



JOÃO PAULO DE ARAUJO

**RESTABELECIMENTO DA COMUNICAÇÃO
ENTRE PARTIÇÕES DESCONEXAS DE UMA
REDE DE SENSORES SEM FIO UTILIZANDO
VEÍCULOS AÉREOS NÃO TRIPULADOS**

LAVRAS - MG

2015

JOÃO PAULO DE ARAUJO

**RESTABELECIMENTO DA COMUNICAÇÃO ENTRE
PARTIÇÕES DESCONEXAS DE UMA REDE DE SENSORES
SEM FIO UTILIZANDO VEÍCULOS AÉREOS NÃO
TRIPULADOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores e Sistemas Embarcados, para a obtenção do título de Mestre.

Orientador

Dr. Tales Heimfarth

LAVRAS - MG

2015

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha
Catalográfica da Biblioteca Universitária da UFLA, com dados
informados pelo próprio autor.**

de Araujo, João Paulo.

Restabelecimento da Comunicação entre Partições Desconexas de
Uma Rede de Sensores Sem Fio Utilizando Veículos Aéreos Não
Tripulados/ João Paulo de Araujo. – Lavras : UFLA, 2015.

87 p. : il.

Dissertação (mestrado acadêmico) – Universidade Federal de
Lavras, 2015.

Orientador: Tales Heimfarth.

Bibliografia.

1. Rede de sensores sem fio. 2. Veículo aéreo não tripulado. 3.
Rede tolerante a atrasos. I. Universidade Federal de Lavras. II.
Título.

JOÃO PAULO DE ARAUJO

**RESTABELECIMENTO DA COMUNICAÇÃO ENTRE
PARTIÇÕES DESCONEXAS DE UMA REDE DE SENSORES
SEM FIO UTILIZANDO VEÍCULOS AÉREOS NÃO
TRIPULADOS**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores e Sistemas Embarcados, para a obtenção do título de Mestre.

APROVADA em 12 de junho de 2015.

Dr. Luiz Henrique Andrade Correia UFLA

Dra. Luciana Bezerra Arantes UPMC

Dr. Tales Heimfarth
Orientador

LAVRAS - MG

2015

Dedico este trabalho à minha mãe, Teresa Aparecida Paulino, pelo amor incondicional e por todo o esforço para que eu atingisse mais este objetivo.

AGRADECIMENTOS

À minha mãe, Teresa Aparecida Paulino, pela confiança e incentivo.

Ao Professor Tales Heimfarth, pelas oportunidades, confiança e amizade.

A Gilson Miranda Junior, um verdadeiro amigo, que sempre esteve presente seja para discutir computação e/ou tomar uma cerveja.

Às festas e ao *rock 'n roll*, por me fazerem lembrar que eu ainda sou um ser humano.

Ao pessoal do GRUBi, pelas piadas, ambiente agradável e aprendizado.

A todos os amigos, colegas e conhecidos que eu não citei explicitamente aqui mas que, de alguma forma, estiveram presentes para que eu atingisse este objetivo.

Meus sinceros agradecimentos.

(...) e que venham as próximas etapas o/

RESUMO

Uma das possíveis definições para o tempo de vida de uma Rede de Sensores Sem Fio (RSSF) é o período que ela se mantém conexa. A formação de partições desconexas na rede pode ser causada tanto por danos físicos ao nó, por falhas internas ou esgotamento de energia. Considerando que desconexões na rede comprometem a sua operação, proporcionar uma alternativa para minimizá-las, pode estender sua vida útil. Diversas abordagens lidam com o problema das desconexões por meio de nós redundantes ou até mesmo utilizando nós móveis como retransmissores. Diferentemente delas, este trabalho apresenta um conjunto de métodos para comunicar partições desconexas de RSSFs utilizando Veículos Aéreos Não Tripulados (VANTs) para transportar fisicamente pacotes entre essas partições. Assim, inicialmente um *cluster head* é eleito como coordenador de cada partição, sendo ele responsável por interagir com o VANT na troca de mensagens entre as partições. O roteamento das mensagens dentro de uma partição é feito utilizando uma extensão de um algoritmo de roteamento geográfico. E, para levar os VANTs às partições, três modelos de movimento são propostos. Um modelo inicial utiliza o ciclo hamiltoniano mínimo entre as partições como rota para um único VANT. Já, o segundo também considera o tráfego existente entre as partições como peso na determinação da rota. Por fim, o terceiro modelo utiliza múltiplos VANTs para gerar rotas menores e que concentrem o tráfego. Experimentos foram realizados, para cada um dos modelos de movimento, em diferentes cenários. Resultados comprovam a eficiência das soluções propostas, observando-se que cada modelo tem sua performance ditada pelas características de tráfego da rede e capacidades de armazenamento dos nós sensores e VANTs.

Palavras-chave: Rede de Sensores sem Fio; Veículo Aéreo Não Tripulado; Rede Tolerante a Atrasos

ABSTRACT

One of the possible definitions for the lifetime of a Wireless Sensor Network (WSN) regards the time it keeps its connectivity. The formation of disjoint segments is related to physical damages to the nodes, internal node failure or energy depletion. Since disconnections compromise the operation of the network, providing alternatives to mitigate them can extend its lifetime. Several proposals tackle the problem of disconnections by means of redundant deployment or even using mobile nodes as relay stations. Differently from previous works, the present one proposes methods to federate disjoint segments of a WSN using Unmanned Aerial Vehicles (UAVs) as data mules, carrying physically packets among the segments. Thus, initially a cluster head is elected as the coordinator for each segment, since it is responsible for the interaction with the UAV when it comes to exchanging packets between segments. The routing process of messages inside a segment is performed by means of an extension of a geographical routing algorithm. And, in order to take the UAVs to the segments, three movement models are proposed. An initial model uses the minimum Hamiltonian cycle among the segments as the route for a single UAV. The second model is a traffic-aware one, since it takes into account the communication between segments to generate the route. Finally, the third one employs multiple UAVs to generate smaller routes that concentrate the traffic as well. Experiments were performed for each of the movement models in different scenarios. Results evince the effectiveness of the proposed solution, showing that each movement model has its performance correlated to the network traffic characteristics along with storage capacity of both sensor nodes and UAVs.

Keywords: Wireless Sensor Network; Unmanned Aerial Vehicle; Delay-Tolerant Network

LISTA DE FIGURAS

Figura 1	Exemplo de uma rede formada por três partições.	14
Figura 2	Visão geral das etapas envolvidas no desenvolvimento do trabalho.....	29
Figura 3	Visualização do critério de formação de RSSFs desconexas.....	31
Figura 4	Exemplo de uso de temporizadores para a eleição do <i>cluster head</i>	34
Figura 5	Parte do cabeçalho de um pacote transmitido na rede.....	35
Figura 6	Exemplo de ciclo realizado por um VANT em cinco partições desconexas, utilizando o modelo ETSP.	41
Figura 7	Exemplo de mapeamento entre um conjunto de <i>cluster heads</i> e o grafo do PDP.....	48
Figura 8	Representação de um cromossomo utilizado pelo algoritmo genético.....	53
Figura 9	Representação do operador de <i>crossover</i>	55
Figura 10	Representação do operador de mutação.....	56
Figura 11	Exemplo das regiões geradas usando o método baseado em algoritmo genético.	57
Figura 12	Latência média para entrega dos pacotes no modelo ETSP.	61
Figura 13	Número médio de pacotes entregues com sucesso para o modelo ETSP.	63
Figura 14	Latência média para entrega dos pacotes no modelo <i>Weighted</i>	66
Figura 15	Escalabilidade da latência para o modelo <i>Weighted</i>	68
Figura 16	Número médio de pacotes entregues com sucesso para o modelo <i>Weighted</i>	69
Figura 17	Latência média para entrega dos pacotes no modelo <i>Genetic</i>	73
Figura 18	Número médio de pacotes entregues com sucesso para o modelo <i>Genetic</i>	75
Figura 19	Número médio de pacotes perdidos nos <i>cluster heads</i> e VANTs.....	76

LISTA DE TABELAS

Tabela 1	Possível tabela de visitas para quatro partições desconexas.....	45
Tabela 2	Parâmetros de configuração dos cenários simulados no modelo ETSP.....	60
Tabela 3	Parâmetros de configuração dos cenários simulados no modelo <i>Weighted</i>	65
Tabela 4	Parâmetros de configuração dos cenários simulados no modelo <i>Genetic</i>	71
Tabela 5	Número médio de pacotes trocados entre os VANTs.....	78

LISTA DE SIGLAS

ETSP	<i>Euclidean Traveling Salesperson Problem</i>
FIFO	<i>First In, First Out</i>
GPSR	<i>Greedy Perimeter Stateless Routing</i>
GRUBi	Grupo de Redes Ubíquas
PDP	<i>Pickup and Delivery Problem</i>
RSSF	Rede de Sensores sem Fio
TSP	<i>Traveling Salesperson Problem</i>
UFLA	Universidade Federal de Lavras
UPMC	<i>Université Pierre et Marie Curie</i>
VANT	Veículo Aéreo Não Tripulado
VRPPD	<i>Vehicle Problem with Pickups and Deliveries</i>

SUMÁRIO

1	Introdução	13
1.1	Contexto	13
1.2	Definição do problema	14
1.3	Justificativa	15
1.4	Objetivos	16
1.4.1	Objetivo geral	16
1.4.2	Objetivos específicos	16
1.5	Contribuições	17
1.6	Estrutura do trabalho	18
2	Referencial teórico	20
2.1	Redes de sensores sem fio	20
2.2	Veículos aéreos não tripulados	21
2.3	Roteamento geográfico	22
2.4	Trabalhos relacionados	23
3	Metodologia	27
3.1	Visão geral	27
3.2	Formação de RSSFs desconexas	30
3.3	Eleição do cluster head	31
3.4	Descoberta de cluster heads	34
3.5	Roteamento intrapartição	35
3.6	Roteamento interpartições	37
3.6.1	Interação entre cluster head e VANT	37
4	Modelos de movimento	39
4.1	ETSP	39
4.2	Weighted	41
4.3	Genetic	45
4.3.1	Problema de otimização interpartições	45
4.3.2	Modelo de movimento para múltiplos VANTs	51
4.3.3	Interações entre VANTs	57
5	Resultados e discussões	59
5.1	ETSP	59
5.1.1	Configuração das simulações	60
5.1.2	Resultados	60
5.2	Weighted	64
5.2.1	Configuração das simulações	64
5.2.2	Resultados	65
5.3	Genetic	70

5.3.1	Configuração das simulações	71
5.3.2	Resultados	72
5.4	Considerações Finais	78
6	Conclusão	80
6.1	Trabalhos futuros	82
	REFERÊNCIAS.....	83

1 Introdução

Neste capítulo, é apresentado o contexto em que se insere este trabalho, assim como o problema abordado e os fatores que justificam a busca por uma solução, os seus objetivos e a estrutura geral do restante deste documento.

1.1 Contexto

As Redes de Sensores Sem Fio (RSSFs) são compostas por nós sensores e podem ser empregadas em diversas situações como, por exemplo, monitoramento de florestas, gerenciamento de desastres e proteção de fronteiras (YICK; MUKHERJEE; GHOSAL, 2008). Uma característica comum de tais aplicações é a sua integração com o ambiente físico, visto que a tarefa de cada nó é relacionada com amostragens desse ambiente.

Em diferentes tipos de aplicações, os ambientes apresentam condições adversas que podem levar a falhas, isolando grupos de nós sensores e, conseqüentemente, comprometendo a operação da rede devido às desconexões (GARCIA-HERNANDEZ et al., 2007). Essas desconexões na rede ocorrem em decorrência de danos físicos causados pelo ambiente onde está inserida, ou mesmo por problemas internos do nó sensor ou esgotamento de energia.

Os nós pertencentes a diferentes partições da rede são incapazes de se comunicarem. Com o aumento das desconexões, eventualmente a rede se tornará particionada a ponto de não conseguir cumprir seus requisitos. Além disso, essas desconexões afetam um requisito importante em diversas aplicações de RSSFs: a longevidade da rede após a sua implantação.

Com a inserção de Veículos Aéreos Não Tripulados (VANTs), é pos-

sível que haja o aumento da capacidade da rede de monitorar objetos de forma mais próxima, além da possibilidade da utilização desses nós como *sinks* móveis (ERMAN et al., 2008). Além disso, também, há a possibilidade do uso da mobilidade para aumentar a conectividade de redes particionadas. Dessa forma, seria tarefa do VANT transportar fisicamente os pacotes entre diferentes partições, por meio de um modelo conhecido como *message ferrying* (ZHAO; AMMAR; ZEGURA, 2004).

Na Figura 1, observa-se um cenário onde o este trabalho se aplica. Uma rede, que inicialmente pode ter sido toda conexa, agora é composta por três partições desconexas. Uma mensagem, que tem como origem o nó S em uma partição e destino D, que se encontra em outra partição, é carregada fisicamente por um VANT, que a transporta de sua origem até o destino.

