *timeslot* 7, já haviam reduzidos 2 estações de sua lista (cada estação comporta 13 *timeslots*), portanto, as probabilidades haviam reduzido 26 de 65 possibilidades, restando 39 diferentes possibilidades (*timeslots* x sensores restantes). Ou seja, desde o encontro do VANT com sensor ID 1 ocorrido no *timeslot* 7, e todas as passagens do VANT ocorridas até que finalmente descobrisse a EC 4 passaram-se 39 *timeslots*, ocorrendo assim o pior cenário possível. Ainda de acordo com a Figura 100, o tempo restante entre as EC cujo ID 2 e 3 é devido ao VANT ter encontrado ID 2 no *timeslot* 0, e em seguida dirigiu-se para a EC ID 3 cujo *timeslot* estava programado para o 4, portanto houve um tempo de aproximadamente 2,5min de aguardo sobre ela para localizá-la e reprogramá-la.

Ao realizarmos a análise, ainda sobre a Figura 102, analisando o cenário com 20 Estações de Coleta, considerando um dos melhores e menores tempos para identificação de todas as ECs, temos uma probabilidade muito menor do VANT tomar uma decisão de início, devido à quantidade de estação e *timeslots*. Ao analisarmos a probabilidade de o VANT tomar decisões, temos 20 ECs, e em média 22 *timeslots* para cenários com esta configuração de ECs, portanto, nos resulta em 440 possibilidades diferentes entre *timeslots* e ECs. Conforme o VANT vai encontrando e programando as EC, o drone vai eliminando em sua base de dados aquele *timeslot*, juntamente com todas as possibilidades de encontrar

o mesmo sensor em quaisquer outros *timeslots*. Como no cenário apresentado pelas imagens anteriormente analisadas, maior quantidade de decisões tomadas não significa que o VANT já percorreu ECs suficientes em *timeslots* repetitivos para começar a tomada de decisões baseados em histórico de voo tais estações de coleta.

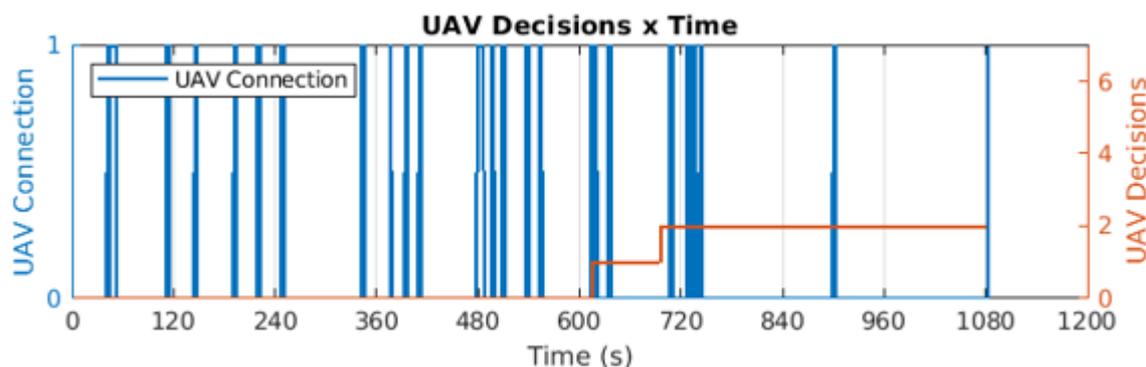


Figura 102 – Análise com 20 timeslots.

Levando em consideração que o tempo médio do cenário com 20 estações de coleta, a tomada da primeira decisão costuma ocorrer no tempo médio 11 minutos de 30 segundos, de acordo com a Figura 103, o VANT estava dentro da média, ou seja, apesar do primeiro VANT ter descoberto 8 sensores de um total de 20, passou a informações referente suas próprias listas de permissão/restrrição, e timeslots/ECs já conhecidos para o segundo VANT que ao descobrir quatro sensores, à partir do *timeslot* 21 de acordo com os *logs* do simulador referente ao cenário em análise, localizou os 12 sensores restantes em apenas 25 *timeslots*.

Além disso, como podemos analisar na Figura 104, temos a distância percorrida pelo VANT até que encontre todos os sensores pendentes na área em questão. Através do gráfico da distância percorrida, podemos ver o momento em que o VANT inicial atinge sua capacidade de autonomia de 10km, ocorrendo sua substituição no *timeslot* 21, de acordo com os logs gerados para este cenário, a partir deste *timeslot*, o segundo VANT assume a pendência de 12 sensores em campo ainda a serem descobertos. A busca por estas EC terminam no *timeslot* 45, ou sejam os 20 sensores foram todos descobertos com uma taxa de aproximadamente 2 *timeslots*/sensor, no melhor dos cenários para 20 Estações de Coleta.

Ao analisarmos um cenário com 10 estações de coleta, porém agora pelo ponto de vista de percurso do VANT, temos algumas considerações. Na Figura 105, o percurso do VANT desde o início da simulação, até o momento em que o VANT localiza todos os dez sensores em questão. Ao traçarmos a trajetória do VANT como um todo, não apenas sua rota padrão, mas suas modificações de rota em função dos sensores conhecidos, o VANT vai aos poucos os removendo de sua rota padrão, criando assim uma nova rota à cada EC

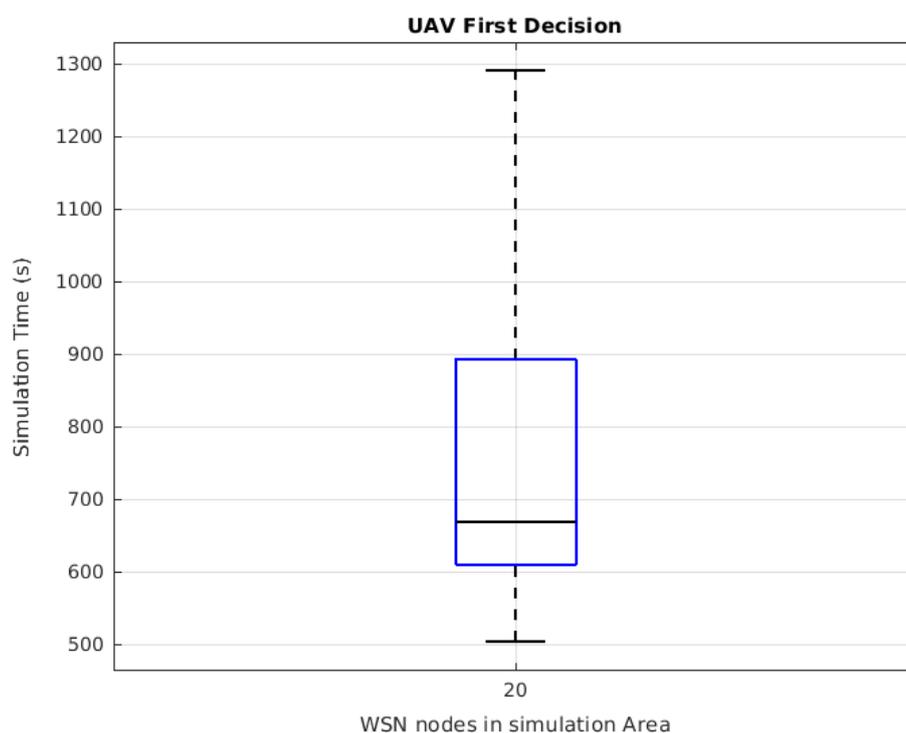


Figura 103 – Tempo médio para primeira tomada de decisão do UAV.

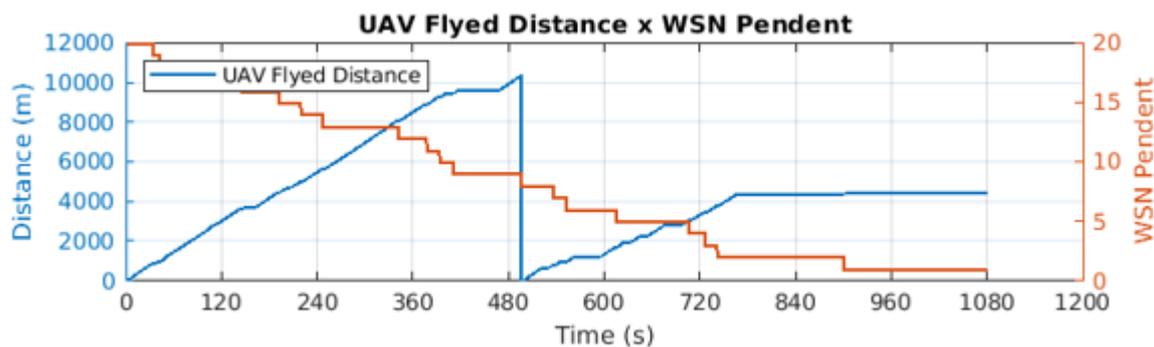


Figura 104 – Distância percorrida e Estações de Coleta Pendentes.

encontrada, portanto temos uma variação do quanto o VANT deveria percorrer em relação ao que efetivamente deveria ter sido percorrido.

Na Figura 105, os pontos em preto são referente à trajetória percorrida pelo VANT em função do tempo, e os círculos em verde, em destaque na mesma figura são referentes aos pontos onde foi identificada conexão entre VANT e a estação de coleta.

Na Figura 106, temos uma relação entre a média de distância percorrida pelo VANT, e sua rota padrão pré-calculada entre todos as estações de coleta.

Essa relação nos dá um parecer do esforço que os VANTs realizam para identificar todos os sensores espalhados em uma área em questão, em função do quanto seria caso

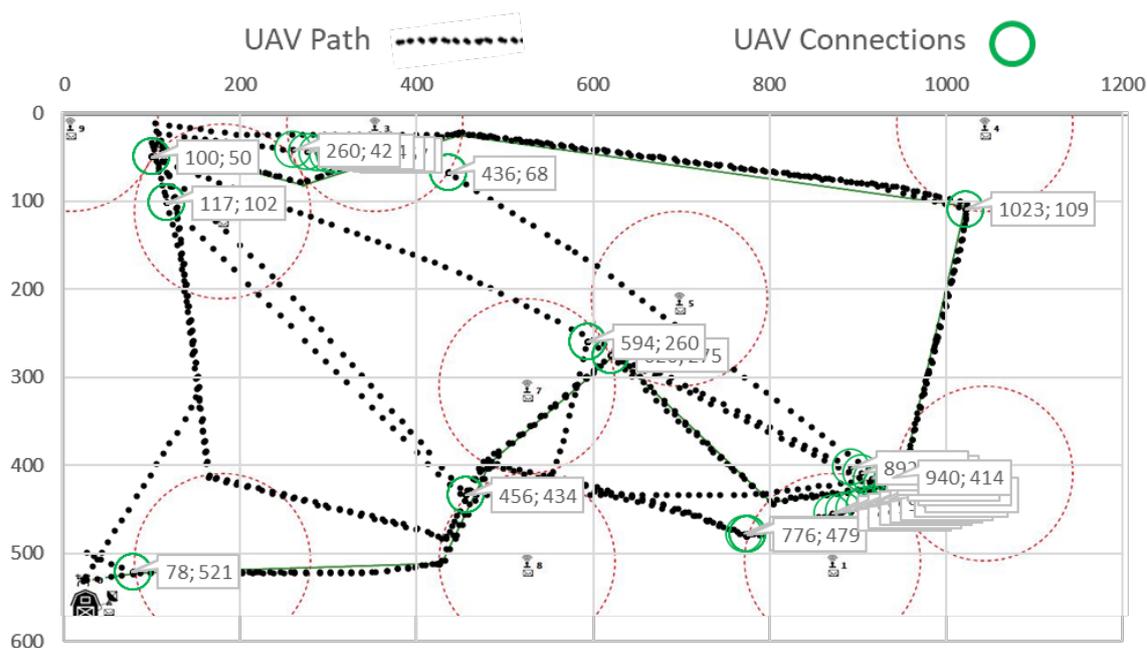


Figura 105 – Percurso do VANT, em função dos pontos de encontro das ECs.