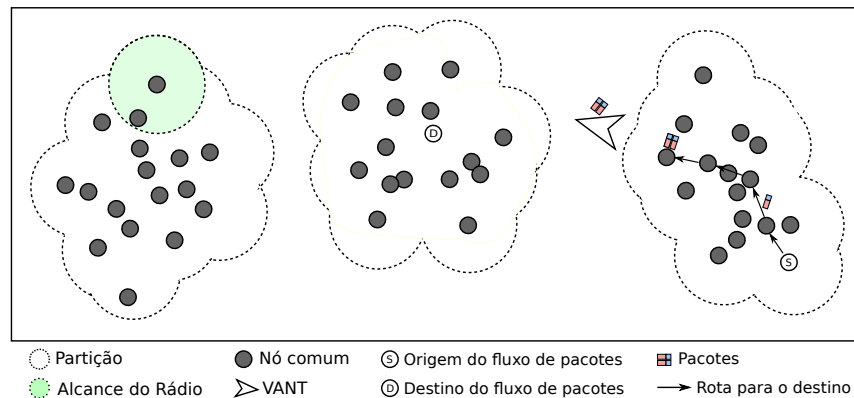


Figura 1 Exemplo de uma rede formada por três partições. Um VANT é utilizado para transportar os pacotes entre as partições.

1.2 Definição do problema

Este trabalho trata do restabelecimento da comunicação em uma RSSF formada por diversas partições desconexas, utilizando VANTs como

meio para transportar as mensagens entre partições distintas.

Para tanto, considera-se uma rede modelada como um grafo $G = (V, A)$, em que V é um conjunto de nós e uma aresta $\{u, v\} \in A$, se e somente se há um *link* de comunicação entre os nós $u \in V$ e $v \in V$.

Há n partições na rede, contidas no conjunto $P = \{p_1, p_2, \dots, p_n\}$. Uma partição é um subconjunto de vértices, ou seja, $p_i = \{v \in V \mid \forall u \in p_i \text{ com } u \neq v, \exists L^{u,v}\}$. $L^{u,v}$ é um caminho conexo entre u e v , definido como $L^{u,v} = \{\{v, x_1\}, \{x_1, x_2\}, \dots, \{x_{g-1}, x_g\}, \{x_g, u\}\}$, para $g \in \mathbb{N}$ e $L^{u,v} \subset A$. Este trabalho trata particularmente do caso onde $n \geq 2$.

Uma propriedade inerente à rede é dada por: $\forall u, \forall v$, com $u \in p_i$, $v \in p_j$ e $i \neq j$, $\nexists L^{u,v}$. Essa propriedade expressa a desconexão entre as partições.

Assim, o problema abordado por este trabalho envolve o restabelecimento da comunicação entre nós em diferentes partições, ou seja, $\forall u \in p_i, \forall v \in p_j, i \neq j$, onde $i, j \in \mathbb{N}_{>0}$ e $i, j \leq n$, uma forma de u e v se comunicarem deve ser provida.

1.3 Justificativa

Algumas aplicações de RSSFs, como, por exemplo, monitoramento de áreas de risco, podem ter sua maior utilidade em momentos em que o meio externo pode causar danos a rede. Um exemplo desse cenário seria a situação na qual um deslizamento de terra, em uma área monitorada, danifica alguns nós sensores, formando partições desconexas na rede.

As desconexões podem afetar a qualidade do serviço oferecido pela rede a ponto de inviabilizar a sua utilização e/ou manutenção. Dessa forma, é necessário postergar ao máximo a ocorrência dessas desconexões, além de

oferecer meios para que partições desconexas da rede possam se comunicar independentemente da existência de um enlace entre elas.

Assim, o uso de VANTs nesse contexto é um objeto de estudo justificado, sendo possível que, com sua utilização, redes que executam tarefas críticas possam continuar a operar mesmo em situações extremas.

1.4 Objetivos

Esta seção descreve o objetivo geral deste trabalho assim como também o distribui em um conjunto de objetivos específicos a serem seguidos na elaboração da metodologia e na obtenção e avaliação de resultados.

1.4.1 Objetivo geral

Este trabalho visa ao desenvolvimento e avaliação de métodos para o restabelecimento da comunicação entre partições desconexas de uma RSSF utilizando VANTs para transportar as mensagens, considerando limitações na capacidade de armazenamento do nós e dos VANTs e também padrões de tráfego da rede.

1.4.2 Objetivos específicos

Com o propósito de atingir o objetivo geral, foram determinados os seguintes objetivos específicos:

- Adaptação de uma ferramenta para simulação de RSSFs desconexas.
- Desenvolvimento do método com o qual os VANTs detectarão a presença de partições desconexas na rede.

- Escolha e implementação de um método para entrega de pacotes dentro de uma mesma partição.
- Desenvolvimento e avaliação de um método em que um único VANT será responsável por cobrir toda a área da rede, sem considerar a vazão dos pacotes enviados.
- Aprimoramento do método anterior, a fim de utilizar os padrões de tráfego da rede na determinação de rotas para um VANT.
- Desenvolvimento e avaliação de um método em que são utilizados múltiplos VANTs para cobrir a área da rede.

1.5 Contribuições

Este trabalho apresenta contribuições relevantes no desenvolvimento e análise de modelos de movimento utilizados para conectar RSSFs particionadas. Essas contribuições foram publicadas em eventos científicos de ampla divulgação:

- Modelo de movimento, baseado na versão euclidiana do problema do caixeiro viajante, utilizando um único VANT para reconectar as partições (HEIMFARTH; ARAUJO, 2014).
- Modelo de movimento para um único VANT e que considera o tráfego presente na rede ao determinar o melhor trajeto (HEIMFARTH; ARAUJO; GIACOMIN, 2014).
- Modelo de movimento para múltiplos VANTs, sendo cada um responsável por um conjunto exclusivo de partições (região). Um ponto de

troca comum é utilizado pelos VANTs para troca de mensagens entre regiões (HEIMFARTH et al., 2014c).

Outras contribuições no desenvolvimento de técnicas para RSSFs e redes oportunistas:

- Avaliação de um *framework* que utiliza programação genética para a geração automática de programas para RSSFs (HEIMFARTH et al., 2014a).
- Desenvolvimento de um protocolo *cross-layer* para controle de acesso ao meio em que é utilizado um *backbone* de nós sensores como via rápida para transporte de mensagens em direção ao nó *sink* (HEIMFARTH et al., 2014b).
- Desenvolvimento e análise matemática de um protocolo *cross-layer anycast* que combina funcionalidades de roteamento geográfico juntamente com as de um protocolo de acesso ao meio baseado em amostragem de preâmbulos (HEIMFARTH; GIACOMIN; ARAUJO, 2015).
- Um protocolo para roteamento oportunista baseado em modelo de movimento sociológico (OLIVEIRA et al., 2015).

1.6 Estrutura do trabalho

O restante deste documento é organizado da seguinte forma: o Capítulo 2 corresponde a um referencial teórico sobre alguns tópicos que serão utilizados no desenvolvimento deste trabalho e um estudo sobre seu estado da arte. O Capítulo 3 traz o cenário do problema, uma visão geral da metodologia desenvolvida e suas etapas. Seguindo a metodologia, um conjunto

de modelos de movimento para VANTs é apresentado no Capítulo 4 e, os resultados de simulações realizadas para esses modelos são apresentados e discutidos no Capítulo 5. O Capítulo 6 conclui o trabalho, também enumerando possíveis futuras contribuições.

2 Referencial teórico

Este capítulo apresenta uma revisão bibliográfica sobre os tópicos envolvidos neste trabalho, assim como um estudo sobre seu estado da arte.

2.1 Redes de sensores sem fio

RSSFs consistem em dispositivos eletrônicos (nós sensores) individuais, capazes de interagir com o ambiente onde estão inseridas. Essa interação pode ser feita por meio de um conjunto de sensores que aferem parâmetros físicos do ambiente (temperatura, pressão, umidade, acelerômetros, entre outros) ou utilizando atuadores para modificá-los (KARL; WILLIG, 2005).

Um nó sensor é composto por uma unidade central de processamento, memórias para armazenamento de programa e também para executá-lo, sensores (e talvez atuadores), um módulo para comunicação sem fio e uma fonte de alimentação. Como a alimentação é normalmente realizada com baterias, o consumo energético é um dos pontos críticos para esta tecnologia.

Em geral, esses nós devem colaborar entre si com o propósito de cumprir uma missão específica, visto que um único nó pode não ser capaz de realizá-la. Para isso, eles utilizam um enlace sem fio para comunicar-se. A comunicação entre os nós independe de uma infraestrutura, implicando na necessidade de protocolos que se autoconfigurem e possam se adaptar a mudanças que a rede sofre, seja por falhas em alguns nós ou crescimento da rede.

A aplicação para essas redes envolvem detecção de eventos, *tracking* de objetos, leituras periódicas de variáveis do ambiente, entre outras (LOUREIRO et al., 2003). Muitas dessas aplicações envolvem am-

bientes críticos, como áreas de risco, e devem ter seu funcionamento mantido mesmo na ocorrência de falhas. A capacidade de autogerenciamento e adaptação a mudanças inesperadas em um sistema é o tópico abordado pela computação autônoma (ZHAO; GAO; DUAN, 2009). Outra classe de aplicações está imersa no conceito de computação ubíqua, onde as aplicações realizam sua computação de forma imperceptível, transparente ao usuário (HANSMANN, 2003).

2.2 Veículos aéreos não tripulados

Na definição do United States of America (2005), VANTs são veículos aéreos que não transportam operadores humanos e são capazes de voar autonomamente ou podem ser pilotados de forma remota. Por não carregarem pilotos, essas aeronaves estão sujeitas a variados tipos de aplicações em que a presença humana implicaria em restrições. Exemplos disso compreendem áreas contaminadas por produtos nocivos à saúde humana. Essas aplicações, segundo Valavanis (2007), têm tido um crescimento expressivo tanto nas áreas civis como militares, podendo ser empregadas também na manutenção da conectividade de redes tolerantes a falhas.

Os modelos de VANTs variam em tamanho, formato, configurações e propósito. Existem VANTs de tamanho similar ao de insetos até aqueles com proporções similares a de aeronaves comerciais (BONE; BOLKCOM, 2003).

Não considerando apenas aspectos como segurança, mas também de extensão, VANTs de vários tamanhos e modelos podem ser utilizados no monitoramento de áreas de desastres ambientais, detecção de eventos, manutenção da conectividade em redes esparsas, entre outras aplicações.

2.3 Roteamento geográfico

Na definição de Kurose e Ross (2012), roteamento trata do processo, no âmbito geral da rede, que determina os caminhos fim-a-fim que os pacotes percorrem desde a sua origem até o destino. Em geral, esses protocolos são baseados em algoritmos de vetor de distância ou de estado de enlace (*Bellman-Ford* e *Dijkstra*, por exemplo). Entretanto, para redes *ad hoc*, em que não há uma infraestrutura centralizadora, existem outros requisitos. Os protocolos, além de serem distribuídos, devem ser capazes de se adaptar a mudanças na topologia da rede.

Dentre as diversas classes em que se dividem tais protocolos de roteamento para redes *ad hoc*, estão aqueles que utilizam posições para endereçar os pacotes. O destino de um pacote, ao ser roteado, pode ser por exemplo, “qualquer dispositivo em uma dada região” ou “o dispositivo mais próximo da posição especificada”.

Para o encaminhamento dos pacotes, em um roteamento geográfico, existem diferentes estratégias, descritas em Karl e Willig (2005):

Menor distância em relação ao destino: assume-se que todos os nós da rede conhecem suas próprias posições e, também, a de seus vizinhos. Um pacote, ao ser enviado, é endereçado com a posição de seu destino. Em uma abordagem gulosa, o pacote é sempre encaminhado para o vizinho mais próximo do destino, minimizando a distância restante que o pacote deve percorrer.

Roteamento direcional: os pacotes são encaminhados para o vizinho que mais se aproxima da direção do destino. Em um exemplo disso, o vizinho escolhido como próximo salto pode ser aquele que está posicionado

mais próximo de uma linha reta entre origem e destino do pacote.

Apesar dessas estratégias, existe um problema que ocorre especialmente em redes pouco densas. Mesmo a rede sendo conexa, pode não haver um caminho completo para o destino onde todos os nós se enquadrem na estratégia utilizada. Esse problema é conhecido como *dead end*. Dessa forma, é necessário que o algoritmo de roteamento retroceda (em distância ou direção), até que encontre uma rota completa para o pacote. Uma implementação que trata esse problema é apresentada por Karp e Kung (2000), em um algoritmo chamado GPSR *Greedy Perimeter Stateless Routing*.

2.4 Trabalhos relacionados

A conectividade é algo crucial em RSSFs e, para mantê-la, diversas abordagens têm sido propostas para tratar do problema causado pelas desconexões. Algumas delas são baseadas em redundância, em que nós adicionais (*relays*) são utilizados com o propósito de manter a conectividade da rede. Um exemplo é dado por Konstantinidis e Yang (2011), que traz um algoritmo evolucionário multiobjetivo que é utilizado para calcular o número necessário de nós a serem inseridos de forma a garantir a k -conectividade da rede. Os autores propõem que, por meio do método desenvolvido, cada nó deve suportar $k - 1$ falhas.

Na proposta de Lee e Younis (2010), a área da rede é modelada como um *grid* que contém células de tamanhos iguais. Com isso, o problema consiste em determinar a menor quantidade de células que devem ser populadas com *relays*, de forma a garantir que as partições da rede possam se comunicar. Como esse problema está na classe NP-Difícil, os autores desenvolvem uma heurística chamada *Cell-based Optimized Relay Node Placement* para

resolvê-lo. Da mesma forma, Li e Hou (2005) abordam o problema que envolve a incorporação da menor quantidade possível de nós adicionais dentro do limite da rede de forma a conectar uma rede inicialmente particionada. O artigo traz um algoritmo baseado em uma Triangulação de *Delaunay*.