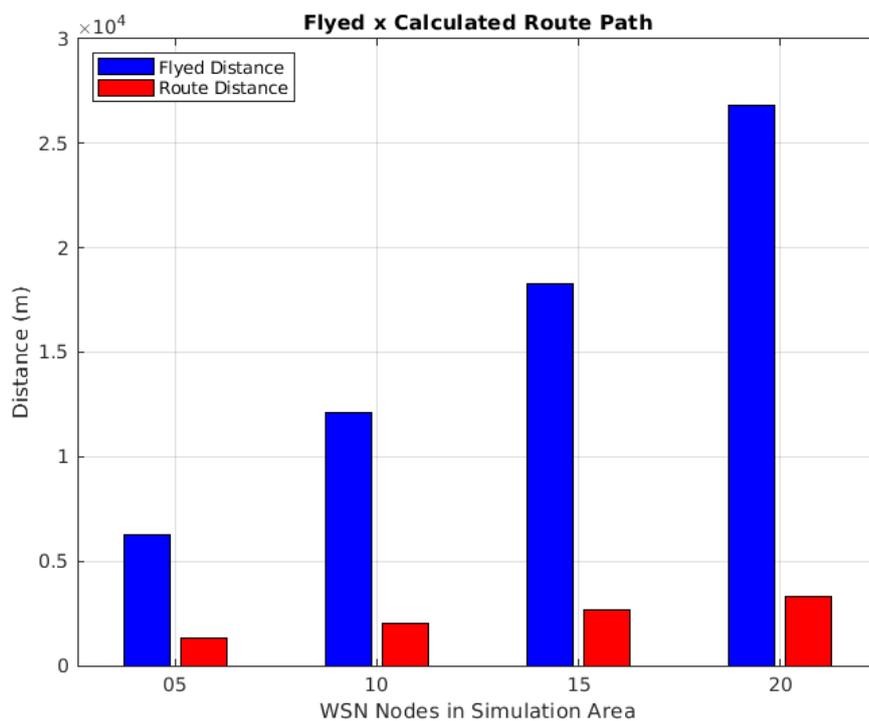


Figura 106 – Diferença de percurso entre caminho planejado e sobrevoado.

eles já estivessem identificados e programados. Portanto, de acordo com a Figura 106 os VANTs em média percorrem 12km para identificar e reprogramar todas as dez estações de coleta, sendo que se estas já estivessem programadas de forma coerente com sua trajetória,

em média levariam 2km, portanto é uma relação de quase seis vezes mais o que o VANT efetivamente deve percorrer para ter o cenário organizado e programado de acordo com sua rota.

Ainda de acordo com a Figura 106, é ainda mais extrema se analisarmos os resultados para vinte estações de coleta em sua área, acompanhando o eixo secundário da imagem em questão, podemos ver que o mínimo que o VANT percorre para cinco ECs seriam aproximadamente 480% de diferença em relação à sua rota original pré-programada.

Acompanhando o gráfico ainda, temos a relação entre as distâncias percorridas pelo VANT e pré-programadas em sua rota, e podemos identificar que a curva que relaciona as distâncias segue um padrão linear, e levando em consideração a autonomia, seja em distância ou mesmo tempo de voo, isso impacta consideravelmente na quantidade de VANTs necessários para percorrer o caminho inicial e identificar as estações de coleta em seu modo inicial, desprogramadas em relação ao percurso do VANT.

Na Figura 105 temos a relação do percurso do VANT apresentado juntamente com a distribuição e área de propagação das estações de coleta. Como podemos observar, o VANT desloca-se continuamente entre uma estação e outra, até que todas sejam localizadas (no tempo/timeslot) e posição. Os *logs* do sistema, apresentam por simulação a posição exata de onde ocorreu o ponto de localização entre VANT e estação de coleta, além de registrar outros dados como a ordem das identificações, ID de conexão do VANT com a EC, momento em que ocorreram conexões, quantidade de vezes em que as ECs estiveram em modo de *broadcast* e hibernação, momentos em que houve transmissão por parte da EC, mas não houve conexão com VANT, , além de outras informações analíticas referente ao cenário em análise.

Através da Figura 107 podemos comparar facilmente a diferença entre a rota original definida pelo VANT, juntamente com o percurso efetivamente percorrido, apresentado na Figura 105.

Em relação ao tempo dos *timeslots*, temos algumas variações, pois no momento em que o VANT inicia seu percurso, ele realiza o cálculo de quantos *timeslots* serão necessários para cobrir seu percurso, sempre levando em consideração o tempo necessário para completar a rota, além do tempo necessário de parada em cada sensor para que seja realizada a leitura dos dados coletados (tempo de leitura), além é claro, do tempo para retornar do último sensor da sua lista, para seu ponto inicial de partida.

Sendo assim, o VANT realiza o cálculo do tempo total, e os divide por uma relação dada, por exemplo 10, 20, 50 ou 100 *timeslots*. Com isso, haverá a divisão do tempo total por este dado número de *timeslots*, resultando no tempo de cada *timeslot* (em segundos). O VANT ainda pode atribuir alguns *timeslots* adicionais ao seu percurso em função de distâncias maiores, com o intuito de ganhar tempo para e sempre chegar à estação de

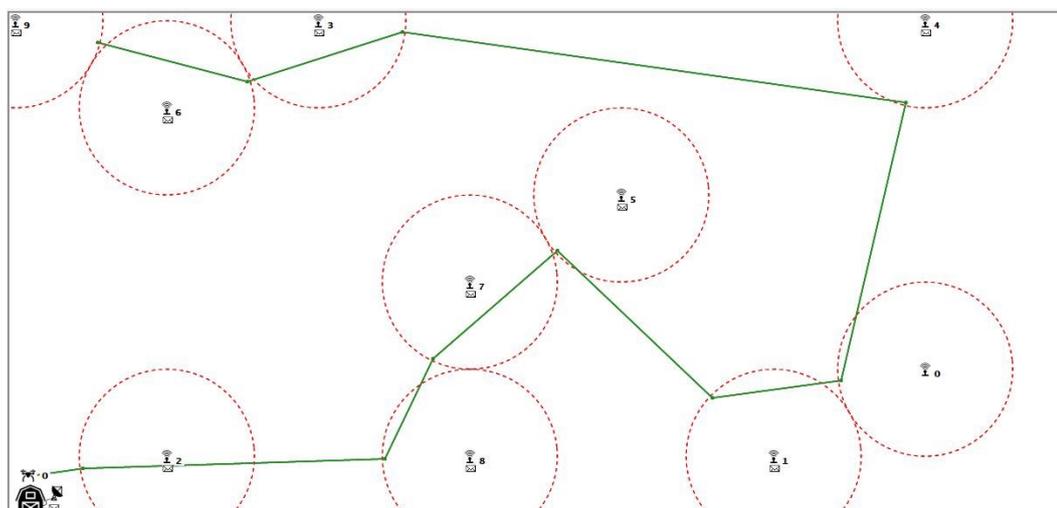


Figura 107 – Rota original planejada antes do percurso.

coleta antes que a mesma já esteja em modo de transmissão, evitando assim, após todas as reprogramações do cenário em questão, que haja transmissão por parte da estação de coleta, sem que haja um VANT já localizado sobre esta, assim evitando consumo de energia desnecessário por parte da EC, maximizando a vida útil das estações.

Como apresentado na Figura 108, existe uma pequena variação da quantidade de *timeslots* média para cada cenário, sendo assim, sendo que este varia de acordo com a distância total do percurso, velocidade média do VANT e quantidade de *timeslots* definidos pelo usuário através da *interface* do simulador, afinal as estações possuem suas posições sorteadas de forma aleatória à cada nova simulação. Essa variação, sempre positiva, ocorre devido ao VANT inserir *timeslots* adicionais em determinados trechos onde necessita de mais tempo para se deslocar de uma EC até outra, e é resultado de sobra propositalmente inserida nos cálculos, por exemplo, se o VANT necessita de 1.6 *timeslots* para se deslocar de um ponto à outro, esse valor é arredondado para 2 *timeslots*, e assim por sucessivamente. Isso gera uma diferença positiva de 0.4 *timeslots* que será somada no próximo cálculo, gerando essa quantidade adicional de *timeslots*. Além disso, o percurso de retorno do último sensor, até o ponto de entrega dos dados é adicionado a este cálculo. Esse adicional é necessário para garantir que o VANT esteja sempre na posição desejada, aguardando a EC entrar em modo de *broadcast*.

Na Figura 109 é apresentado as variações do tempo de cada *timeslot*, em função da quantidade de sensores na área de simulação. Também existe uma pequena variação no tempo de cada *timeslots*, isso se dá devido à variação da distância total percorrida pelo VANT variar de acordo com a posição das estações de coleta, em função dos diferentes possíveis cenários gerados nas simulações para o VANT.

O tempo de *broadcast* de cada estação de coleta, foi definido por padrão para dois *timeslots*, portanto, como o tempo de *broadcast* varia em função do tempo de *timeslot*

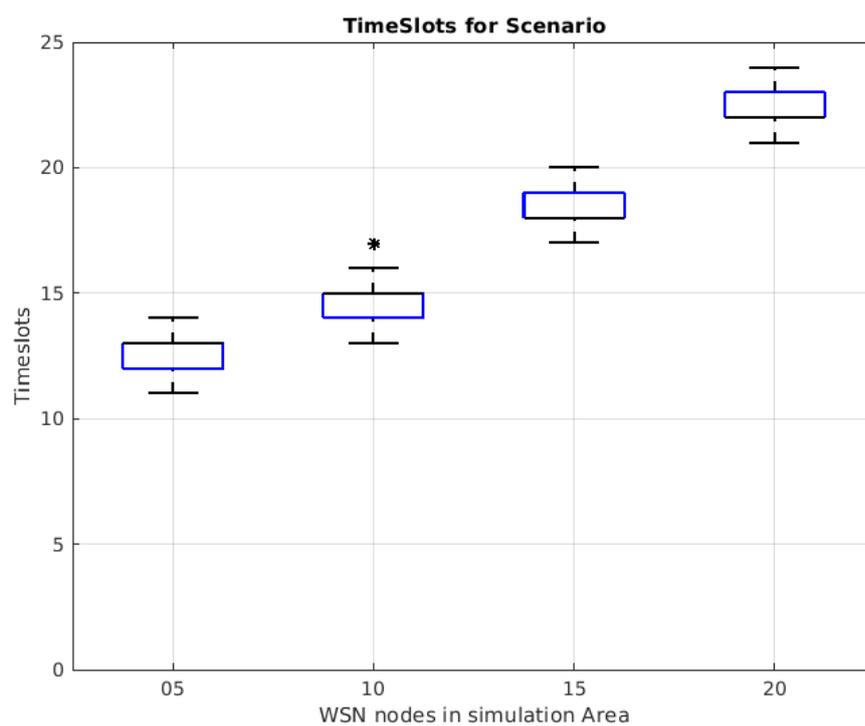


Figura 108 – Quantidade de timeslots por cenário.

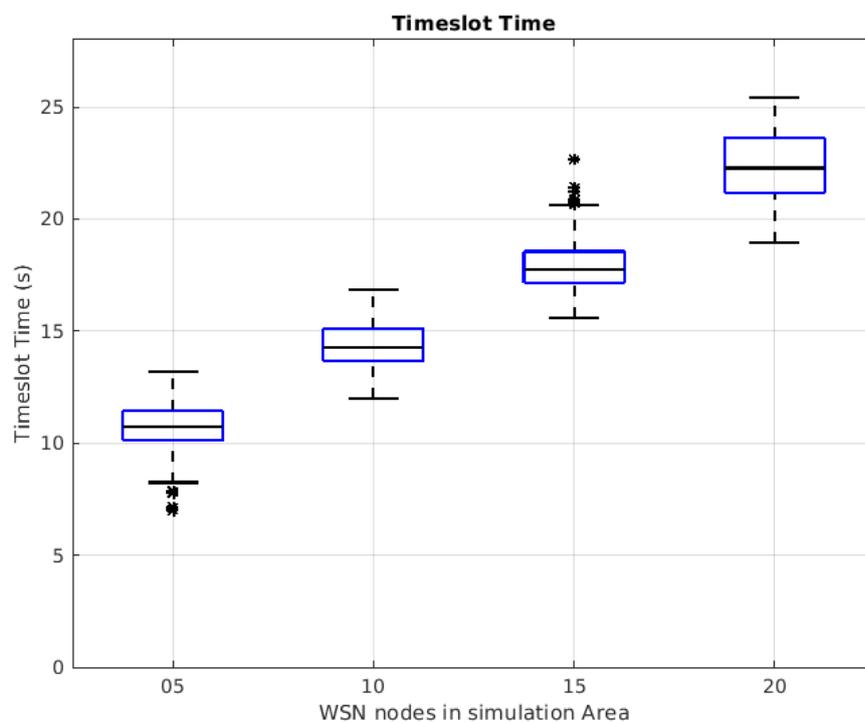


Figura 109 – Variações de Tempo de cada timeslot para quantidade de ECs.

para cada cenário. O tempo de dois *timeslots* para o tempo de *broadcast* foi adotado como padrão para todos os cenários trabalhados, trata-se de um parâmetro ajustável no simulador, portanto a Figura 110 apresenta a variação de tempo de *broadcast* com 5, 10, 15 e 20 estações de coleta.

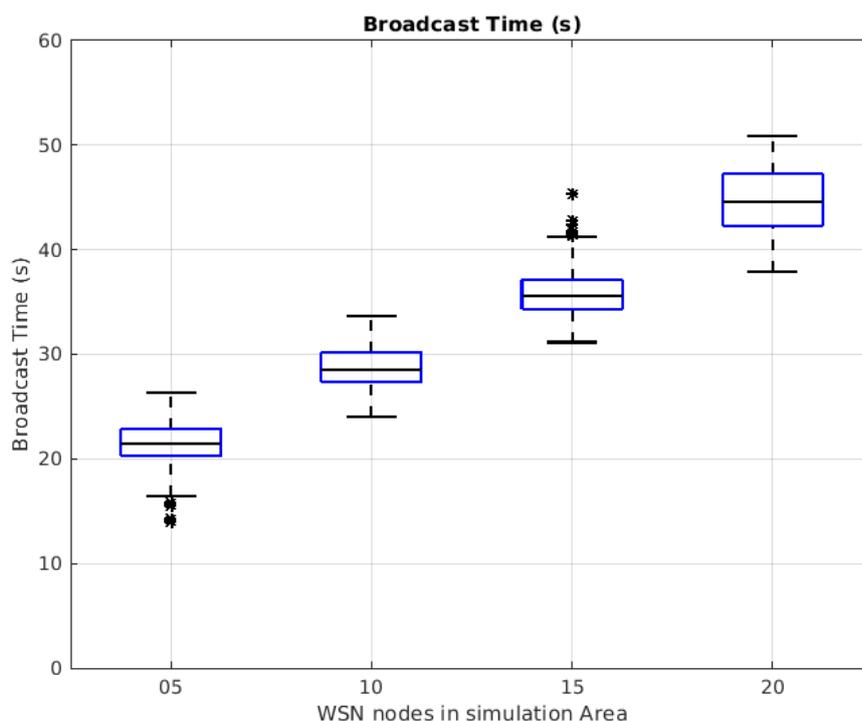


Figura 110 – Variação de tempo de Broadcast para cada cenário.

Como o VANT reprograma as estações de coleta de acordo com sua passagem, e após sua programação o VANT coloca a estação para que ela entre em modo de hibernação por um período, que esteja sincronizado com seu período, ao fazer o VANT chegar antes da EC acorde e posicione-se sobre sua área de cobertura, fazendo com que ela acorde, receba e transmita dos dados necessários, e o VANT a programe em modo de hibernação após a coleta dos dados sensoriais.

Sendo assim, uma vez que o cenário está completamente programado de acordo com o percurso do VANT, temos uma significativa redução do consumo de energia por parte da Estação de Coleta. De acordo com a Figura 111 temos uma média de transmissão de aproximadamente 29s para um cenário com dez sensores. Ao reprogramá-los de acordo com o percurso do VANT, o tempo de transmissão de cada sensor é reduzido para apenas 3s (tempo de leitura, ajustável pelo simulador) em todos os cenários. A Figura 112

apresenta os períodos de ativação e hibernação da estação de coleta, utilizando cenário com 10 sensores. Através do gráfico, podemos ver o período de transmissão da EC de 29s inicial, em seguida 15s em que foi descoberta pelo VANT e colocada para hibernação

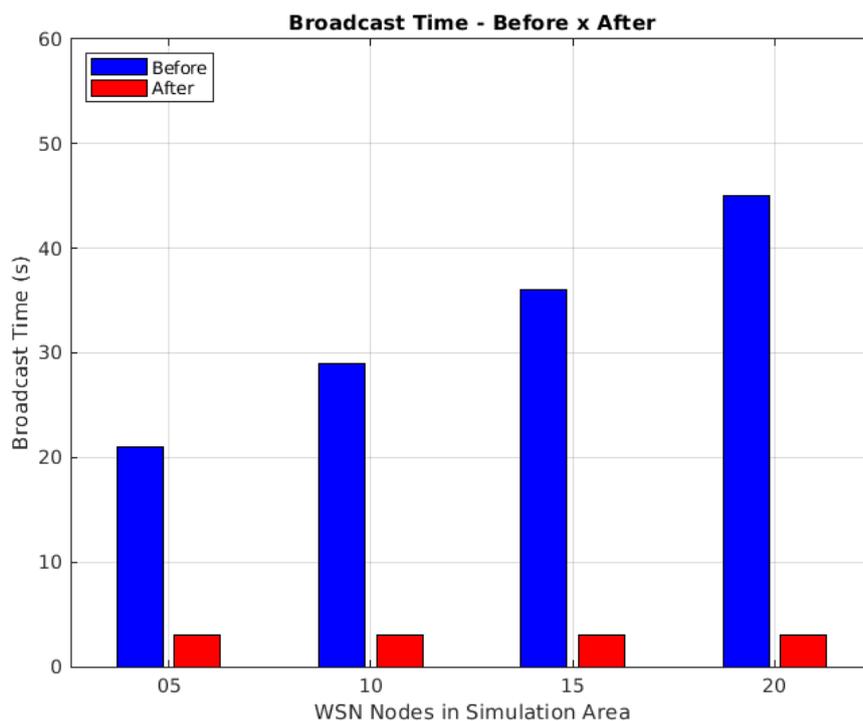


Figura 111 – Tempo médio de Broadcast x Tempo programado por EC.

posteriormente cortando seu ciclo em 14s, e em seguida transmite por mais 29 segundos, afinal o VANT não passa mais por ela após programada. Após a fase de descoberta das ECs, quando as estações pendentes são todas localizadas e reprogramadas, de acordo com o eixo vertical secundário da mesma figura abaixo, o VANT pode se posicionar de acordo com o *timeslot* programado para cada estação, sendo que no momento em que a EC acordará para transmitir/receber, o VANT já estará posicionado sobre ela, e após o tempo de leitura (programado para 3s), a mesma será desligada até seu próximo ciclo de transmissão/recepção, e o VANT partirá para a próxima EC.

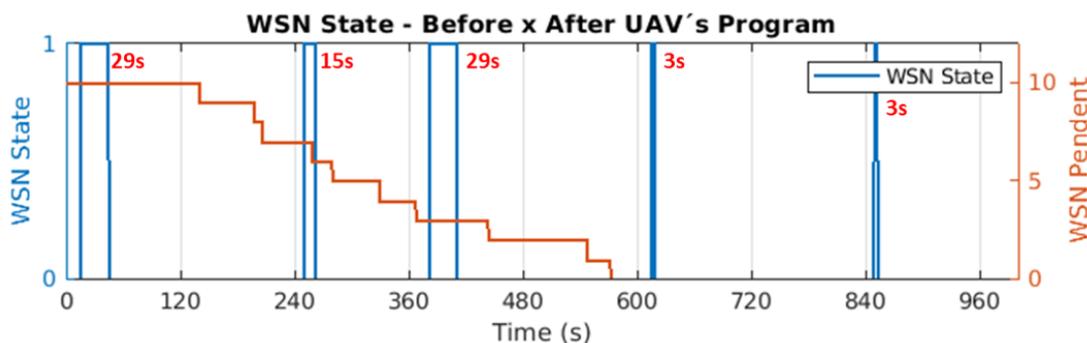


Figura 112 – Tempo de broadcast e Sleep da Estação de Coleta.

O mesmo padrão pode ser visualizado em diferentes ECs (IDs 1, 2 e 3) do mesmo

cenário de simulação, na Figura 113 é apresentadas os demais estados em função do tempo para outras estações de coleta, durante a fase de identificação e reprogramação dos sensores é a fase crítica, que haverá maior consumo de energia por parte das estações, porém, após o período de reprogramação das mesmas, esse valor pode ser reduzido significativamente, de acordo com os dados apresentados anteriormente.