Embora a maior parte das propostas de redundância tente minimizar a quantidade de *relays*, essa tendência é contraposta por Kimence e Bekmezci (2014). Nesse trabalho, é utilizada uma função de otimização que tem como parâmetros os pesos associados aos pontos da área e resulta em um conjunto de posições onde os *relays* devem ser instalados, de modo que o peso total seja mínimo e a rede torne-se totalmente conexa. Segundo os autores, os pesos podem representar o custo financeiro da instalação de um retransmissor em um dado ponto ou os perigos associados, como, por exemplo, o risco de deslizamento de terra em uma região montanhosa, o que danificaria o nó.

Assim, abordagens que usam redundância dependem da instalação desses *relays* em pontos ideais, o que pode não ser viável em situações de emergência, seja por restrições físicas, técnicas ou financeiras. No entanto, existem outras propostas. Uma delas apresenta o uso de robôs móveis como retransmissores, controlando o deslocamento desses robôs para o local apropriado e os utilizando como pontes para comunicar regiões desconexas (ATAY; BAYAZIT, 2009). O emprego de múltiplos coletores de dados capazes de entregar dados para a estação base é apresentado em Al-Turjman, Hassanein e Ibnkahla (2011). O posicionamento de tais coletores é calculado com uma abordagem baseada em *grid* e programação linear inteira mista. O objetivo é otimizar o reposicionamento periódico desses *sinks* móveis, utilizando um conjunto relativamente pequeno de possíveis

posições. Esse protocolo, no entanto, tem utilização limitada para aplicações que, embora sejam distribuídas, não necessitem de colaboração de toda a rede. Um exemplo seriam aplicações de monitoramento em que os dados obtidos podem ser entregues a múltiplos coletores, sem a necessidade de interação entre partes desconexas da rede.

Também, por meio de mobilidade, Azzaz, Saidane e Minet (2011) tratam do uso de um pequeno número de robôs móveis em uma rede estática para repor os nós danificados. O trabalho prediz o esgotamento de energia dos nós e, com isso, as falhas podem ser reparadas antes mesmo de acontecerem. A movimentação dos robôs é discutida em duas estratégias, ambas centralizadas em um nó coordenador, o que insere um ponto de falha no sistema.

O uso de mobilidade controlada para a manutenção da conectividade dos nós de uma rede pode ser realizada utilizando Veículos Aéreos Não Tripulados (VANTs) como nós móveis, conforme apresentado em Hunjet e Hui (2011), Llorca, Milner e Davis (2007) e Marinho et al. (2013). Em Llorca, Milner e Davis (2007), um conjunto de VANTs mantém conexão direta entre si, formando um *backbone* sobre as partições. A comunicação entre as partições em solo é realizada utilizando esse *backbone* de VANTs. Hunjet e Hui (2011) utilizam micro-VANTs, para repor nós estáticos defeituosos, como forma de restabelecer a comunicação entre regiões desconexas. O trabalho apresenta métodos para o cálculo distribuído de posicionamento dos VANTs, de forma a restaurar a conectividade da rede. Já em Marinho et al. (2013), os autores sugerem o uso de um grupo de VANTs e técnicas de *Multiple-Input Multiple-Output* (MIMO) para o aumentar o raio de comunicação de partições isoladas.

Outros protocolos utilizam o conceito de *message ferrying* (ZHAO; AMMAR; ZEGURA, 2004), no qual uma mensagem é transportada fisicamente de sua partição de origem até a de destino por nós que possuem mobilidade controlada. Joshi e Younis (2014) propõem um método baseado nesse conceito onde é utilizado um número limitado de VANTs para reconectar partições formadas por nós estáticos. Para minimizar as rotas de cada VANT e distribuir a carga igualmente entre eles, uma abordagem baseada em árvore geradora mínima é utilizada. E, para realizar a troca de mensagens entre os VANTs, o ponto escolhido é dado pelo centro de massa de um polígono em que cada vértice representa o segmento de cobertura de um VANT. Embora o trabalho proponha uma forma de reconectar diversas partições, ele não considera limitações reais da rede, como velocidade de voo dos VANTs e *buffer* dos nós, o que impacta na latência e também na taxa de entrega.

Diferente das outras abordagens, este trabalho utiliza um número limitado de VANTs para restabelecer a comunicação entre partições desconexas de uma RSSF. Os nós móveis são usados para transportar fisicamente a mensagem entre partições, ao invés de apenas funcionarem como retransmissores, permitindo que sejam tratados cenários com partições distantes umas das outras. E, os modelos de movimento propostos consideram limitações reais impostas pelos cenários, como *buffers* e tráfego na rede. Uma análise detalhada do impacto dessas variáveis na manutenção da conectividade também é apresentada neste trabalho.

3 Metodologia

Dado o cenário em que há o interesse de se restabelecer a comunicação em uma RSSF, formada por partições desconexas, este capítulo apresenta uma visão geral da metodologia proposta, bem como as etapas que a compõem.

3.1 Visão geral

O método desenvolvido utiliza um conjunto de VANTs e o conceito de *message ferrying* para restabelecer comunicação entre partições desconexas de uma RSSF. Essa proposta foi apresentada na Figura 1.

Em uma rede particionada, cada partição elege um nó especial que atuará como *cluster head*, sendo responsável por interagir com o VANT. O *cluster head* deve armazenar os pacotes de sua partição que possuem como destino outras partições. O processo de escolha do nó *cluster head* será apresentado na etapa denominada ***Eleição do Cluster Head***.

Os VANTs possuem uma lista das posições dos *cluster heads* e, para construí-la, é realizada uma ronda de reconhecimento na área da rede. Esse processo é descrito em ***Descoberta de Cluster Heads***.

Dentro de uma mesma partição, é utilizado um algoritmo de roteamento geográfico simplificado (sem tratamento de *dead ends*) para rotear os pacotes aos seus destinos. O não tratamento dos *dead ends* deve-se ao fato de que a desconexão entre duas partições seria tratada como um *dead end* pelo algoritmo. Entretanto, se o destinatário dos pacotes está em outra partição, esses pacotes são encaminhados até o *cluster head* da partição de origem utilizando o mesmo algoritmo de roteamento. No *cluster head* esses pacotes são armazenados dentro de um *buffer*, respeitando sua capacidade

máxima. Essa fase é denominada ***Roteamento Intrapartição***.

A comunicação entre as partições é realizada utilizando um VANT. Periodicamente um VANT percorre os *cluster heads*, coletando os pacotes que estão em seus *buffers* e entrega-os ao *cluster head* da partição de destino. Esse processo está descrito na etapa do projeto chamada ***Roteamento Interpartições*** e também envolve os modelos de movimento propostos para os VANTs.

Com base no princípio de que a rede inicialmente não estava particionada, qualquer nó poderia se comunicar com outro. Conforme ocorrem falhas e a rede se torna desconexa, a comunicação entre nós em partições distintas depende dos mecanismos discutidos anteriormente. Para simular essa situação, as trocas de pacotes entre nós de duas partições distintas são chamadas de fluxos. Cada fluxo é determinado entre dois nós escolhidos aleatoriamente no início de cada simulação. Um deles é a origem do pacote e o outro, o destino, e cada fluxo é disparado periodicamente.

A Figura 2 exibe uma visão geral dessas etapas. Na Figura 2(a), o processo de ***Eleição do Cluster Head*** é apresentado. Uma rede inicialmente conexa, agora é formada por duas partições desconexas. Com isso, para cada partição, é eleito um nó que passará a agir como o *cluster head*. A Figura 2(b) traz a situação onde o nó denominado S envia pacotes para o nó D. Nessa situação, o processo de ***Roteamento Intrapartição*** ocorre, encaminhando os pacotes para o seu destino, utilizando roteamento geográfico, até que os pacotes atinjam uma extremidade da partição. Como é impossível entregar os pacotes ao seu destino, por ele estar em uma partição distinta, os pacotes são roteados, também, por roteamento geográfico, para o *cluster head*, onde aguardarão para serem transportados pelo VANT.

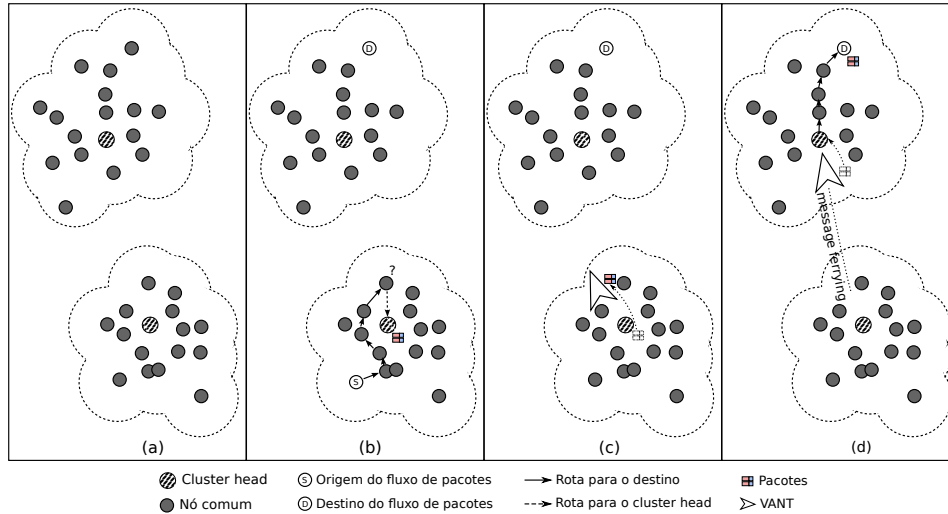


Figura 2 Visão geral das etapas envolvidas no desenvolvimento do trabalho.

Periodicamente, o VANT inicia um processo de negociação com os *cluster heads*, como parte do **Roteamento Interpartições**. Os pacotes que são endereçados a outras partições são transferidos do *buffer* do *cluster head* para o *buffer* do VANT, para que sejam transportados, como mostra a Figura 2(c). Por fim, a Figura 2(d) traz o transporte das mensagens, por meio do modelo de *message ferrying*, até o *cluster head* da partição de destino. Quando o VANT sobrevoa o *cluster head* de uma partição, os pacotes que estão contidos no *buffer* do VANT e que são endereçados à partição atual são entregues ao *cluster head* que os encaminha para os seus destinos finais.

Para a apresentação dos algoritmos propostos nesta metodologia, é utilizada uma notação orientada a eventos, que consiste na declaração e inicialização de um conjunto de variáveis, sendo seguida por um conjunto de ações. Uma ação é composta por um *trigger*, ou seja, o evento que desencadeia a execução do código da ação. p representa o nó atual e outras

letras representam outros nós. $Neigh_p$ denota o conjunto de nós no raio de alcance do rádio de p . $send \langle msg \rangle to dest$ é o comando que envia uma mensagem $\langle msg \rangle$ para o nó $dest$. De forma análoga, o comando $receive \langle msg \rangle from q$ é a função para receber mensagens. O mesmo código é executado em todos os nós da rede.

Nas seções a seguir estão descritas em detalhes as etapas da metodologia supracitada, bem como o método para a geração de RSSFs desconexas para uso nas simulações.

3.2 Formação de RSSFs desconexas

Para avaliar os métodos de restabelecimento de conexão, é necessário que sejam simuladas RSSFs desconexas. Cada rede é composta por n partições as quais partições não possuem interseções, como definido na Seção 1.2.

Considera-se que cada partição possui um raio máximo r_p e que cada nó sensor possui alcance máximo de r_r . Com o propósito de avaliar os modelos de movimento para VANTs, as partições devem ser distantes umas das outras. Essa distância deve ser suficientemente grande, de forma a garantir que um VANT estacionário no ponto médio entre duas partições quaisquer não possa servir como uma ponte entre essas partições.

Assim, $\forall p_i, \forall p_j$, com $i \neq j$, a distância entre os centros de p_i e p_j é dada por $d_{i,j} > 2 \times r_p + \alpha \times r_r$, onde $\alpha \geq 2$. Respeitando esse critério, cada partição tem seu centro determinado aleatoriamente dentro da área de simulação. Adicionalmente, as partições possuem a mesma quantidade de nós cada.

A Figura 3 mostra uma área onde há uma rede desconexa formada

por três partições. O raio dos nós situados nas extremidades de duas partições distintas indica a distância entre elas. Dessa forma, um VANT posicionado entre duas partições quaisquer não poderia servir como uma ponte entre elas.

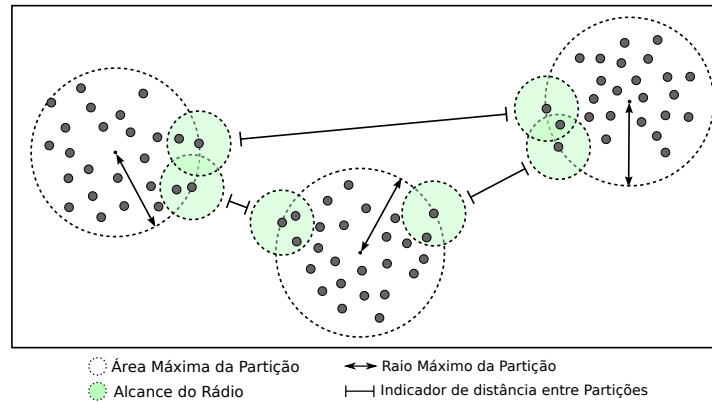


Figura 3 Visualização do critério de formação de RSSFs desconexas.

3.3 Eleição do cluster head

Esta etapa é responsável por escolher um nó sensor dentro de cada partição para ser responsável por trocar dados com o VANT. Isso é necessário para restringir a trajetória e interação do VANT com apenas um nó por partição. Se fosse permitida a comunicação direta entre o VANT e qualquer nó, seria necessário mapear toda a área da partição, de forma a assegurar que cada nó pudesse trocar pacotes com nós em outras partições.