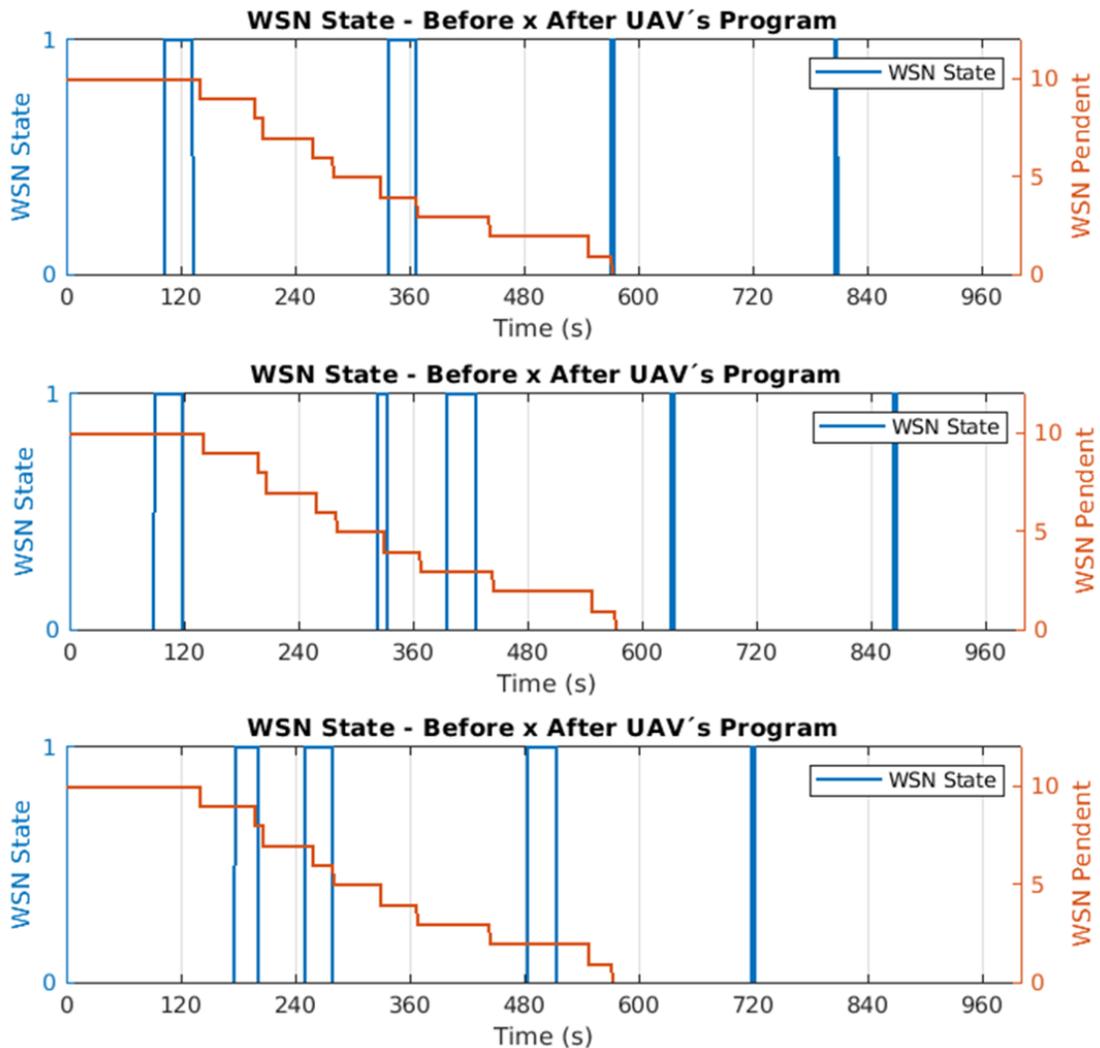


Figura 113 – Ciclo de transmissão e hibernação das ECs.

Além disso, existe uma balança que deve ser analisada, pois o tempo de *broadcast* configurado para os sensores na fase inicial é de dois *timeslots*. Caso optássemos por reduzir para um único *timeslot* o tempo de *broadcast* das estações, por exemplo, a probabilidade de o VANT encontrar e reprogramar a estação de coleta diminui, o que impacta em mais ciclos de transmissão por parte da EC, sem que seja descoberta pelo VANT, ou mesmo em mais VANTs para localizá-las, dada sua autonomia de voo.

Através dos resultados apresentados no sétimo cenário com tempo de *broadcast* de apenas um *timeslot* de *broadcast* para as estações, em comparação com o primeiro

cenário, onde o tempo de *broadcast* é de dois *timeslots*, notamos uma expressiva diferença, na Figura 95, referente ao tempo de voo, com dois *timeslots* a média para 15 ECs é de cerca de 12 minutos, porém ao reduzirmos o tempo do *broadcast* para um único *timeslot*, a média de tempo para reprogramação de todo o cenário sobe consideravelmente para 1 hora e 5 minutos. Além disso, se levarmos em consideração a contagem de *timeslots* total para identificar e reprogramar o cenário, de acordo com a Figura 98 temos uma média 74 *timeslots* para 20 ECs com tempo de *broadcast* de dois *timeslots*, e ao reduzirmos para um único *timeslot*, a contagem de *timeslots* para finalização do cenário sobe para 465 *timeslots*. De acordo com a Figura 114, uma amostra do cenário de simulação 36 para 10 ECs com apenas um único *timeslot* de *broadcast*, temos um prolongamento da quantidade de ativações da EC, até que todas as estações de coleta sejam identificadas e reprogramadas. Lembrando que de acordo com a Figura 110, temos uma média de aproximadamente 14s de tempo de *broadcast* para cenários com 10 estações de coleta.

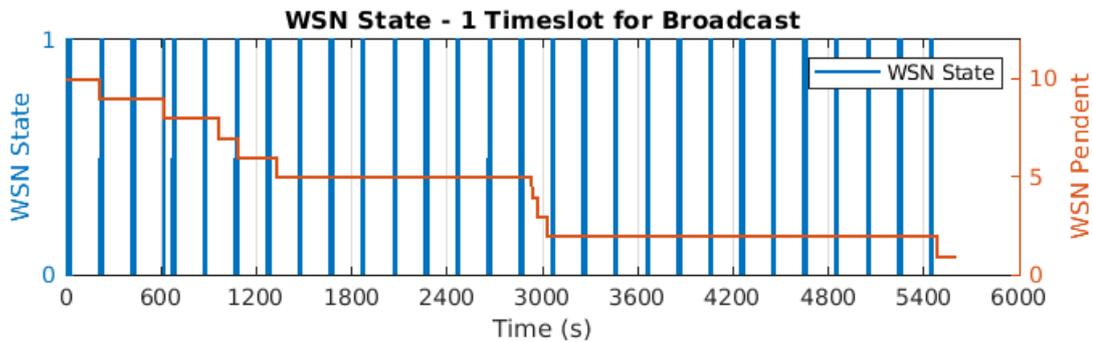


Figura 114 – Estado da EC até que sejam todas encontradas.

Ou seja, de acordo com os resultados obtidos, ao reduzirmos o tempo de transmissão da EC, aumentamos a quantidade de vezes em que ela se ativa para transmitir, prolongando o consumo de energia por parte dela. De acordo com os resultados obtidos quanto mais tempo a EC estiver transmitindo durante a fase de identificação e reprogramação, maior será a probabilidade de o VANT localizá-la, ou seja, ela estará exposta por mais tempo, facilitando assim o encontro entre VANT e Estação de Coleta.

Ao levarmos em consideração, os resultados obtidos no primeiro cenário, em comparação com o segundo cenário, sendo que a única diferença entre tais é que o *timeslot* em que as estações de coleta transmitem podem se repetir. Sendo assim, analisando os resultados obtidos, através da Figura 94 até a Figura 98 notamos uma diferença expressiva à partir de 15 estações de coleta em relação à distância total percorrida pelos VANTs, tempo de voo, quantidade de saltos e conseqüentemente de decisões tomadas pelos VANTs. Como o sistema de aprendizado se dá por experiência, ou seja, quanto mais o VANT se desloca pelas estações em diferentes *timeslots*, mais o VANT agrega conhecimento para suas próprias tomadas de decisão. De acordo com a Figura 97, temos um aumento de

quase 50% nas tomadas de decisões por parte do VANT. No sistema onde os *timeslots* não são compartilhados, o VANT sabe que se encontrar uma EC transmitindo naquele instante, ele automaticamente sabe que não deve se deslocar mais procurando por outras estações até que aquele *timeslot* termine, portanto ele aguarda no local até que o *timeslot* termine e se dirija até a próxima EC de sua rota. A consequência disso é que o VANT não se desloque sem necessidade, buscando ECs em *timeslots* já ocupados.

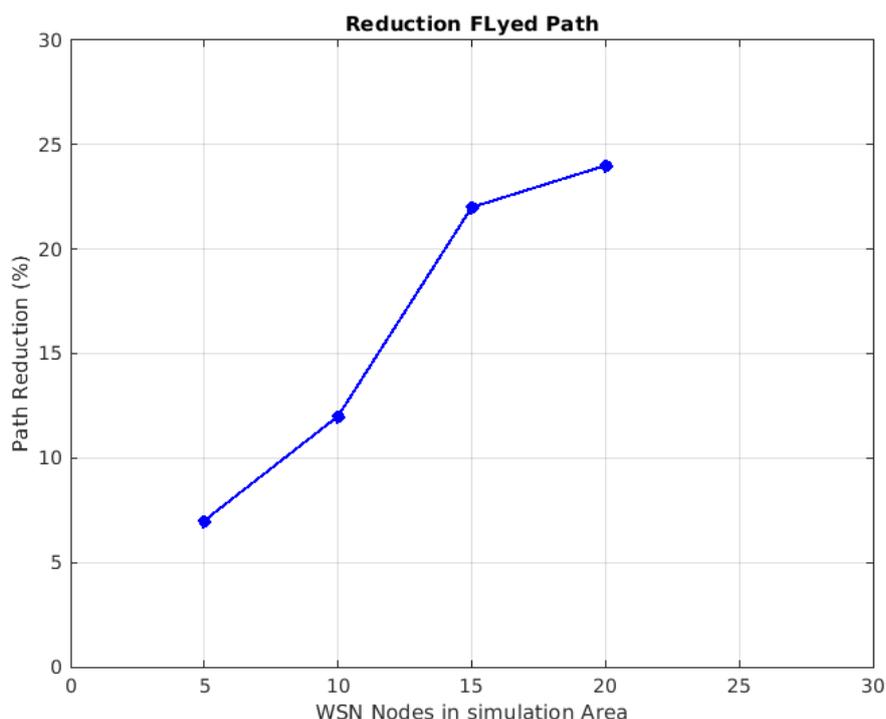


Figura 115 – Percentual de redução de percurso do VANT.