A eleição do nó que atuará como *cluster head* é dada pela capacidade de cada nó de se comunicar com os demais. Quanto mais vizinhos um nó possuir, maior será a probabilidade que ele se torne *cluster head*. Isso é justificado pelo fato de que todos os pacotes endereçados a partições externas devem ser transferidos para o *cluster head*.

O Algoritmo 1 mostra o processo de eleição. Seu propósito é, ao detectar uma desconexão na rede, informar todos os nós dentro da partição sobre essa situação e eleger um nó que será o *cluster head*.

O processo tem início quando uma desconexão é detectada. Isso é feito por uma modificação feita no algoritmo de roteamento geográfico. Quando não é possível enviar um pacote para um ponto adiante em direção ao destino, assume-se que a rede está desconexa e o destino do pacote é outra partição. Nesse momento, o Algoritmo 1 inicia o procedimento de escolha do *cluster head*, como pode ser observado entre as linhas 5 e 8. A função CHECKSTATE (linhas 41-49) verifica se o estado do nó ainda é *conectado* (o que significa que a desconexão ainda não havia sido detectada). Nesse caso, o nó assume o estado de *desconectado*, envia para todos os seus nós vizinhos uma mensagem de desconexão e inicia o processo de eleição. Um nó, ao receber essa mensagem de desconexão, também executa a função citada (linhas 9-12).

A função responsável pela eleição é chamada STARTELECTION e utiliza dois temporizadores. *ElectionTimer* é o tempo com o qual o nó tenta se estabelecer como *cluster head*. Esse tempo é dado pela equação (linha 37) que leva em consideração a quantidade de vizinhos que o nó possui e o grau de conectividade (λ) da rede. Um exemplo do processo de eleição é dado na Figura 4, onde o grau de conectividade da rede é 30. O temporizador difere em cada nó, dependendo da sua quantidade de vizinhos. O temporizador expirará antes no nó que possuir a maior quantidade de vizinhos. O primeiro nó que tiver seu temporizador encerrado é eleito como *cluster head*.

Algoritmo 1 Eleição do *Cluster Head*

```

1: nodeType: enumeration [ordinary, clusterHead] init ordinary
2: state: enumeration [connected, disconnected] init connected
3: waitingTime: real
4: electionExecuting: boolean init false
5: { Disconnection has been detected }
6: Begin
7:   CHECKSTATE()
8: End
9: { Message with <Disconnection> has arrived }
10: Begin
11:   CHECKSTATE()
12: End
13: { ElectionTimer has been expired }
14: Begin
15:   if electionExecuting = true then
16:     nodeType ← clusterHead
17:     for all q ∈ Neighp do
18:       send <EndElection> to q
19:     end for
20:   end if
21: End
22: { Message with <EndElection> has arrived }
23: Begin
24:   if electionExecuting = true then
25:     electionExecuting ← false
26:     for all q ∈ Neighp do
27:       send <EndElection> to q
28:     end for
29:   end if
30: End
31: { ClearTimer has been expired }
32: Begin
33:   state ← connected
34: End
35: function STARTELECTION
36:   electionExecuting ← true
37:   waitingTime ←  $\frac{\lambda}{|Neigh_p|} + random(3)$ 
38:   start ElectionTimer with time waitingTime
39:   start ClearTimer with time 120
40: end function
41: function CHECKSTATE
42:   if state = connected then
43:     state ← disconnected
44:     for all q ∈ Neighp do
45:       send <Disconnection> to q
46:     end for
47:     STARTELECTION()
48:   end if
49: end function

```

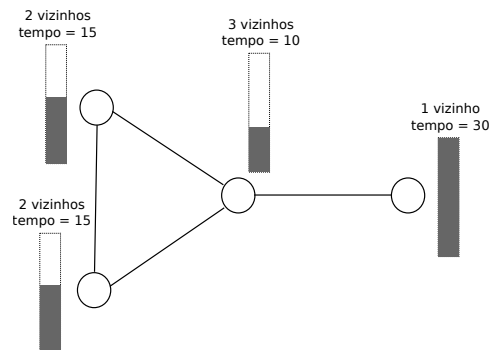


Figura 4 Exemplo de uso de temporizadores para a eleição do *cluster head*.

Após um nó se transformar em *cluster head*, ele envia em *broadcast* uma mensagem para todos os outros nós da partição, como forma de finalizar o processo de eleição (linhas 13-21). Essa mensagem de término de eleição é disseminada a todos os nós da rede (linhas 22-30). Já *ClearTimer* é o tempo para o nó assumir o estado de *conectado* novamente, após o início da eleição (linha 31-34).

3.4 Descoberta de cluster heads

Um VANT possui dois modos de operação: encaminhamento e busca. O primeiro trata do percurso realizado por um VANT para recolher e entregar pacotes entre *cluster heads* distintos. Já no *modo de busca*, os VANTs sobrevoam a área monitorada procurando por *cluster heads* e inclui suas posições em uma lista que será percorrida durante o *modo de encaminhamento*. Durante esse processo de busca, os VANTs percorrem toda a área monitorada, enviando pacotes de reconhecimento (*beacons*). Um *cluster head*, ao receber tal mensagem, notifica o VANT sobre sua presença.

O mecanismo como um VANT percorre a área monitorada, quando em *modo de busca* é longitudinal, em intervalos que utilizam o raio de alcance

dos rádios tanto do VANT como dos nós que estão na superfície. Dessa forma, evitam-se áreas de sombra.

3.5 Roteamento intrapartição

Esta etapa é proposta para determinar as rotas dos pacotes internamente dentro de uma partição, sendo o seu código inserido na camada de rede de cada nó sensor. A Figura 5 contém parte do cabeçalho de um pacote transferido na rede. Os campos *Origem* e *Destino* são o remetente e destinatário final do pacote, respectivamente. O campo chamado *NextHop* armazena a próxima posição para onde o pacote irá, em caminho ao destino final. Já *Estado* contém o *status* atual do pacote e pode ser *Intra*, *Inter* ou *InterVANT*. *Intra* diz que o pacote está sendo roteado internamente dentro da partição. *Inter* é o estado que o pacote assume quando é percebido que este deve ir para outra partição e, *InterVANT* é a situação em que o pacote está pronto para ser transportado pelo VANT ou já está sendo transportado.

Origem	Destino	NextHop	Estado	...	Dados
= {Intra, Inter, InterVANT}					

Figura 5 Parte do cabeçalho de um pacote transmitido na rede.

O Algoritmo 2 descreve esse processo de roteamento dentro de uma partição. Quando um pacote chega para ser roteado, é verificado se o campo *Destino* dele é o nó atual (linhas 5-8). Se isso for verdade, o processo de roteamento é finalizado e o pacote é entregue para a camada superior. A próxima condição verifica se o pacote está no *cluster head* e se seu estado é *Inter*. Sendo verdadeiro, isso diz que o pacote deve ser armazenado no *buffer* do *cluster head* para que possa ser encaminhado pelo VANT (linhas 9-13).

Algoritmo 2 Roteamento Intrapartição

```

1: nextDestination: node
2: nextHop: node
3: { Packet <packet> has been received }
4: Begin
5:   if  $p = \langle packet \rangle . Destination$  then
6:     Send <packet> to upper layer
7:     return
8:   end if
9:   if  $p = clusterHead$  and  $\langle packet \rangle . State = Inter$  then
10:     $\langle packet \rangle . State \leftarrow InterUAV$ 
11:    STOREINBUFFER(<packet>)
12:    return
13:   end if
14:   if  $p = clusterHead$  and  $\langle packet \rangle . State = InterUAV$  then
15:     $\langle packet \rangle . State \leftarrow Intra$ 
16:   end if
17:   if  $\langle packet \rangle . State = Intra$  then
18:    nextDestination  $\leftarrow \langle packet \rangle . Destination$ 
19:    nextHop  $\leftarrow GEOROUTING(nextDestination)$ 
20:    if nextHop = NULL then
21:       $\langle packet \rangle . State \leftarrow Inter$ 
22:    end if
23:   end if
24:   if  $\langle packet \rangle . State = Inter$  then
25:    nextDestination  $\leftarrow clusterHead$ 
26:    nextHop  $\leftarrow GEOROUTING(nextDestination)$ 
27:   end if
28:   send <packet> to nextHop
29: End

```

Se o pacote está sendo encaminhado a partir do VANT, como apresentado na linha 14, ele deve ser roteado pelo *cluster head* como um pacote *Intra*, pois ele já está na sua partição de destino. Já o próximo teste (linha 17) é responsável por tratar de pacotes que estão em roteamento interno na partição. Na linha 19, a função `GEOROUTING` é chamada, sendo responsável por utilizar um algoritmo de roteamento geográfico baseado no GPSR (KARP; KUNG, 2000) para decidir qual o próximo salto do pacote. Caso esse algoritmo falhe, é assumido que o pacote deve ser levado para outra partição pelo VANT. O estado do pacote é alterado para *Inter* (linha 21) e ele é roteado para o *cluster head*. Esse processo é realizado no trecho

que compreende as linhas 24 a 27. Por último, após decidido qual o próximo salto do pacote (em direção ao destino dentro da partição ou se ele deve ser enviado para o *cluster head*), ele é enviado para o próximo salto.

É importante ressaltar que a função `STOREINBUFFER` armazena o pacote no *buffer* do *cluster head* apenas se houver espaço disponível. Caso contrário, o pacote é descartado.

3.6 Roteamento interpartições

O roteamento interpartições é o processo pelo qual um pacote é transportado de sua partição de origem até a partição destino. Esse processo corresponde ao *modo de encaminhamento* citado na Seção 3.4. Inicialmente é apresentado o procedimento de interação para troca de pacotes entre *cluster heads* e VANTs. Em sequência, no Capítulo 4, estão descritos os modelos de movimento desenvolvidos para os VANTs percorrerem os *cluster heads*. Como o trajeto do VANT (ou múltiplos VANTs) pode influenciar na latência e na taxa de entrega dos pacotes, diferentes modelos são propostos.

3.6.1 Interação entre cluster head e VANT

Sempre que um VANT atinge o raio de alcance de um *cluster head*, inicia-se uma negociação visando à troca de pacotes entre VANT e *cluster head*. O VANT envia ao *cluster head* os pacotes que estão em seu *buffer* e se destinam à partição em questão. Analogamente, o *cluster head* entrega para o VANT os pacotes que estão em seu *buffer* e se destinam a outras partições. Esse procedimento é realizado de acordo com as seguintes etapas:

1. O VANT envia todos os pacotes que estão em seu *buffer* e que são

destinados à partição atual, para que sejam encaminhados aos seus destinos.

2. O VANT envia um pacote de requisição para o *cluster head*, contendo a quantidade livre de *buffer* que possui.
3. Ao receber essa requisição, o *cluster head*, caso tenha pacotes destinados a outras partições, envia-os para o VANT. A quantidade de pacotes enviada é limitada pelo espaço livre em *buffer* informado pelo VANT e. O *buffer* do *cluster head* utiliza uma política *First In, First Out* (FIFO).

4 Modelos de movimento

Este capítulo apresenta os três modelos de movimento para VANTs propostos nos objetivos específicos deste trabalho. Esses modelos fazem parte do *Roteamento Interpartições*, introduzido na Seção 3.6, sendo eles responsáveis por determinar a distribuição das partições entre os VANTs presentes no cenário, bem como a ordem e frequência em que essas partições são visitadas.

4.1 ETSP

Um modelo inicial propõe o uso de um único VANT por rede e não considera o tráfego entre as partições para determinar a melhor rota. Apesar de sua simplicidade, este modelo é base para o desenvolvimento de outros que explorem mais características da rede.

Cada partição da rede desconexa é representada pelo seu *cluster head*, pois a única interação entre os nós sensores e o VANT é realizada por este nó especial. Assim, o VANT, quando em *modo de encaminhamento*, deve visitar, periodicamente, cada *cluster head* com o propósito de entregar os pacotes destinados à partição atual e, também, para coletar os pacotes endereçados a outras partições. Como o tempo do percurso entre as partições pode impactar na taxa de entrega e latência dos pacotes, a rota é determinada pela solução da versão euclideana do problema do caixeiro viajante (*Euclidean Traveling Salesperson Problem* ou ETSP, do inglês) para as posições dos *cluster heads*. Com esse procedimento, assegura-se o menor caminho possível para a realização de um ciclo entre as partições.

A partir disso e com base na definição do problema deste trabalho apresentada na Seção 1.2, o problema de otimização para o cálculo da rota

é definido como (HROMKOVIC, 2004):

Entrada: um grafo completo ponderado não dirigido (G', c) , onde $G' = (V', A')$ e $V' = \{CH(p_1), CH(p_2), \dots, CH(p_n)\}$ é o conjunto de todos os *cluster heads*. $CH: P \rightarrow V$ é a função que mapeia uma partição para o seu *cluster head*. $c: A' \rightarrow \mathbb{R}_{>0}$ representa a distância euclideana entre quaisquer dois *cluster heads*.

Restrições: para qualquer instância de (G', c) , $\eta(G', c) = \{v_{i_1}, v_{i_2}, \dots, v_{i_n}, v_{i_1} \mid (i_1, i_2, \dots, i_n) \text{ é qualquer permutação de } (1, 2, \dots, n) \text{ e } v \in V'\}$, ou seja, o conjunto dos ciclos hamiltonianos para G' .