De acordo com a Figura 115, o percentual de redução de percurso chega à 24% em relação à cenários onde o VANT simplesmente avança sobre o *timeslot* já ocupado. Isso tem um impacto direto sobre a autonomia do VANT como já discutido anteriormente, e diretamente associado à quantidade de VANTs que serão utilizados no cenário para identificação e reprogramação das estações de coleta, além do tempo de voo.

A Figura 116 apresenta o cenário de simulação 09 para 15 estações de coleta (pior caso da Figura 57), trabalhando com *timeslots* repetidos. Como podemos ver na simulação, o eixo vertical principal está associado a distância percorrida pelo(s) VANT(s) e o eixo secundário é referente a quantidade de ECs pendentes para serem descobertas ainda. O VANT em função do tempo move-se constantemente e por longos períodos não encontra uma EC, praticamente consumindo sua autonomia de deslocamento.

Ainda sob o mesmo cenário, na Figura 117 temos as tomadas de decisão por VANT, em função da distância percorrida por todos os VANTs no cenário de simulação.

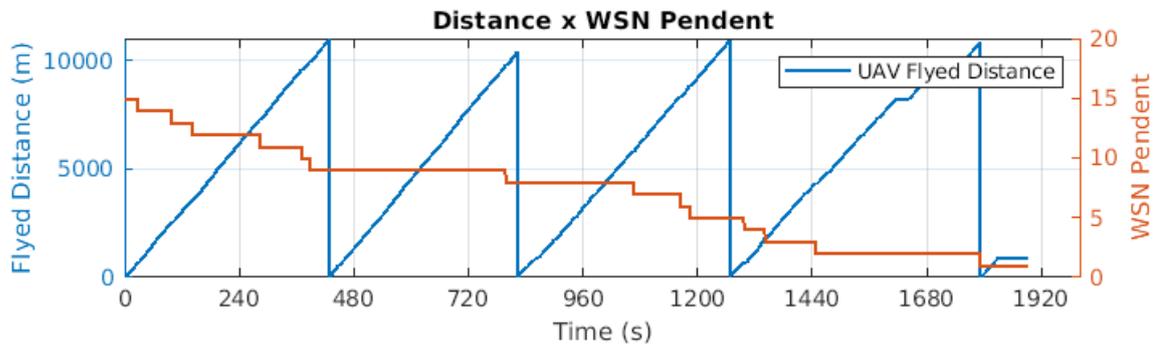


Figura 116 – Distância percorrida, e EC pendentes em função do tempo.

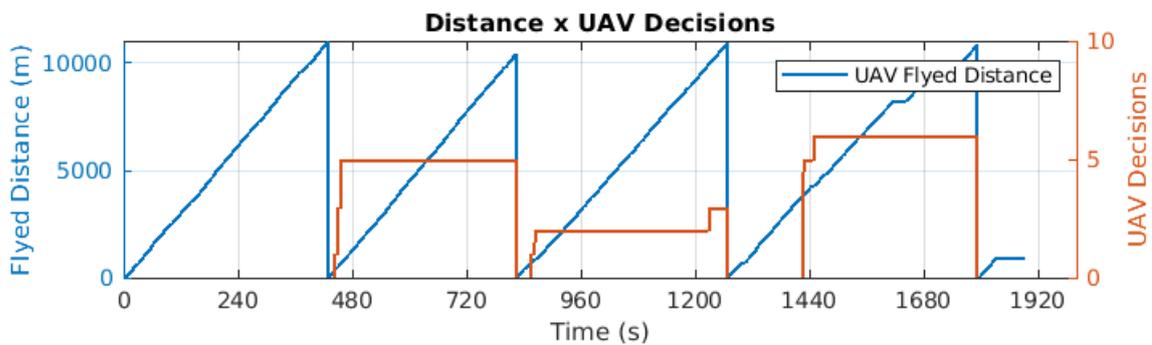


Figura 117 – Decisões por VANT em função das ECs pendentes.

Como podemos ver na Figura 117, a experiência de um VANT é transmitida para os próximos ao ocorrer a substituição de um VANT devido sua autonomia, criando assim um sistema colaborativo, onde os novos VANTs que entram em cena, não tem a necessidade de percorrer todas as estações, por todos os *timeslots* novamente para ter acesso ao conhecimento sobre o cenário que o VANT anterior continha. Percebemos que o primeiro VANT fez o mapeamento dos *timeslots* e sensores já conhecidos. Ao entrar o novo VANT em cena, ele já passa a operar e tomar decisões de acordo com a base de dados transmitida pelo VANT anterior, e o sistema se repete com as demais substituições. Embora este não seja um comportamento exclusivo deste cenário, o fato de o VANT se locomover mais faz com que aumente o aprendizado referente ao cenário, e transmitisse esses dados aos demais VANTs.

Além disso, foram realizadas simulações com velocidade do VANT de 15m/s e 25m/s. Como podemos ver nas Figura 94 em diante, não houve impacto significativo em trabalhar com 15m/s ou 25m/s à não ser em relação ao tempo de voo. O fato de o VANT voar em 15 ou 25m/s e, em média percorrerem a mesma distância, apenas fará com que voe por mais tempo ou menos, e foi exatamente isso que os resultados apresentaram. Além disso, conforme apresentado na Figura 118, temos uma redução do tempo de descoberta e reprogramação considerável ao aumentarmos a velocidade do VANT.

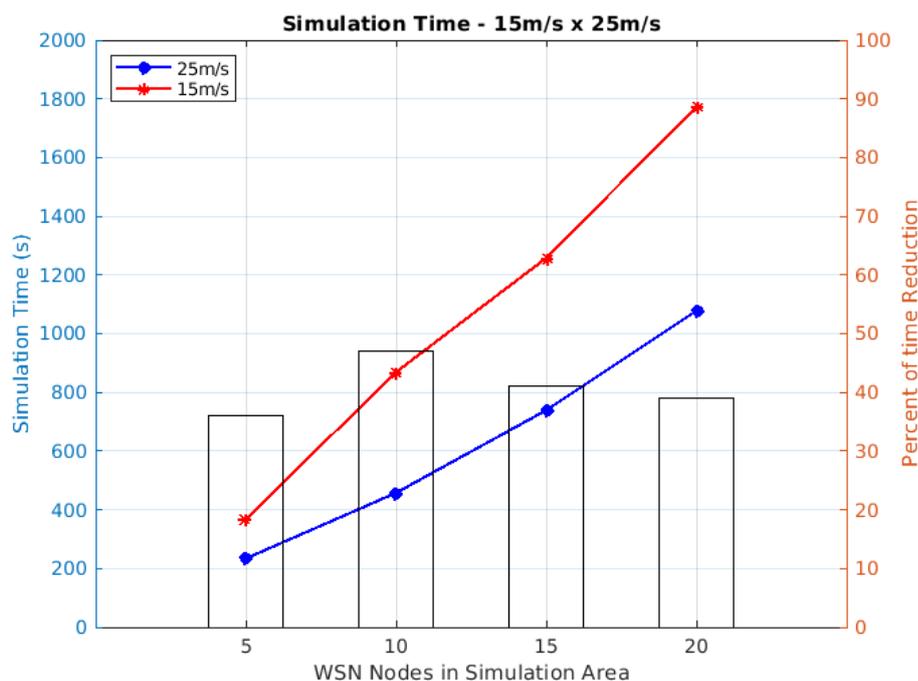


Figura 118 – Tempo de simulação para 15m/s e 25m/s.

Ao analisarmos o quinto cenário “*WaitUntillFind*” onde o VANT fica aguarda sobre a área de cobertura da EC até que a mesma entre em modo de transmissão, para coletar suas informações e reprogramá-la, para que em seguida vá para a próxima EC em sua rota, temos os dados apresentados à partir da Figura 77. Ao analisarmos o pior caso, houve 291 *timeslots*, para 20 ECs, ou seja, uma média aproximada de 15 *timeslots* (de um total de 20 *timeslots* para o cenário) por EC para encontrá-la. Ainda assim, através da Figura 119, podemos perceber um intervalo de tempo maior entre 45min e 59min, onde o simplesmente ocorre um salto entre os *timeslots* 129 e 168. A diferença entre estes intervalos foi de 39 *timeslots*, porém como no cenário apenas considerávamos 20 *timeslots*, ocorreu que no momento em que o VANT estava pra localizar a EC, como podemos ver na Figura 119, sobre o eixo vertical secundário ocorreu a substituição do VANT por questões de autonomia de tempo, obrigando-o a permanecer mais uma rodada de *timeslots* sobre a mesma posição, elevando ainda mais o tempo necessário para que a EC de coleta fosse identificada.

Já ao analisarmos o melhor cenário do sistema *WaitUntillFind* dado pela Figura 120, em que o VANT aguarda sobre a posição da estação de coleta para identificá-la e reprogramá-la, temos a análise sobre 10 EC onde houve nosso melhor cenário, com apenas 35 *timeslots* sequenciais para identificação de todas as 10 ECs e reprogramação das mesmas, em comparação com a média de 68 *timeslots*, de acordo com a Figura 81. Como podemos ver, as conexões entre o VANT e a EC ao qual aguarda são quase que contínuas, o que favoreceu a descoberta dos sensores.

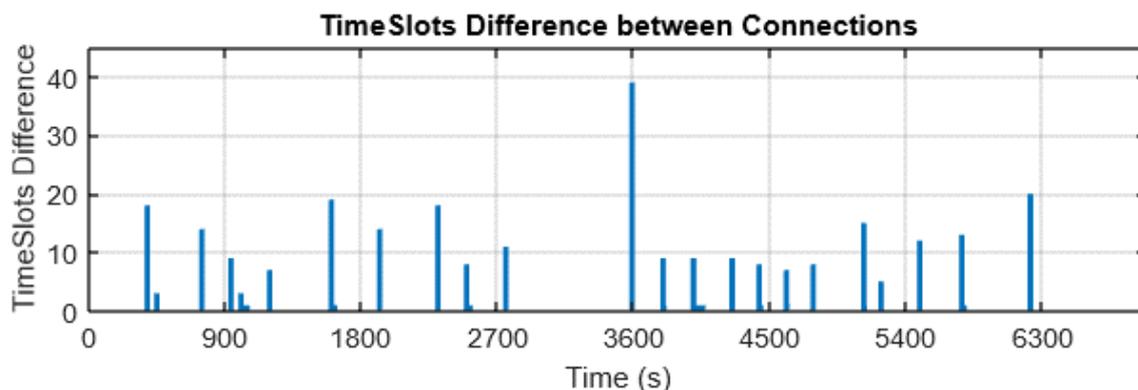


Figura 119 – Diferença de Timeslots entre Conexões do VANT.