Custos: para qualquer ciclo hamiltoniano $H' = \{v_{i_1}, v_{i_2}, \dots, v_{i_n}, v_{i_1}\} \in \eta(G', c)$, $\text{custo}((v_{i_1}, v_{i_2}, \dots, v_{i_n}, v_{i_1}), (G', c)) = \sum_{j=1}^n c(\{v_{i_j}, v_{i_{(j \bmod n)+1}}\})$, ou seja, o custo de qualquer ciclo hamiltoniano H' é a soma dos pesos de todas as arestas de H' .

Objetivo: Custo mínimo.

Uma implementação, utilizando *branch-and-bound*, do ETSP foi utilizada para encontrar a rota ótima para o VANT quando em *modo de encaminhamento*.

A Figura 6 apresenta uma rota para uma rede formada por cinco partições desconexas. O VANT, ao começar sua operação, inicia o trajeto pelo *cluster head* mais próximo de sua posição de partida. A partir disso, são realizados ciclos passando por todos os *cluster heads*, respeitando o menor ciclo possível encontrado no cálculo do ETSP.

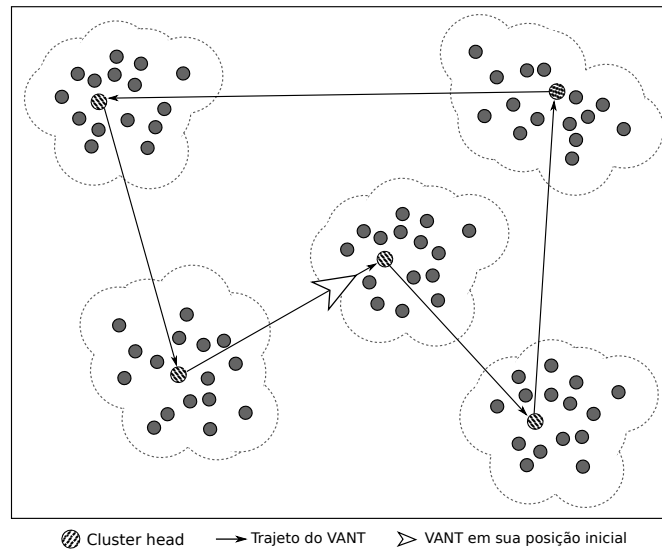


Figura 6 Exemplo de ciclo realizado por um VANT em cinco partições desconexas, utilizando o modelo ETSP.

4.2 Weighted

Uma extensão do modelo apresentado na Seção 4.1 é a utilização de um único VANT, embora considerando o tráfego existente entre as partições para determinar a ordem e frequência de visitação do VANT aos *cluster heads*.

Dessa forma, este modelo, denominado *Weighted*, propõe que o ciclo do VANT quando em *modo de encaminhamento* é composto de m rodadas. Na rodada inicial, todos os *cluster heads* são visitados, entregando os pacotes cujo destino são a partição atual e coletando aqueles endereçados a outras partições. Além desta tarefa, nessa primeira rodada é realizada uma coleta de estatísticas do tráfego de entrada e saída de cada partição. Dessa forma, é possível construir um grafo de comunicação em que cada vértice representa uma partição. $C = (P, E', w)$ é o grafo dirigido de comunicação. (P, E') é um

grafo dirigido em que P é o conjunto de partições e uma aresta $e = (p_i, p_k)$ pertence a E' se e somente se existir algum tráfego saindo de p_i com destino a p_k . w é uma função de E para \mathbb{R} que descreve a quantidade de tráfego ($\frac{kb}{s}$) entre duas partições, sendo $w(p_i, p_k)$ a quantidade de tráfego saindo de p_i com destino a p_k .

Com o propósito de reduzir o tempo das rodadas subsequentes, que podem impactar na latência e taxa de perda da rede, utiliza-se uma priorização para aquelas partições que possuem maior tráfego. Para isso, a seguinte metodologia é aplicada:

Seja $O : P \rightarrow \mathbb{R}$ uma função que mapeie uma dada partição para a sua quantidade total de tráfego de saída: $O(p_i) = \sum_{j=1}^n w(p_i, p_j)$.

Seja $I : P \rightarrow \mathbb{R}$ uma função que mapeie uma dada partição para a sua quantidade total de tráfego de entrada : $I(p_i) = \sum_{j=1}^n w(p_j, p_i)$.

Seja $T : P \rightarrow \mathbb{R}$ uma função que mapeie uma dada partição para a sua quantidade total de tráfego de entrada/saída: $T(p_i) = O(p_i) + I(p_i)$.

p_{max} é a partição com a maior quantidade total de tráfego, enquanto p_{min} é aquela que possui a menor quantidade total de tráfego.

Seja $N : P \rightarrow \mathbb{R}$ uma função que mapeie uma dada partição para a normalização da sua quantidade total de tráfego: $N(p_i) = \frac{T(p_i)}{T(p_{max})}$.

O primeiro passo para determinar o ciclo do VANT é definir o número total de rodadas que ele será composto. $h = \text{round}(\frac{1}{N(p_{min})})$ é o número necessário de rodadas para realizar a comunicação com base em priorização. $m = h + 1$ é o número total de rodadas, incluindo a primeira rodada completa utilizada para coletar estatísticas de tráfego.

A partir desse ponto, uma tabela de visitas com m linhas e n colunas é construída. As linhas representam as m rodadas e as colunas são as

partições. b_{ij} , onde $0 \leq i < m$ e $0 \leq j < n$, representa a posição localizada na linha i e coluna j . Um *zero* na célula indica que, naquela rodada i , a partição p_j não é visitada. Caso contrário, o valor da célula será um .

Esta tabela é preenchida de acordo com o seguinte procedimento:

- Todas as células da primeira linha são sempre preenchidas com números um . Esta é a rodada inicial onde informações de tráfego são coletadas.
- Para $1 \leq i < m$, $b_{ix} = 1$, onde $x = p_{max}$. Esta etapa determina que em todas as rodadas a partição com maior tráfego deverá ser visitada.
- Com isso, a tabela é preenchida considerando todas as demais partições, em ordem decrescente de $N(p_g)$. Para cada partição p_g considerada, o número de linhas em que ela deve aparecer (excetuada a primeira) é dado por $k_g = N(p_g) \times h$. O intervalo em linhas entre cada ocorrência de visita à partição p_g é dado por $sp_g = \frac{h}{k_g}$. Adicionalmente, a primeira rodada (linha) em que a partição será visitada é aquela com menos alocações, ou seja, é menor valor de $\sum_{l=0}^{n-1} b_{il}$, para $i = 0, 1, \dots, m$. Assim, garante-se uma distribuição uniforme das partições a serem visitadas em cada uma das m rodadas.

Para ilustrar a construção dessa tabela, pode-se considerar o seguinte exemplo: uma rede com 4 partições desconexas ($P = \{p_1, p_2, p_3, p_4\}$) e assumindo a seguinte função T :

$$T(d) = \begin{cases} \frac{1}{5} & \text{se } d = p_1 \\ \frac{1}{10} & \text{se } d = p_2 \\ \frac{1}{2} & \text{se } d = p_3 \\ \frac{1}{5} & \text{se } d = p_4 \end{cases} \quad d \in P$$

Resultando na função normalizada:

$$N(d) = \begin{cases} \frac{2}{5} & \text{se } d = p_1 \\ \frac{1}{5} & \text{se } d = p_2 \\ 1 & \text{se } d = p_3 \\ \frac{2}{5} & \text{se } d = p_4 \end{cases} \quad d \in P$$

O número de rodadas é $\frac{1}{N(p_2)} + 1$, i.e., 6. A tabela de visitas resultante é apresentada na Tabela 1. A primeira linha contém todas as partições. p_3 é a partição com maior tráfego, portanto é marcada para visitaç o em todas as rodadas (linhas). p_1 e p_4 devem aparecer em duas rodadas cada e p_2   a partição com menor tráfego, devendo portanto aparecer em apenas uma rodada.

Com esse procedimento, a tabela contendo todas as partições que devem ser visitadas em cada rodada   constru da. Partições com tráfego mais intenso t m uma maior frequ ncia de visitaç o, ou seja, t m prioridade superior  quelas que possuem menor tráfego.   importante ressaltar que a tabela apresenta quais partições devem ser visitadas em cada rodada, mas n o determina a ordem e visitaç o. Para definir essa ordem, uma adaptaç o do m todo proposto na Seç o 4.1   utilizado. Cada rodada no *Weighted* ter  sua ordem de visitaç o definida utilizando-se o ETSP, garantindo assim o

Tabela 1 Possível tabela de visitas para quatro partições desconexas.

Visitas por rodada				
<i>Rodadas</i>	<i>Partições</i>			
	p_1	p_2	p_3	p_4
ro_1	1	1	1	1
ro_2	1	0	1	0
ro_3	0	0	1	1
ro_4	1	0	1	0
ro_5	0	0	1	1
ro_6	0	1	1	0

menor caminho possível entre todas as partições presentes na rodada. Para a primeira rodada, a primeira partição a ser visitada é aquela que se encontra mais próxima do VANT. Nas demais rodadas, a partição que estiver mais próxima da última do caminho anterior (e que não for igual) é escolhida como a primeira.

4.3 Genetic

Um modelo mais geral para movimentação de VANTs é apresentado nesta seção. Esse modelo trata não apenas da utilização de um ou mais VANTs, mas também da distribuição das partições entre os VANTs e a forma de comunicação entre eles. Inicialmente é apresentada uma descrição formal para o modelo com base em um problema conhecido na literatura, seguindo do método proposto para solucioná-lo.

4.3.1 Problema de otimização interpartições

Aqui é definido formalmente o problema de roteamento interpartições utilizando múltiplos VANTs. Essa definição deriva do *Vehicle Problem*

with *Pickups and Deliveries* (VRPPD), que por sua vez vem da classe geral dos *Pickup and Delivery Problems* (PDPs), segundo Berbeglia, Cordeau e Laporte (2010). Esse problemas tratam do roteamento de veículos em que objetos ou pessoas têm de ser transportados de uma origem até um destino.

O VRPPD está na subclasse dos problemas “um para um”. Nesses problemas, cada item tem uma origem e um destino. No caso do problema deste trabalho, os itens são as mensagens e origem/destino são os *cluster heads* das diversas partições desconexas da rede. No VRPPD, uma frota de veículos é utilizada e eles servem a mais de uma requisição por vez, o que significa que cada VANT da frota deve ter uma dada capacidade de transporte (*buffer*). As coletas e entregas são pareadas, ou seja, pacotes vindo de um dado *cluster head* devem ser transportados para uma dada partição de destino.

A formulação matemática do problema de comunicação interpartições é uma generalização do Problema do Caixeiro Viajante (*Traveling Salesperson Problema* ou TSP, do inglês) (BEKTAS, 2006). Assim, o modelo proposto nesta seção é uma derivação daquele apresentado por Berbeglia, Cordeau e Laporte (2010):

n ... número de *cluster heads* de coleta.

\tilde{n} ... número de *cluster heads* de entrega ($n = \tilde{n}$).

U ... conjunto dos *cluster heads* de coleta.

D ... conjunto dos *cluster heads* de entrega.

K ... conjunto dos VANTs.

q_i ... entrada/saída no vértice i ; vértices de coleta - valor positivo, vértice de entrega - valor negativo.

c_{ij} ... custo para cruzar o arco (i, j) com um VANT. No caso deste trabalho, o custo é a distância.

C^k ... capacidade de transporte do VANT k , em pacotes.

Considerando que o problema abordado use vértices de coleta e entrega pareados ($n = \tilde{n}$), refere-se a tal par de vértices como uma requisição, indexada por $i = 1, \dots, n$, sendo i a origem ou vértice de coleta (um *cluster head*) e $n+i$ o vértice (outro *cluster head*) de entrega correspondente.

Para definir o modelo proposto nesta seção, é necessário realizar um mapeamento entre as partições apresentadas na Seção 1.2 e os conjuntos de *cluster heads* de coleta e entrega. Considerando que *cluster head* é o ponto de comunicação entre partições, define-se o conjunto dos *cluster heads* $H = \{h_1, h_2, \dots, h_n\}$, onde h_i é o *cluster head* da partição p_i . Pode-se também mapear o fluxo de mensagens como $U = \{i \in H \mid i \text{ tem pacotes aguardando transmissão}\}$ e $D = \{j \in H \mid j \text{ é o destino de algum vértice em } U\}$. É importante destacar que um mesmo vértice no grafo de *cluster heads* apresentado na Seção 1.2 pode ser representado por dois vértices no PDP (quando ele possui pacotes para enviar e receber). Esse grafo de PDP é modelado como $PDP_G = (W, A)$, onde W é o conjunto de todos vértices de coleta e entrega, juntamente com a posição inicial e final dos VANTs $W = U \cup D \cup \{0, n + \tilde{n} + 1\}$ e, A é o conjunto de todos os arcos. Também, como já apresentado, c_{ij}^k é o custo (distância) entre os vértices i e j utilizando o VANT k .

A Figura 7 traz um exemplo desse mapeamento. Na figura, o *cluster head* h_1 tem um pacote para enviar para h_2 , portanto é representado por um vértice no conjunto U . O *cluster head* h_2 é o destino final do pacote e, ao mesmo tempo, tem um pacote para ser enviado para o *cluster head* h_3 . Portanto, h_2 é representado em ambos os conjuntos U e D do PDP. Como h_3 é apenas destino, ele é representado no conjunto D , enquanto que como

h_4 não apresenta nenhum tráfego, ele não é mapeado para nenhum conjunto do PDP.

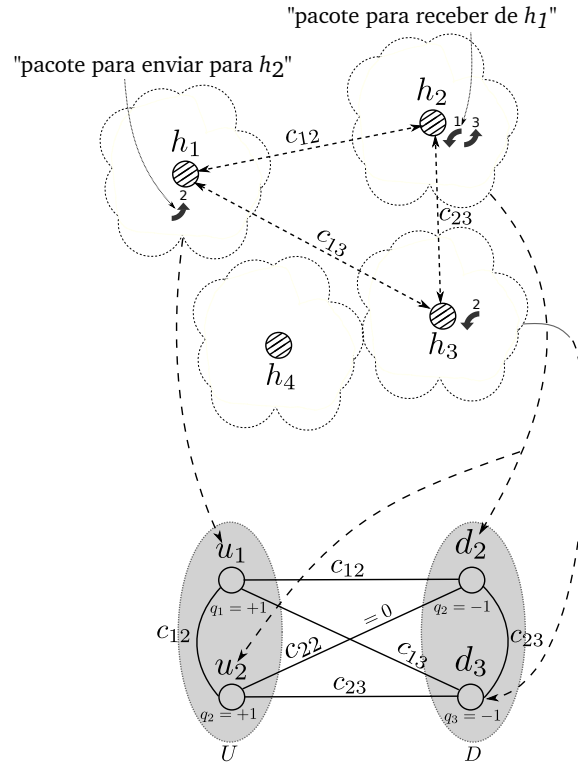


Figura 7 Exemplo de mapeamento entre um conjunto de *cluster heads* e o grafo do PDP.

As variáveis de decisão são:

$Q_i^k \dots$ carga do VANT k quando deixando o vértice i

$B_i^k \dots$ início do serviço do VANT k no vértice i

$$x_{ij}^k = \begin{cases} 1 & \text{se o arco } (i, j) \text{ é visitado pelo VANT } k \\ 0 & \text{caso contrário} \end{cases}$$

Minimização do caminho total viajado:

$$\text{minimizar } \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \quad (1)$$

Minimização da latência total:

$$\text{minimizar } \sum_{k \in K} \sum_{i \in W} B_{n+i}^k - B_i^k \quad (2)$$

sujeito a:

$$\sum_{k \in K} \sum_{j: (i,j) \in A} x_{ij}^k = 1 \quad \forall i \in U \cup D \quad (3)$$

$$\sum_{j: (0,j) \in A} x_{0,j}^k = 1 \quad \forall k \in K \quad (4)$$

$$\sum_{i: (i, n+\bar{n}+1) \in A} x_{i, n+\bar{n}+1}^k = 1 \quad \forall k \in K \quad (5)$$

$$\sum_{i: (i,j) \in A} x_{ij}^k - \sum_{i: (j,i) \in A} x_{ji}^k = 0 \quad \forall j \in U \cup D, k \in K \quad (6)$$

$$x_{ij}^k = 1 \Rightarrow B_j^k \geq B_i^k + c_{ij} \quad \forall (i,j) \in A, k \in K \quad (7)$$

$$x_{ij}^k = 1 \Rightarrow Q_j^k \geq Q_i^k + q_j \quad \forall i \in V, k \in K \quad (8)$$

$$Q_i^k \geq \max\{0, q_i\} \quad \forall i \in V, k \in K \quad (9)$$

$$Q_i^k \leq \min\{C^k, C^k + q_i\} \quad \forall i \in V, k \in K \quad (10)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i,j) \in A, k \in K \quad (11)$$

$$\sum_{j: (i,j) \in A} x_{ij}^k - \sum_{j: (n+i,j) \in A} x_{n+i,j}^k = 0 \quad \forall i \in W, k \in K \quad (12)$$

$$B_i^k \leq B_{n+i}^k \quad \forall i \in W, k \in K \quad (13)$$

A Equação 1 é a função objetivo que visa à minimização da distância total percorrida (reduzir o trajeto total que os VANTs devem percorrer). Já a Equação 2 é uma função objetivo alternativa. Ela visa à minimização da latência total de cada entrega. A restrição 3 diz que cada vértice tem que ser visitado exatamente uma vez, enquanto as igualdades 4 e 5 garantem que cada veículo comece e termine seu percurso em um ponto comum a todos os VANTs (usualmente, o centro da área). Nem todos os VANTs precisam ser usados, um veículo pode usar apenas o arco $(0, n + \tilde{n} + 1)$, ou seja, ele não sai do ponto de encontro entre os VANTs. A conservação dos fluxos de mensagens é garantida pela equação 6 (uma mensagem sempre terá origem e destino), enquanto a Equação 7 garante a sequência correta do tempo de serviço para percorrer um arco. A Equação 8 representa a quantidade de carga que cada VANT possui ao sair de um vértice. As restrições 9 e 10 garantem que a capacidade (*buffer*) de um veículo não vai ser excedida. A Equação 12 determina que tanto origem como destino de uma requisição sejam servidas pelo mesmo VANT e, a entrega de um pacote só pode ocorrer após a sua coleta, o que é representado pela Equação 13.

O modelo apresentado nesta seção representa um modelo exato cuja solução representa a rota ótima que cada VANT deve executar para minimizar a função objetivo. Por outro lado, na proposta deste trabalho, não é possível utilizar esta formulação. O principal motivo deve-se ao fato de que a frota de VANTs não possui uma visão completa das requisições o tempo todo. Apenas com o auxílio de uma espécie de “oráculo” com capacidade de conhecer todas as requisições de forma dinâmica, seria possível resolver essa formulação a fim de obter o resultado ótimo. Nesse caso, cada nova requisição desencadearia o recálculo da rota ótima para todos os veículos

da frota e, como consequência, as rotas para mínima latência e distância percorrida poderiam ser obtidas.

No caso real do modelo proposto, as requisições apenas são descobertas quando o veículo de fato visita um determinado *cluster head*. Portanto, mesmo uma partição que não possua qualquer requisição de coleta/entrega conhecida deve ser periodicamente visitada, como forma de determinar seu novo estado (se há ou não novas requisições). Por esse motivo, o modelo matemático introduzido nesta seção é adaptado e apresentado na Seção 4.3.2 de forma que as partições são divididas em regiões distintas e cada região é responsabilidade de um único VANT.

4.3.2 Modelo de movimento para múltiplos VANTs

Na Seção 4.3.1, foi apresentada uma formulação matemática mais abrangente para o problema de otimização abordado neste trabalho. No entanto, essa formulação não é aplicável em um cenário real pois ela depende de uma visão global de todas as requisições que ocorrem na rede em todos os momentos.

Assim, o método proposto gera uma solução aproximada utilizando estatísticas de tráfego coletadas pelos próprios VANTs para determinar as rotas. E, como o VRPPD é *NP-Difícil* (BERBEGLIA; CORDEAU; LAPORTE, 2010), o método utiliza algoritmos genéticos (HOLLAND, 1975) para se obter soluções. Nessa técnica, um conjunto de indivíduos (potenciais soluções) formam uma população que é evoluída por várias gerações seguindo os princípios da seleção natural darwiniana, em que o indivíduo mais apto tende a sobreviver, ou seja, a melhor solução encontrada para um dado problema deve ser escolhida como a ótima.

Um indivíduo (cromossomo) na proposta deste modelo, compreende um conjunto de genes, sendo que cada gene representa uma região. Uma região, por sua vez, é uma lista de *cluster heads* que um VANT deve visitar periodicamente. Não existe interseção entre os elementos de diferentes genes, devido ao fato de que as regiões são isoladas umas das outras. A Figura 8 traz a representação de um cromossomo. Na parte superior da figura está a representação do genótipo para 6 pontos (*cluster heads*) distribuídos em 2 regiões. Cada região é definida pelo elementos de um gene (representadas na figura pelos identificadores dos *cluster heads*). Um VANT é atribuído por região e ele viaja ciclicamente visitando todos os *cluster heads* da região, respeitando a ordem imposta pelo gene correspondente. Por exemplo, na figura, $Gene_1$ diz ao VANT para viajar em ciclos do *cluster head* 4 para o 6, seguindo depois para o 1 e então reiniciando o ciclo no *cluster head* 4.

O número de genes em um cromossomo é igual à quantidade de VANTs presentes no sistemas e, a distribuição dos *cluster heads* entre os genes é dada de acordo com as seguintes restrições e função de minimização.

Um cromossomo pode ser visto como uma matriz

$$C_m = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

em que cada linha representa um gene, ou seja, o ciclo de um VANT, respeitando as seguintes restrições:

Um gene não pode ter elementos repetidos: $a_{ij} \neq a_{ik}, \forall j \neq k$ com $i = [1, m]$ e $j, k = [1, n]$.

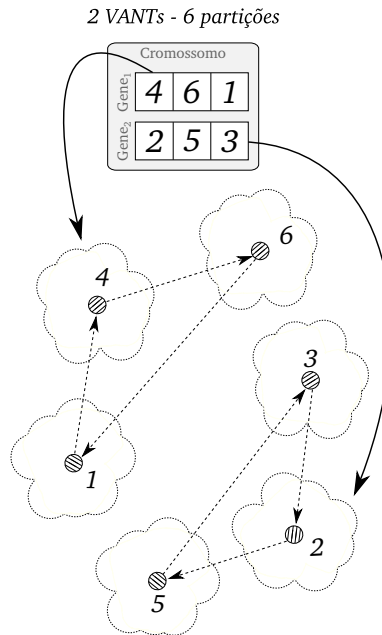


Figura 8 Representação de um cromossomo utilizado pelo algoritmo genético.

Não existe interseção entre elementos de diferentes genes:

$$\bigcup_{j=1}^n a_{ij} \cap \left[\bigcup_{k=1}^m a_{kj} - \bigcup_{j=1}^n a_{ij} \right] = \emptyset, \forall i.$$

Essas duas restrições dão uma visão geral de como a distribuição dos *cluster heads* em regiões deve ser realizada. Assim, uma função de minimização é empregada para selecionar a solução mais apta para o problema, com respeito a essas restrições. Essa função de aptidão (*fitness*) leva em conta o tamanho do ciclo formado pelo elementos de um gene (região) e o total de comunicação existente dentro dessa região, ou seja, o número de pacotes que são coletados e entregues em *cluster heads* dentro dessa mesma região.

Para calcular a aptidão de um cromossomo são necessárias duas outras funções:

$Dist : H \times H \rightarrow \mathbb{R}$, onde $Dist$ é a distância euclidiana entre dois *cluster heads*.

$PD : H \times H \rightarrow \mathbb{N}$, onde PD retorna o número de itens (pacotes) que serão trocados entre dois *cluster heads*, ou seja, pacotes que são coletados em um e entregues no outro.

O objetivo do modelo proposto nesta seção é minimizar a soma de todos os ciclos, embora ao mesmo tempo, o sistema deva tentar manter partições com grande comunicação entre si em uma mesma região. A Equação 14 formaliza esse procedimento, pois ela pondera como ciclo utilizando o inverso de seu tráfego interno: uma região com intenso tráfego interno tende a possuir um menor coeficiente. No entanto, o denominador utilizado possui um comportamento logarítmico, devido ao fato de que concentrar tráfego em uma região é uma melhoria no método, mas não é mandatório. Como fora empiricamente observado, concentrar tanto quanto possível o tráfego em uma única região tende a criar ciclos muito grandes, portanto um comportamento logarítmico para um dado resultado de PD é desejável.

$$\text{minimizar } \sum_{i=1}^m \sum_{j=1}^n \frac{\text{Dist}(a_{ij}, a_{i(j \bmod n)+1})}{1 + \ln(PD(a_{ij}, a_{i(j \bmod n)+1}) + 1)} \quad (14)$$

Com o propósito de executar o processo evolucionário proposto pelos algoritmos genéticos, os indivíduos são avaliados pela função *fitness* a cada geração. A cada geração, alguns indivíduos são selecionados para cruzamento (*crossover*) por um processo chamado torneio. Na técnica do torneio, um número fixo de indivíduos da geração atual é aleatoriamente selecionado e os dois indivíduos mais aptos são selecionados para *crossover*. Um operador de *crossover* é então aplicado a esses indivíduos. A Figura 9 mostra o processo de *crossover* utilizado no modelo proposto: um ponto é selecionado como ponto de corte de ambos os pais. Inicialmente, Pai_1 é inteiramente re-

plicado para o novo indivíduo. Em seguida, o segmento seguinte ao ponto de corte do Pai_2 é replicado para a mesma posição do novo indivíduo, de acordo com um procedimento específico. Cada elemento da segunda parte do genótipo do Pai_2 ocasiona uma operação de troca no novo indivíduo, a fim de evitar interseção entre os genes, respeitando assim as restrições apresentadas anteriormente. Por exemplo, para inserir o elemento 8 no novo indivíduo, uma troca entre os elementos 5 e 8 deve ser feita para evitar “oitos” repetidos.

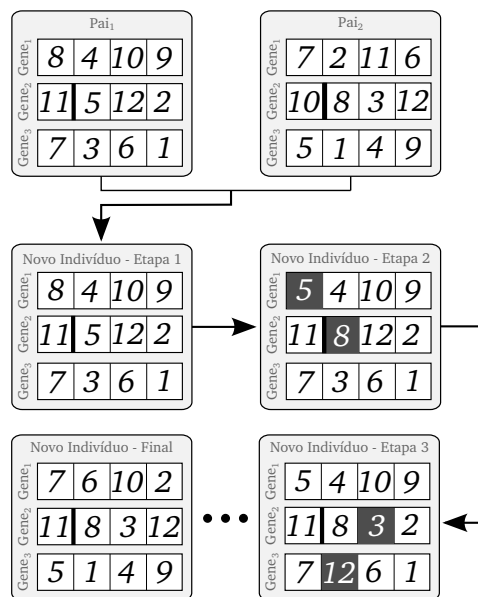


Figura 9 Representação do operador de *crossover*.