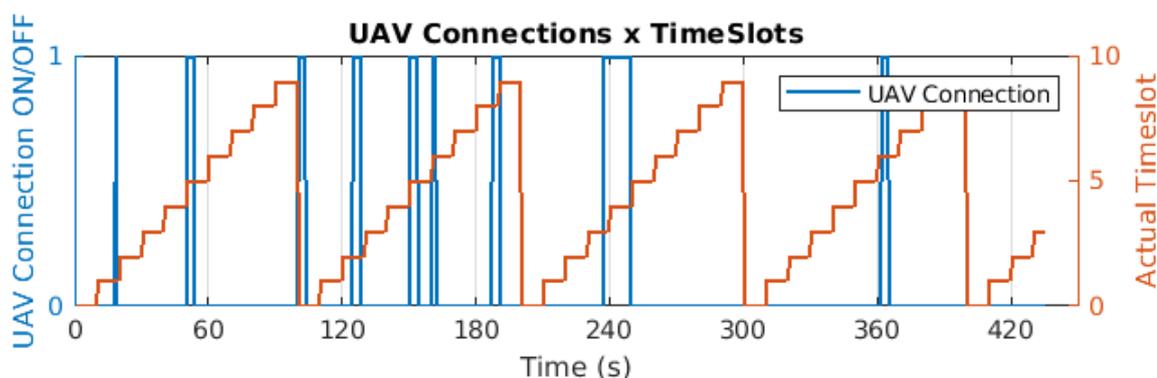


Figura 120 – Conexões do VANT em função dos timeslots.

Porém, no geral, ao analisarmos o modelo apresentado em relação a distância percorrida, ou mesmo quantidade de saltos é claro que ele apresentará resultados favoráveis, porém ao analisarmos pelo ângulo do tempo de voo ou mesmo pela contagem de *timeslots* total do cenário, que são variáveis diretamente dependentes, o modelo não apresenta bons resultados em relação ao primeiro cenário, onde o VANT desloca-se por todas as estações até localizar todas em campo.

Na Figura 93, onde comparamos a quantidade de VANTs necessários para identificar todas as ECs, e Figura 95 onde comparamos o tempo necessário para que isso ocorra, o sistema *WaitUntillFind* apresenta um consumo significativo de VANTs e tempo necessário para completar o percurso. Em relação à Figura 97, referente a comparação de tomadas de decisões, é lógico concluirmos que como o VANT aguarda sobre a EC célula atual até que a mesma esteja em modo de *broadcast*, não há como o VANT possui conhecimento do estado das demais ECs para tomada de decisões, portanto, o modelo de tomada de decisões não é aplicável neste cenário, não há como realizarmos tais comparações.

Ao realizarmos a comparação entre o cenário com e sem restrição de *timeslots*, temos uma grande vantagem sobre o cenário onde o VANT utiliza e toma decisões baseadas

nessas informações. De acordo com a Figura 121, o ganho em relação a tempo de voo chega a 250% no cenário com 20 Estações de Coleta.

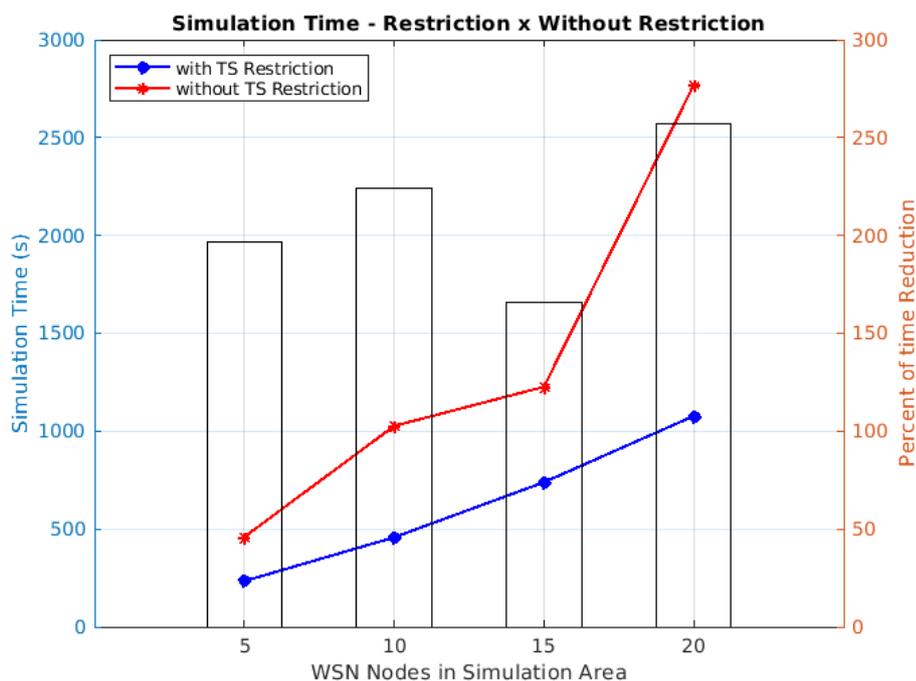


Figura 121 – Tempo de Simulação com e sem restrição de timeslots.

De acordo com a Figura 122, vejamos que a diferença de distância percorrida pelos VANTs não é tão significativa em relação à diferença apresentada no tempo de voo durante o mesmo período de simulação, porém ainda assim apresenta uma diferença que pode chegar até 25%.

O sistema de restrição de *timeslots* que foi implantado ao sistema dos VANTs permite que o tempo de identificação e reprogramação do cenário seja consideravelmente reduzido, assim como vimos na Figura 121, além da distância percorrida que também foi reduzida, evitando assim que o VANT se locomova-se sem necessidade, uma vez que o timeslot já foi identificado, não há por que ele sobrevoar outras estações de coleta.

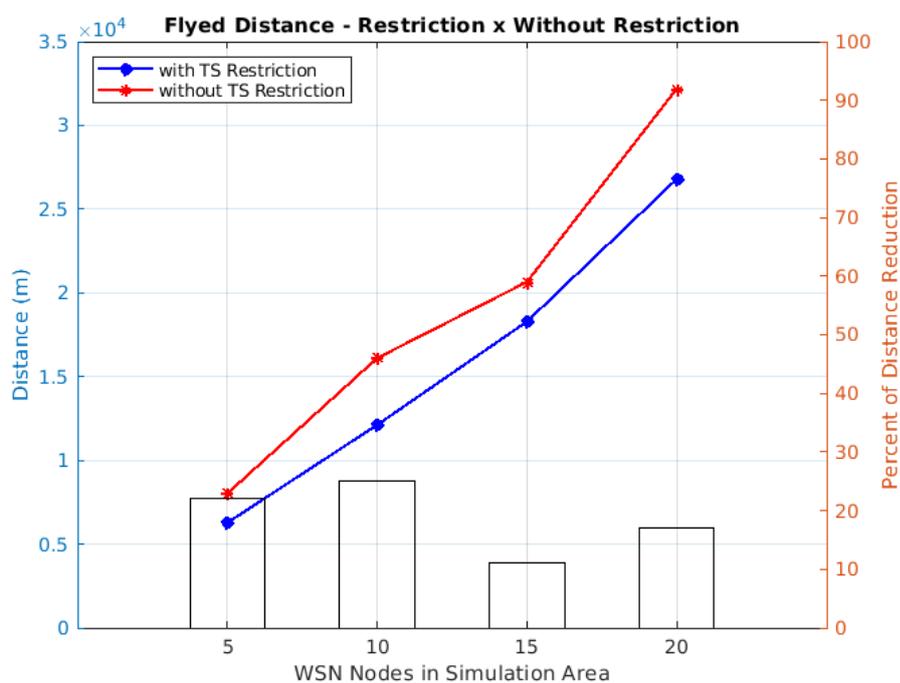


Figura 122 – Distância percorrida pelos VANTs com e sem restrição de timeslots.

## 7 Conclusão

Neste trabalho primeiramente foi apresentado um modelo de otimização de percurso criado por Kashuba, seu modelo foi implementado com sucesso em nosso simulador desenvolvido. Podemos considerar que o modelo foi aplicado com sucesso mediante os dados das nossas simulações estarem muito próximos aos apresentados por Kashuba em seu artigo, como pudermos ver nas Figuras 24 a 29 deste trabalho. Além disso, o trabalho de Kashuba nos deu oportunidade para comparar o modelo de otimização de percurso *NearestPoint* que já estávamos utilizando em nosso simulador até então, garantindo assim que o *NearestPoint* seja o modelo ideal de planejamento de percurso, atingindo melhorias de até 10% em relação aos resultados apresentados por Kashuba. Ainda assim, foi possível realizar um aprofundado estudo em centenas de cenários simulados para que pudéssemos afirmar em que situação o modelo de otimização *NearestPoint* se sobressai sobre o modelo apresentado por Kashuba, e vice-versa.

Para efeitos de comparação, também foi implementado o modelo de percurso não otimizado, onde o simulador faz com que o VANT sobrevoe diretamente cada sensor, para efeitos de comparação, o modelo *NearestPoint* apresentou redução de percurso de aproximadamente 30% em relação ao modelo não otimizado. Ao levarmos em consideração a Figura 106, onde apresenta a distância total percorrida pelo VANT para reprogramar todas as estações, podemos ver a otimização que o *NearestPoint* apresenta no total do percurso, em termos de redução do percurso do VANT, e isso tem um impacto direto sobre sua autonomia sobre o sistema.

Definido o modelo de otimização de percurso, foi possível partir para a segunda fase do projeto, onde tratamos a programação de tempo de *broadcast* e hibernação das estações de coleta, de acordo com o percurso do VANT. Para melhor gerenciamento do tempo de hibernação, tempo de *broadcast*, tempo de percurso total, optamos por dividir estes tempos em *timeslots*, facilitando o gerenciamento de todos temporizadores dos objetos em nosso simulador. Além disso, um sistema que trouxe grande melhoria ao projeto foi o banco de dados de aprendizado do VANT em relação às estações de coleta restritas/permitidas para cada *timeslot*. Com isso, o VANT pode aprender sobre o cenário e evitar sobrevoar determinadas estações em determinados *timeslots*. Como podemos ver, o impacto desse sistema de restrições foi apresentado na Figura 95, fazendo com que o tempo de voo médio para 20 estações de coleta caísse de 45 minutos para 15 minutos, e para cenários menos populosos, com cinco estações apresentou uma redução de 7 minutos para 3 minutos de tempo de voo, portanto, baseado nos valores apresentados concluímos que o sistema de restrição de *timeslots* implementado tem um efeito positivo e significativo sobre o modelo.