Subsequentemente, um operador de mutação é utilizado visando ao aumento da variabilidade genética da população. A Figura 10 exemplifica o processo empregado no modelo. Quando operador é aplicado (dado uma probabilidade), duas posições randômicas em um cromossomo são selecionadas e trocadas entre si. Na figura, os elementos 4 e 7 são trocados.

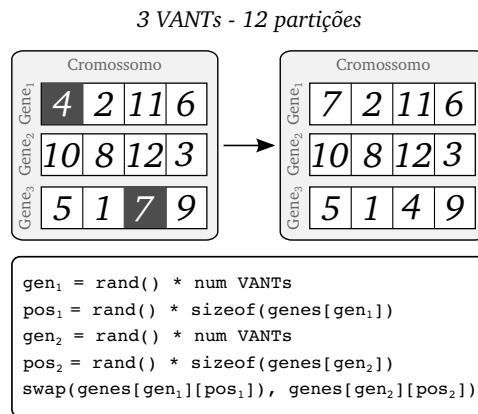


Figura 10 Representação do operador de mutação.

Além da técnica para evoluir as rotas para cada um dos VANTs, é necessário considerar o fato de que uma região deve ser capaz de se comunicar com as outras. Para tanto, propõe-se que o centro de massa compreendendo todas as regiões seja usado como ponto de encontro para os VANTs. Esse ponto é calculado considerando as coordenadas geográficas de cada *cluster head* ponderadas pelo seu tráfego total (entrada/saída). Isso garante que o VANT responsável pela região com maior tráfego realizará o menor desvio de sua rota (ciclo), tendendo a diminuir nessa região a taxa de perda devido à saturação de *buffers* ocasionada pela sua comunicação interna.

A Figura 11 apresenta uma plotagem das rotas criadas com o resultado da execução do algoritmo genético com a função *fitness* proposta para um conjunto de indivíduos durante várias gerações. O ponto de encontro também foi calculado conforme o procedimento descrito, sendo representado pelo símbolo \times na figura. O exemplo utilizou 21 partições e 7 VANTs, resultando em 7 regiões, cada uma com 3 partições. É notável a sobreposição entre duas regiões próximas ao ponto de encontro dos VANTs. Esse fenômeno ocorre pois o método tenta não apenas minimizar os ciclos, mas

também concentrar o tráfego dentro das regiões, como descrito na função *fitness*.

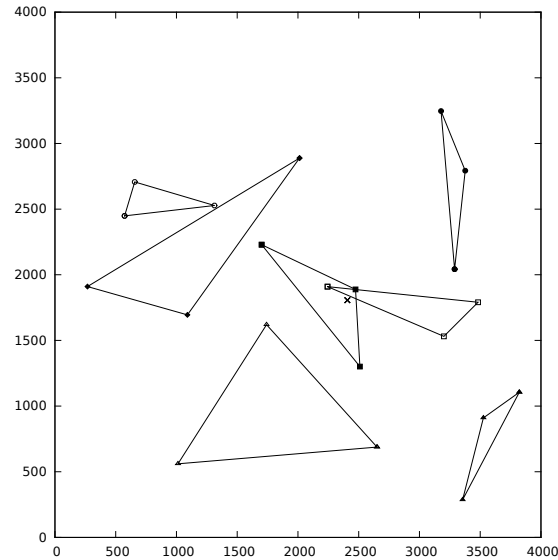


Figura 11 Exemplo das regiões geradas usando o método baseado em algoritmo genético. O ponto de encontro é representado por \times .

4.3.3 Interação entre VANTs

Periodicamente, no ponto de encontro, os VANTs se comunicam uns com os outros, com o propósito de trocar pacotes endereçados a regiões diferentes. Nesta seção esse processo é descrito.

Todas as unidades da frota estabelecem o próximo momento de encontro com base no VANT com o maior ciclo, sendo assim possível assegurar que cada VANT visitará pelo menos uma vez cada partição de sua região antes de se deslocar para o ponto de encontro. Após visitar um *cluster head*, cada VANT determina se é o momento de alterar seu comportamento de visitação de partições para ir ao ponto de encontro. Para tomar essa decisão,

o VANT avalia se ao visitar a próxima partição irá atrasar sua chegada ao ponto de encontro no momento marcado. Caso esse atraso possa ocorrer, o VANT ao invés de visitar a partição se desloca para o ponto de encontro, garantindo assim que chegue lá dentro do prazo limite fixado.

Com esse procedimento, garante-se que todos os VANTs se encontrarão no ponto de encontro determinado no máximo no instante estabelecido. Durante esse encontro, são realizados dois processos. O primeiro é determinar a nova alocação dos VANTs para o novo cálculo de rotas, considerando que as requisições para roteamento de mensagens podem ter mudado com o tempo. Nessa etapa, cada veículo transmite em *broadcast* as estatísticas de tráfego que coletou durante as visitas às partições de sua região. Uma vez que essas informações forem disseminadas, um VANT previamente selecionado calcula se são necessárias alterações nas rotas. Se esse for o caso, é necessário recalcular todo o PDP utilizando o algoritmo genético apresentado na Seção 4.3.2.

O segundo processo envolve a troca de pacotes endereçados a diferentes regiões. Nessa etapa, cada VANT difunde o estado de seu *buffer*, ou seja, quantas posições estão atualmente livres em seu *buffer* e quantos pacotes são endereçados para partições de cada uma das outras regiões. Com base nessa informação, os *buffers* são ordenados por ordem de espaço livre. O VANT com mais espaço livre será o primeiro a receber pacotes dos outros. Após isso, o estado dos *buffers* é atualizado e e reordenado. Esse processo continua até que não seja mais possível trocar nenhum pacote. É importante ressaltar que alguns pacotes endereçados a regiões diferentes podem não ser trocados devido a saturação de *buffers*. Assim, esses pacotes são descartados e o sistema trata isso como uma falha de transmissão.

5 Resultados e discussões

Este capítulo apresenta os resultados obtidos a partir de simulações para os modelos de movimento propostos no Capítulo 4. Cada um desses modelos foi avaliado com base em diferentes configurações para *buffers*, fluxos de mensagens na rede e número de partições.

As simulações foram realizadas utilizando o simulador GrubiX, uma extensão do simulador de redes *ad hoc* ShoX (LESSMANN; HEIMFARTH; JANACIK, 2008). O modelo de rádio utilizado baseia-se no IEEE 802.11b, com potência fixa de transmissão e assumindo o modelo de Friis para propagação isotrópica. Foi considerado um meio de transmissão livre de colisões e um roteamento geográfico livre de *dead ends* nas partições.

Cada partição da rede é distante das demais no mínimo mais que o dobro do alcance do rádio. Essa distância deve-se a ideia de forçar o uso de um VANT para transportar fisicamente o pacote e não apenas como uma ponte fixa entre partições. Além disso, as partições utilizadas nas simulações possuem 60 nós sensores cada, com raio de alcance de 50m. Os VANTs possuem esse mesmo raio de alcance e têm a velocidade de voo de $60 \frac{km}{h}$.

E, para se obter resultados estatisticamente relevantes, cada cenário de cada modelo de movimento foi simulado 50 vezes e, o nível de confiança para o cálculo do intervalo de confiança dos resultados é $\gamma = 1 - \alpha$, $\alpha = 0,05$.

5.1 ETSP

Esta seção traz as configurações de simulação e resultados para um modelo baseado no ETSP, conforme apresentado na Seção 4.1. Embora se trate de um modelo simples, ele é base para outros e, sua avaliação também

corroborar sua eficiência para alguns cenários.

5.1.1 Configuração das simulações

Para avaliar o modelo, foram propostos diferentes cenários, que têm suas configurações apresentadas na Tabela 2. Um fluxo é definido como sendo o envio periódico de um pacote com origem em uma partição e destino em outra. Para este modelo, os fluxos foram gerados seguindo uma distribuição uniforme. Além disso, as trocas de mensagens dentro de uma mesma partição não são consideradas.

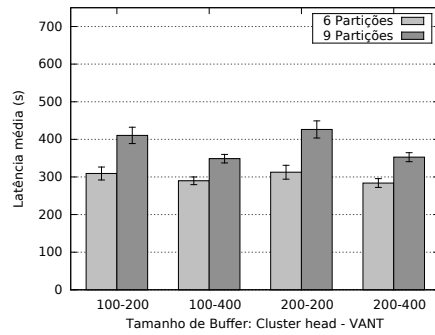
Tabela 2 Parâmetros de configuração dos cenários simulados no modelo ETSP

Parâmetros de Simulação	
<i>Parâmetro</i>	<i>Valor</i>
Área de simulação (km^2)	2×2 e 3×3
Número de partições	6 e 9
Número de VANTs	1
Número de fluxos – NF	10 e 20
Pacotes por fluxo – PF	100
Vazão de cada fluxo ($\frac{pacotes}{s}$)	0,1
Número de pacotes enviados	NF \times PF
Tamanho de <i>buffer</i> para <i>cluster head</i> (<i>pacotes</i>)	100, 200
Tamanho de <i>buffer</i> para VANT (<i>pacotes</i>)	200, 400

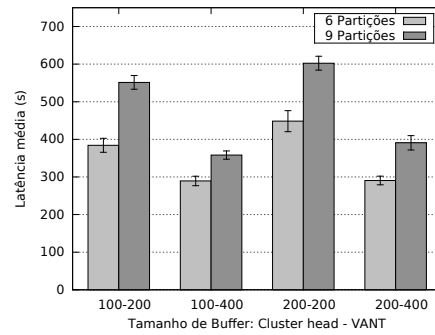
5.1.2 Resultados

A Figura 12 apresenta os gráficos para latência média de entrega de pacotes para duas áreas de simulação ($4km^2$ e $9km^2$) e com números diferentes de fluxos (dez e vinte). Um pacote de cada fluxo é enviado entre duas

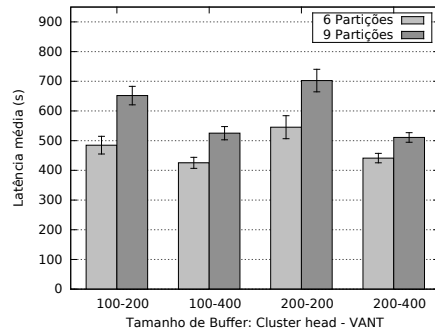
partições diferentes a cada dez segundos. Em cada gráfico, diferentes configurações de partições e *buffers* para *cluster heads* e VANT são utilizadas.



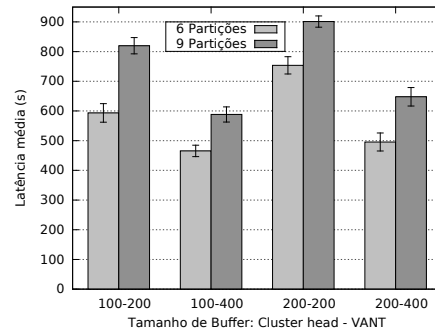
(a) 10 fluxos de dados, $4km^2$ de área.



(b) 20 fluxos de dados, $4km^2$ de área.



(c) 10 fluxos de dados, $9km^2$ de área.



(d) 20 fluxos de dados, $9km^2$ de área.

Figura 12 Latência média para entrega dos pacotes no modelo ETSP.

Comparando os resultados, nota-se que o aumento da latência é proporcional ao aumento da área de simulação. Isso deve-se ao fato de que o percurso que o VANT deve realizar é aumentado. Uma ronda completa do ETSP, em média, aumenta proporcionalmente ao aumento da área monitorada. Também é possível concluir a partir dessas simulações que o número de partições influencia na latência. Apesar de uma mesma área, quanto maior a quantidade de partições, maior é o movimento que o VANT deve

realizar para percorrer um ciclo do ETSP.

Também é possível inferir que as diferentes combinações de tamanho de *buffers* para *cluster heads* e VANT têm um papel importante na latência. Quanto maior o *buffer* do VANT em relação ao dos *cluster heads*, menor é a latência dos pacotes. Isso é especialmente verdadeiro para o cenário com 9 partições e 20 fluxos e pode ser explicado da seguinte maneira: quando o *buffer* do VANT tem tamanho similar ao do *cluster head*, não é possível carregar todos os pacotes que aguardam por transporte nos *cluster heads* em uma única ronda, portanto alguns pacotes têm de aguardar uma próxima ronda.

Já a Figura 13 exhibe os gráficos para o percentual de pacotes entregues para as duas áreas simuladas ($4km^2$ e $9km^2$) e com duas quantidades de fluxos (dez e vinte). Os fluxos de pacotes entre as partições e os tamanhos de *buffers* foram configurados da mesma forma descrita anteriormente. É importante notar, que para dez fluxos, o número total de pacotes enviados é mil, enquanto que para vinte fluxos, dois mil pacotes são enviados durante a simulação.

A Figura 13(a) representa o cenário com menor estresse para o sistema proposto. O ciclo do VANT é menor devido à menor área ($4km^2$) e baixo número de fluxos (dez). É possível notar que quase todos os pacotes são entregues aos seus destinos finais, excetuando-se aqueles que são descartados quando o *buffer* de seu *cluster head* está cheio. Isso pode ser observado em cenários onde os tamanhos de *buffers* são reduzidos.