Como pudemos ver na Figura 111 após a reprogramação de todos os sensores por parte do VANT, o tempo de *broadcast* das estações de coleta chegaram à cair consideravelmente após a reprogramação, já que o VANT em seu percurso se posiciona no limite do raio de comunicação possível da estação de coleta, e aguarda seu início de *broadcast*, coleta os dados e em seguida já a coloca para hibernar novamente, minimizando o tempo de transmissão, e conseqüentemente o consumo de energia por parte das estações de coleta. Como já mencionado anteriormente, o maior responsável por consumo de energia nas estações é o sistema de transmissão sem fio, portanto minimizando este cenário, teremos as estações de coleta com maior tempo de vida útil. Ainda sobre este tema, inicialmente optamos por trabalhar com apenas um único *timeslot* de transmissão por parte das estações de coleta, o que tornava seu tempo de *broadcast* muito rápido. Ao alterarmos o parâmetro de simulação de tempo de *broadcast* das estações de um para dois *timeslots*, obtivemos uma redução na quantidade de VANTs utilizados, distância total percorrida e principalmente no tempo de voo levado pelos VANTs para localizar e reprogramar todas as estações de coleta. Portanto, devido à estas variações muito grandes, optamos por consumir um pouco mais de energia no período inicial de reprogramação das estações, minimizando o tempo de busca delas. De acordo com os resultados apresentados, como apresentado nas Figuras 114 e 115, percebemos que com dois *timeslots* de tempo de broadcast, as estações de coleta transmitem em média três vezes antes de serem encontradas e reprogramadas pelo VANT até que o cenário esteja completamente reprogramado, o que já não ocorre ao trabalharmos com apenas um único *timeslot*, fazendo com que as estações de coleta transmitam por mais de dez *timeslots* até serem reprogramadas, consumindo muito mais energia.

Após a validação do modelo, geramos 14000 simulações variando parâmetros como quantidade de estações de coleta, tempo de *broadcast*, velocidade do VANT, habilitando ou não o sistema de restrição de *timeslots*, repetindo os *timeslots* ou individualizando os mesmos para que pudéssemos obter a maior quantidade de resultados médios comparativos, além da análise de resultados analíticos, e extremos fora da curva, portanto foi fundamental entendermos quais as variáveis que afetam diretamente o desempenho do sistema como um todo. Como os autores em (ZHAN Y. ZENG, 2018) afirmam em seu artigo abaixo, tradução livre.

“O problema em projetar o período de ativação dos sensores (estações de coleta) em sincronia com o percurso do VANT para coleta de dados de forma eficiente em termos de energia, é um tema novo e desafiador, do qual ainda não foi rigorosamente estudado.” – Tradução Livre.

Futuros trabalhos que podem ser implementados no simulador estariam relacionados à criação de um sistema de controle de bateria/energia das estações de coleta, assim como foi feito com os VANTs. Ao esgotar a energia de uma estação, ela simplesmente não tem mais capacidade de coletar informações sensoriais, processá-las, e nem mesmo transmitir ao

VANT. O que conseqüentemente traria um impacto ao VANT, saber se a estação de coleta simplesmente não está em seu *timeslot* de transmissão, ou se sua energia acabou e parte para a próxima estação ou depósito, informado ao controle de que tais coordenadas não há mais comunicação sensorial. Além disso, o sistema atualmente está projetado para um VANT por vez realizar o percurso, e sendo assim, quando sua autonomia chega ao limite ele retorna à base, repassa os dados das rotas percorridas e sensores já conhecidos com o próximo VANT, que continua seu percurso. Uma possível implementação para trabalhos futuros ao simulador seria coordenar múltiplos VANTs simultaneamente, dividindo as estações em grupos para cada VANT, reduzindo o percurso e tempo necessário para localização e reprogramação de todas as estações.

## Referências

- ADAM, Lamont B. Gary Pohl J. Multi-objective uav mission planning using evolutionary computation. p. 1268—1279, 2008. Citado 3 vezes nas páginas 39, 40 e 41.
- ALEMAYEHU, J. H. Kim T. S. Efficient nearest neighbor heuristic tsp algorithms for reducing data acquisition latency of uav relay wsn. *Wirel. Pers. Commun.*, p. 3271—3285, 2017. Citado na página 60.
- ARDUPILOT. Ardupilot – the open source autopilot. 2017. Disponível em: <<http://ardupilot.org/>>. Acesso em: 15 de outubro de 2017. Citado na página 13.
- BADIS A. MUNARETTO, K. Al Agha G. Pujolle H. Qos for ad hoc networking based on multiple metrics: Bandwidth and delay. *Mob. Wirel. Commun. Networks - Proc*, p. 15–18, 2003. Citado na página 27.
- BANCOMUNDIAL. Banco mundial – the world bank. 2017. Disponível em: <<http://www.worldbank.org/>>. Acesso em: 5 de Novembro de 2017. Citado na página 14.
- BAUMANN F. LEGENDRE, P. Sommer BR. Generic mobility simulation framework (gmsf). *Proc. Int. Symp. Mob. Ad Hoc Netw. Comput.*, p. 49–56, 2008. Citado 2 vezes nas páginas 22 e 43.
- BIOMO T. KUNZ, M. St-Hilaire Y. Zhou J.-D. Unmanned aerial ad hoc networks: Simulation-based evaluation of entity mobility models' impact on routing performance. *Aerospace*, vol. 2, no. 3, p. 392—422, 2015. Citado 3 vezes nas páginas 36, 37 e 44.
- BOUACHIR A. ABRASSART, F. Garcia N. Larrieu O. A mobility model for uav ad hoc network. *Unmanned Aircr. Syst. (ICUAS)*, p. 383—388, 2014. Citado na página 36.
- BRUGGEMANN J.J. FORD, R. A. Walker T.S. Control of aircraft for inspection of linear infrastructure. *IEEE Transactions on Control Systems Technology*, p. 1397–1409, 2011. Citado na página 23.
- BUJARI C. T. CALAFATE, J. C. Cano P. Manzoni C. E. Palazzi D. Ronzani A. Flying ad-hoc network application scenarios and mobility models. *Int. J. Distrib. Sens. Networks*, p. 1–17, 2017. Citado na página 44.
- C++. "linguagem de programação c++". 2018. Disponível em: <<http://www.cplusplus.com/>>. Acesso em: 7 de fevereiro de 2018. Citado 2 vezes nas páginas 43 e 44.
- CAI F.R. YU, J. Li Y. Zhou L. Lamont Y. Medium access control for unmanned aerial vehicle (uav) ad-hoc networks with full- duplex radios and multipacket reception capability. *IEEE Transactions on Vehicular Technology*, p. 390–394, 2013. Citado na página 29.
- CHAVES, Q. N. Proposta de modelo de veículos aéreos não tripulados (vants) cooperativos aplicados a operações de busca. p. 149, 2013. Citado 6 vezes nas páginas 13, 14, 20, 21, 39 e 44.

- CHENG MOORE, S. Bow-Nan. A comparison of manet routing protocols on airborne tactical networks. *Military Communications Conference, MILCOM*, p. 1–6, 2012. Citado 3 vezes nas páginas 22, 25 e 29.
- COELHO, W. L. Simões E. F. Onde posicionar sensores de umidade e de tensão de Água do solo próximo da planta para um manejo mais eficiente da Água de irrigação. *International Conference on Unmanned Aircraft Systems (ICUAS)*, p. 1—6, 2015. Citado 4 vezes nas páginas 15, 16, 18 e 31.
- DESTA ARI KERÄNEN, Teemu Kärkkäinen Esa Hyytiä Jörg Ott Michael Solomon. Evaluating (geo) content sharing with the one simulator. *Proceedings of the 14th ACM Symposium Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2013. Citado na página 44.
- DRONECODE. Dronecode - the open source uav plataform. 2017. Disponível em: <<https://www.dronecode.org>>. Acesso em: 20 de agosto de 2017. Citado na página 13.
- EMBRAPA. Embrapa - drone pode trazer economia de insumos no campo e aumento da produtividade. 2017. Disponível em: <<https://www.embrapa.br/busca-de-noticias/-/noticia/2448225/drone-pode-trazer-economia-de-insumos-no-campo-e-aumento-da-produtividade>>. Acesso em: 5 de Novembro de 2017. Citado na página 15.
- EZE S. ZHANG, E. Liu E. C. Vehicular ad hoc networks (vanets): Current state, challenges, potentials and way forward. *Proc. 20th Int. Conf. Autom. Comput. Futur. Autom. Comput. Manuf.*, p. 176—181, 2014. Citado na página 27.
- FREW, T.X. Brown E.W. Airborne communication networks for small unmanned aircraft systems. *Proceedings of the IEEE*, p. 2008–2027, 2008. Citado na página 29.
- FREW, T. X. Brown E. W. Networking issues for small unmanned aircraft systems. *Journal of Intelligent and Robotic Systems*, p. 21—37, 2009. Citado 2 vezes nas páginas 29 e 32.
- FUJIMOTO K. S. PERUMALLA, G. F. Riley R. M. Network simulation. p. 66, 2007. Citado 2 vezes nas páginas 37 e 43.
- GHICA, O. C. Sidnet-swans: A simulator and integrated development platform for sensor networks applications. *Proc.the 6th ACM Conference on Embedded Networked Sensor Systems*, p. 385–386, 2008. Citado na página 45.
- JIN, L. Tang J. Optimal coverage path planning for arable farming on 2d surfaces. *Trans. ASABE*, p. 283—295, 2010. Citado 3 vezes nas páginas 38, 39 e 43.
- JO J. HEO, J. Jung B. Kim H. Min K. A rendezvous point estimation considering drone speed and data collection delay. *4th Int. Conf. Comput. Appl. Inf. Process. Technol. CAIPT*, p. 1–4, 2018. Citado 2 vezes nas páginas 45 e 72.
- JORGE, Ricardo Y. Inamasu Lúcio André de Castro. Uso de veículos aéreos não tripulados (vant) em agricultura de precisão. *EMPRAPA Instrumentação*, 2014. Citado 3 vezes nas páginas 14, 31 e 39.

- KARAGIANNIS O. ALTINTAS, E. Ekici G. Heijenk B. Jarupan K. Lin T. Weil G. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *Communications Surveys and Tutorials*, p. 584–616, 2011. Citado 2 vezes nas páginas 27 e 30.
- KASHUBA V. I. NOVIKOV, O. I. Lysenko I. V. Alekseeva S. V. Optimization of uav path for wireless sensor network data gathering. *IEEE Int. Conf. Actual Probl. Unmanned Aer. Veh. Dev.*, p. 280—283, 2015. Citado 7 vezes nas páginas 6, 45, 55, 57, 58, 59 e 67.
- KERÄNEN JÖRG OTT, Teemu Kärkkäinen Ari. The one simulator for dtn protocol evaluation. *2nd International Conference on Simulation Tools and Techniques*, 2009. Citado na página 44.
- KERÄNEN TEEMU KÄRKKÄINEN, Jörg Ott Ari. Simulating mobility and dtns with the one. *Journal of Communications*, p. 92–105, 2010. Citado na página 44.
- KUIPER, S. Nadjm-Tehrani E. Mobility models for uav group reconnaissance applications. *Second Int. Conf. Wirel. Mob. Commun. ICWMC*, p. 2—8, 2006. Citado 2 vezes nas páginas 36 e 44.
- KUMAR, J. Kumar S. Comparative analysis of proactive and reactive routing protocols in mobile ad-hoc networks (manet). *Journal of Information and Operations Management*, p. 92–95, 2012. Citado na página 26.
- LI M. ST-HILAIRE, T. Kunz Y. Enhancements to reduce the overhead of the reactive-greedy- reactive routing protocol for unmanned aeronautical ad-hoc networks. *8th International Conference on Wireless Communications (WICOM)*, p. 1–4, 2012. Citado na página 29.
- LI M. ST-HILAIRE, T. Kunz Y. Improving routing in networks of uavs via scoped flooding and mobility prediction. *IFIP Wireless Days (WD)*, p. 1–6, 2012. Citado na página 29.
- LIN Q. SUN, J. Li F. Yang L. A novel geographic position mobility oriented routing strategy for uavs. *Journal of Computational Information Systems*, p. 709—716, 2012. Citado 2 vezes nas páginas 34 e 35.
- MAHMOUD, N. Mohamed S. Collaborative uavs cloud. *International Conference on Unmanned Aircraft Systems, ICUAS*, 2014. Citado 3 vezes nas páginas 21, 23 e 28.
- MAHMOUD, N. Mohamed S. Broker architecture for collaborative uavs cloud computing. *Int. Conf. Collab. Technol. Syst. CTS*, p. 212–219, 2015. Citado na página 23.
- MATLAB. Matlab. 2018. Disponível em: <<https://www.mathworks.com/solutions/wireless-communications.html>>. Acesso em: 7 de fevereiro de 2018. Citado na página 43.
- MAXA G. ROUDIÈRE, N. Larrieu J.-A. Emulation-based performance evaluation of routing protocols for uanets. *Communication Technologies for Vehicles. Springer*, p. 227—240, 2015. Citado 2 vezes nas páginas 33 e 34.
- MAXA M. SLIM, B. Mahmoud N. Larrieu T. Resco F.-Toulouse J. Extended verification of secure uanet routing protocol. 2016. Citado na página 44.

MAXA MOHAMED SLIM BEM MAHMOUD, Nicolas Larrieu Jean-Aimé. Survey on uanet routing protocols and network security challenges. *ACM Computing Surveys (CSUR)*, 2017. Citado 11 vezes nas páginas 13, 14, 20, 29, 30, 31, 32, 33, 34, 35 e 70.

MICROSOFT. "microsoft visual studio professional". 2018. Disponível em: <<https://www.visualstudio.com/pt-br/>>. Acesso em: 5 de abril de 2018. Citado na página 47.

MOHAMED J. AL-JAROODI, I. Jawhar S. Lazarova-Molnar N. Middleware requirements for collaborative unmanned aerial vehicles. *International Conference on Unmanned Aircraft Systems (ICUAS)*, p. 1051—1060, 2013. Citado 7 vezes nas páginas 13, 14, 21, 23, 32, 41 e 42.

MOHAMED J. AL-JAROODI, I. Jawhar S. Lazarova-Molnar N. A service-oriented middleware for building collaborative uavs. *J. Intell. Robot. Syst. Theory Appl*, p. 309—321, 2014. Citado 5 vezes nas páginas 14, 21, 22, 25 e 32.

MOHAMED J. AL-JAROODI, I. Jawhar S. Lazarova-Molnar N. Middleware requirements for collaborative unmanned aerial vehicles. *International Conference on Unmanned Aircraft Systems (ICUAS)*, p. 1051—1060, 2017. Citado na página 14.

MOHSENI R. HASSAN, A. Patel R. Razali S. Comparative review study of reactive and proactive routing protocols in manets. *4th IEEE International Conferenc*, p. 304—309, 2010. Citado 3 vezes nas páginas 20, 21 e 34.

NS2. The ns2 network simulator. 2018. Disponível em: <<http://nsnam.sourceforge.net/wiki/index.php>>. Acesso em: 7 de fevereiro de 2018. Citado 2 vezes nas páginas 37 e 43.

NS3. The ns-3 network simulator. 2018. Disponível em: <<http://www.nsnam.org>>. Acesso em: 7 de fevereiro de 2018. Citado na página 37.

OLIVIERI, M. Endler B. A distributed algorithm for aerial data collection from wireless sensors networks by uavs. p. 129—136, 2017. Citado na página 46.

OMNET++. Omnet++ simulator. 2018. Disponível em: <<https://www.omnetpp.org/intro>>. Acesso em: 7 de fevereiro de 2018. Citado na página 44.

OPENUAV. "open uav project". 2017. Disponível em: <[https://openuav.us/wiki/index.php/OpenUAV\\_Project](https://openuav.us/wiki/index.php/OpenUAV_Project)>. Acesso em: 15 de outubro de 2017. Citado na página 13.

OPNET. Opnet. 2018. Disponível em: <<http://www.opnet.com>>. Acesso em: 7 de fevereiro de 2018. Citado na página 37.

PAPARAZZI. Paparazzi uav. 2017. Disponível em: <[http://wiki.paparazziuav.org/wiki/Main\\_Page](http://wiki.paparazziuav.org/wiki/Main_Page)>. Acesso em: 15 de outubro de 2017. Citado na página 13.

POLO G. HORNERO, C. Duijneveld A. García O. Casas J. Design of a low-cost wireless sensor network with uav mobile node for agricultural applications. *Comput. Electron. Agric*, p. 19—32, 2015. Citado na página 71.

POVO, Gazeta do. "gazeta do povo", matéria: “o que se aprende na universidade não se leva ao campo”. 2018. Disponível em: <<http://www.gazetadopovo.com.br/agronegocio/agricultura/o-que-se-aprende-na-universidade-nao-se-leva-ao-campo-ee5a9sfawatxctrceucrquj0>>. Acesso em: 5 de Janeiro de 2018. Citado na página 14.

- PUCPR. Curso de pós-graduação para vants. 2017. Disponível em: <<http://especializacao.pucpr.br/cursos/rpas-drones-e-vants-em-aplicacoes-civis-e-comerciais-ead/>>. Acesso em: 5 de Novembro de 2017. Citado na página 15.
- PX4. Px4 – open source for drones. 2017. Disponível em: <<http://px4.io/>>. Acesso em: 15 de outubro de 2017. Citado na página 13.
- RASHED, M. Soy Turk S. Analyzing the effects of uav mobility patterns on data collection in wireless sensor networks. *Sensors*, p. 413, 2017. Citado na página 46.
- RATHINAM, S. Vision based following of locally linear structures using an unmanned aerial vehicle. *In Proc. of the IEEE Conference on Decision and Control*, 2005. Citado na página 23.
- RYAN, A. Decentralized control of unmanned aerial vehicle collaborative sensing missions. *Proc. of the American Control Conference*, p. 4672–4677, 2007. Citado 2 vezes nas páginas 23 e 24.
- SAHINGOZ, O. K. Mobile networking with uavs: Opportunities and challenges. *International Conference on Unmanned Aircraft Systems, ICUAS*, 2013. Citado na página 23.
- SAHINGOZ, O. K. Networking models in flying ad-hoc networks (fanets): Concepts and challenges. *Intell. Robot. Syst. Theory Appl*, p. 513—527, 2014. Citado na página 25.
- SENOUCI, G. Pujolle S.M. An overview and study of security issues and challenges in mobile ad-hoc networks (manet). *IEEE Int. Conf. Commun.*, p. 4057—4061, 2004. Citado na página 26.
- SHIRANI M. ST-HILAIRE, T. Kunz Y. Zhou J. Li-L. Lamont R. Combined reactive-geographic routing for unmanned aeronautical ad-hoc networks. *Proceedings of the IEEE*, p. 2820—2826, 2012. Citado 2 vezes nas páginas 29 e 33.
- SHRIT S. MARTIN, K. Alagha G. Pujolle O. A new approach to realize drone swarm using ad-hoc network. *16th Annu. Mediterr. Ad Hoc Netw. Work*, p. 1–5, 2017. Citado na página 24.
- SUJIT, R. Beard P. B. Multiple uav path planning using anytime algorithms. *Proc. Am. Control Conf*, p. 2978—2983, 2009. Citado na página 39.
- TAMINE, K. A hybrid multiagent routing approach for wireless ad hoc networks. 2012. Citado 2 vezes nas páginas 35 e 43.
- TEICHMANN M. DORDA, J. Vitek V. Smrž V. Michalík D. Unmanned aerial vehicles routing problem. p. 602—607, 2014. Citado na página 44.
- THEONE. The one. 2018. Disponível em: <<http://www.netlab.tkk.fi/tutkimus/dtn/theone/>>. Acesso em: 7 de fevereiro de 2018. Citado na página 44.
- TOTH P., D. Vigo. The vehicle routing problem. *Society for Industrial and Applied Mathematics*, 2001. Citado 2 vezes nas páginas 39 e 40.
- TUCHELMANN, P.B.Y. Context-aware qos control for wireless mesh networks of uavs. *International Conference Computer Communications and Networks (ICCCN)*, p. 1–6, 2011. Citado na página 29.

- UAVSIMULATOR. Uavsimulator. 2019. Disponível em: <<https://youtu.be/GzV9ihQQMSQ>>. Acesso em: julho de 2019. Citado 6 vezes nas páginas 6, 47, 57, 59, 62 e 74.
- UMESH S. MEWADA, L. Iaddhani K. Bunkar K. S. An overview and study of security issues and challenges in mobile ad-hoc networks (manet). *International Journal of Computer Science and Information Security*, p. 106—111, 2011. Citado na página 26.
- VACHTSEVANOS L. TANG, J. Reimann G. An intelligent approach to coordinated control of multiple unmanned aerial vehicles. *The 60th Annual American Helicopter Society*, 2004. Citado na página 23.
- WANG, A. Botea K.-H. C. A scalable multi-agent path planning algorithm with tractability and completeness guarantees. *JAIR - J. Artif. Intell. Res.*, p. 55—90, 2011. Citado 2 vezes nas páginas 38 e 39.
- WANG F. MA, J. Yan D. De S. K. Das C. Efficient aerial data collection with uav in large-scale wireless sensor networks. 2015. Citado na página 45.
- WEINSTEIN, C. Schumacher A. L. Uav scheduling via the vehicle routing problem with time windows. *Proc. AIAA Infotech Aerosp*, p. 1—14, 2007. Citado na página 40.
- WOLFRAMMATHWORLD. Site wolfram mathworld. 2019. Disponível em: <<http://mathworld.wolfram.com/CirclePacking.html>>. Acesso em: 3 de outubro de 2019. Citado na página 69.
- XIONG, F. Shan R. Dronetank: Planning uavs' flights and sensors' data transmission under energy constraints. *Sensors (Switzerland)*, 2018. Citado na página 45.
- XIONG, F. Shan R. Dronetank: Planning uavs' flights and sensors' data transmission under energy constraints. *Sensors (Switzerland)*, 2018. Citado na página 71.
- YUE, Z. Jiang W. Path planning for uav to collect sensors data based on spiral decomposition. *Procedia Comput. Sci*, p. 873—879, 2018. Citado na página 71.
- ZENG, R. Zhang Y. Energy-efficient uav communication with trajectory optimization. *IEEE Trans. Wirel. Commun*, p. 3747—3760, 2017. Citado na página 71.
- ZHAN Y. ZENG, R. Zhang C. Energy-efficient data collection in uav enabled wireless sensor network. *IEEE Wirel. Commun. Lett*, p. 328—331, 2018. Citado 3 vezes nas páginas 45, 71 e 129.