Conforme o número de pacotes transmitidos por segundo aumenta (vinte fluxos), como visto na Figura 13(b), a rede começa a descartar mais pacotes, embora ainda entregue a maior parte deles. No gráfico, é possível

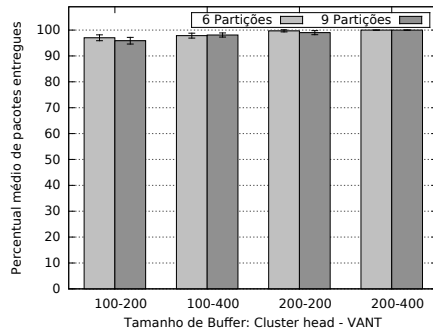
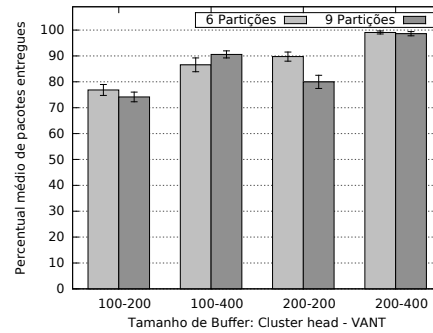
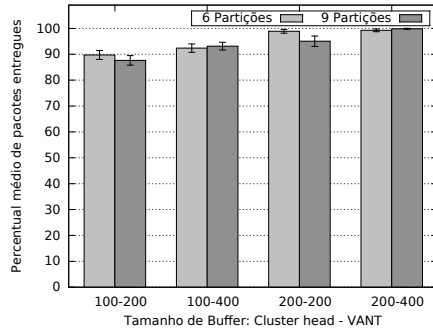
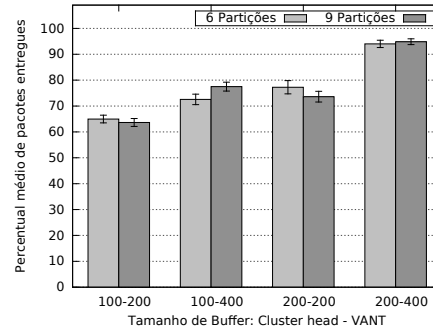
(a) 10 fluxos de dados, $4km^2$ de área.(b) 20 fluxos de dados, $4km^2$ de área.(c) 10 fluxos de dados, $9km^2$ de área.(d) 20 fluxos de dados, $9km^2$ de área.

Figura 13 Número médio de pacotes entregues com sucesso para o modelo ETSP.

identificar como o tamanho dos *buffers* tem influência na taxa de entrega da rede. No caso onde há os maiores *buffers* (200 para *cluster heads* e 400 para VANT), a perda é mínima. Para os dois cenários intermediários (100-400 e 200-200), verifica-se percentuais médios de entrega de 81% e 90% respectivamente. As perdas são maiores se comparadas com o caso onde os *buffers* são maiores, mas menores que no cenário que utiliza a combinação 100-200. Por outro lado, aumentar o *buffer* do VANT traz melhores resultados em redes mais particionadas.

Os resultados da Figura 13(c) se assemelham um pouco com os apre-

sentados na Figura 13(a). Uma taxa de perda levemente maior pode ser vista como resultado de um ciclo maior para o VANT. A Figura 13(d) apresenta o pior resultado: com *buffers* pequenos, para qualquer número de partições, as perdas são altas. A combinação de alta vazão de pacotes com uma área maior de simulação requer *buffers* grandes tanto para *cluster heads* como para VANT, como forma de evitar perdas.

5.2 Weighted

Como uma melhoria no modelo ETSP, o *Weighted* (ou *ETSP com Rodadas*) proposto na Seção 4.2 tem seus resultados nesta seção, considerando o fluxo de mensagens entre as diferentes partições como peso na ordem e frequência de visitação.

5.2.1 Configuração das simulações

Os cenários para avaliação deste modelo são apresentados na Tabela 3. Da mesma forma que para o modelo anterior, as trocas internas de mensagens não são consideradas fluxos.

Este modelo tem como característica tratar de redes com tráfego não balanceado entre partições, ou seja, não há uma distribuição uniforme dos fluxos. Assim, para estressar o sistema nesse ponto e avaliar a sua performance, são simulados dois tipos de fluxos de dados: *Concentrado* e *Esparso*. No primeiro, os fluxos são concentrados em uma área, ou seja, nem todas as partições são selecionadas para gerar/receber tráfego. Com essa política, o tráfego é restrito em 50% da área total de simulação, criando uma área com grande tráfego que cobre metade da área de simulação ($4,5km^2$). Isso mimetiza redes com atividade concentrada em uma pequena área, o que

Tabela 3 Parâmetros de configuração dos cenários simulados no modelo *Weighted*.

Parâmetros de Simulação	
<i>Parâmetro</i>	<i>Valor</i>
Área de Simulação (km^2)	3×3
Número de partições	6 e 9
Número de VANTs	1
Número de fluxos – NF	20
Pacotes por fluxo – PF	100
Vazão de cada fluxo ($\frac{pacotes}{s}$)	0,1
Número de pacotes enviados	NF \times PF
Tamanho do <i>buffer</i> para <i>cluster head</i> (<i>pacotes</i>)	100, 200
Tamanho do <i>buffer</i> para VANT (<i>pacotes</i>)	200, 400
Tipo de Fluxo	<i>Concentrado, Esparso</i>
Tipo de Movimento	<i>ETSP, ETSP com Rodadas</i>

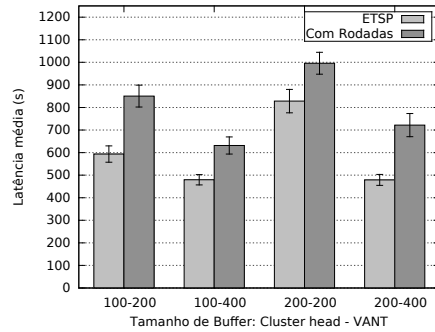
é um padrão realístico em RSSFs. Já para a simulação com fluxos esparsos, o tráfego é uniformemente distribuído na área de simulação, balanceando-o por toda a rede.

Também, para avaliar o impacto das melhorias deste modelo quando comparado ao ETSP (que não considera o tráfego desbalanceado da rede), todas as simulações foram executadas para os dois modelos de movimento.

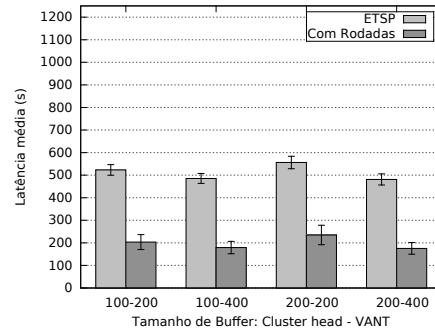
5.2.2 Resultados

A Figura 14 apresenta a latência média de entrega de pacotes para vinte fluxos. Na Figura 14(a) é possível notar uma latência similar para quase todas as configurações de *buffer*. A configuração 200 – 200 apresenta para ambos os tipos de movimentos um aumento na latência, devido ao fato de uma menor relação entre o tamanho de *buffers* para *cluster head*/VANT: apesar de um *cluster head* poder armazenar mais, o VANT não é capaz de transportá-los todos de uma vez. Assim, o VANT coleta tanto quanto

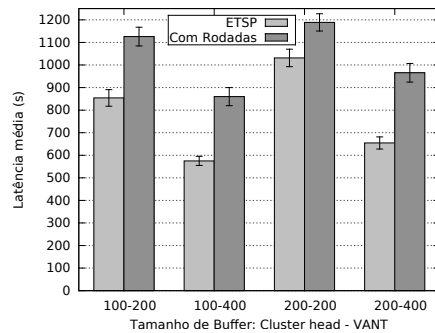
possível e o restante deve aguardar no *cluster head*, aumentando a latência. Esse padrão de resultado pode ser notado persistentemente em diferentes configurações.



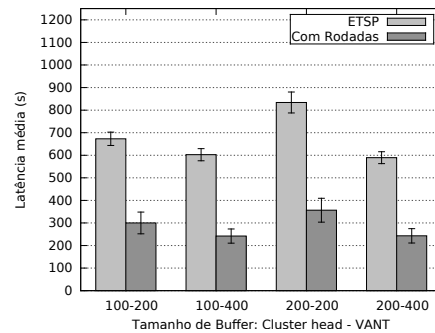
(a) 6 partições, fluxos esparsos.



(b) 6 partições, fluxos concentrados.



(c) 9 partições, fluxos esparsos.



(d) 9 partições, fluxos concentrados.

Figura 14 Latência média para entrega dos pacotes no modelo *Weighted*.

Para fluxos esparsos (Figuras 14(a) e (c)) é possível notar uma latência de entrega maior para o *ETSP com Rodadas* quando comparada ao ETSP puro. Isso acontece pelo seguinte fenômeno: o ETSP trata todas as partições de forma igualitária, visitando cada uma delas uma única vez por ciclo. O modelo proposto determina tipos de fluxos diferentes com base na forma do tráfego. Mesmo em um cenário esparsos, existem pequenas diferen-

ças no tráfego. Portanto, em algumas rodadas, apenas algumas partições são visitadas. Algumas vezes o VANT atravessa partições não agendadas para visitação em uma rodada (já que estão no caminho) mas não coleta/entrega pacotes para elas, aumentando assim a latência média total. Isso significa que o custo de priorizar partições mais ativas pode não compensar o custo de viajar por algumas áreas da rede sem visitar os *cluster heads* próximos devido ao tráfego reduzido. Esses *cluster heads* são visitados mais tarde e seus pacotes possuem um maior tempo de espera, o que colabora com esse aumento na latência.

Por outro lado, os cenários com tráfego concentrado apresentam uma tendência diferente: as Figuras 14(b) e (d) demonstram uma redução expressiva na latência para o *ETSP com Rodadas* proposto neste modelo. Enquanto para o ETSP puro a latência é similar àquela obtida para fluxos esparsos, o *ETSP com Rodadas* pode transportar pacotes com uma latência média reduzida para todas as configurações de *buffer*.

Também, como meio de avaliar a escalabilidade do modelo proposto nesta seção no que diz respeito à quantidade de fluxos, foram executadas simulações com 10 fluxos de dados. A Figura 15 mostra a relação entre as latências obtidas para dez e vinte fluxos. O resultado contém o percentual de aumento quando os cenários com vinte fluxos são comparadas com aqueles de dez fluxos. Analisando o resultado, nota-se que para todos os cenários, ao se dobrar a quantidade de fluxos (e conseqüentemente pacotes), a latência é aumentada no máximo 51% (para seis partições, fluxos esparsos e utilizando ETSP sem priorização).

O pior resultado geral é aquele com a menor razão entre tamanho de *buffer* para VANT/*cluster head* (200/200). Isso pode ser explicado pelo fato

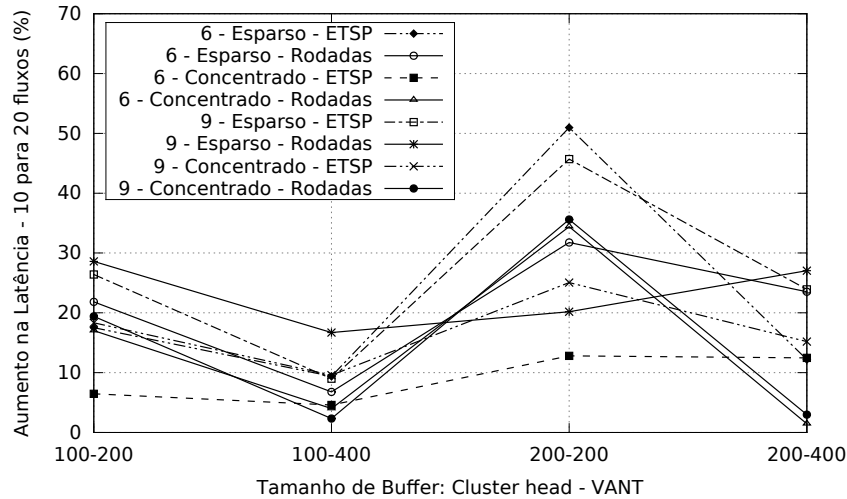
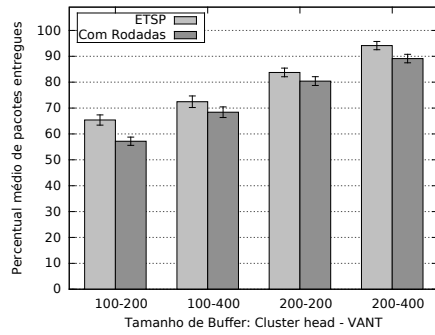


Figura 15 Escalabilidade da latência para o modelo *Weighted*.

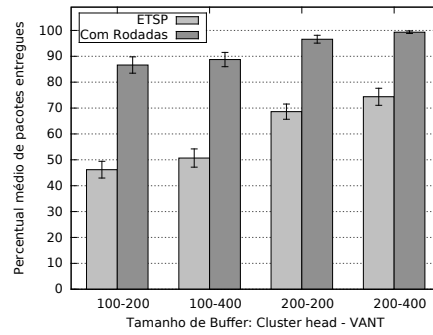
de que há pacotes armazenados para transporte futuro no *buffer* do *cluster head* durante mais de um ciclo do VANT. Como o *buffer* do VANT é um fator restritivo na análise da latência, embora um *buffer* maior para *cluster head* possa diminuir a taxa de perda de pacotes, ele também pode aumentar a latência total, já que o VANT não é sempre capaz de transportar todos os pacotes de uma vez.

No cenário supracitado, os experimentos realizados com tráfego esparso e utilizando o ETSP sem priorização para a combinação de *buffer* 200 – 200 apresentaram o maior ganho na latência, ou seja, menor escalabilidade. Isso acontece devido à presença de uma menor razão entre os tamanhos de *buffer* juntamente com o tráfego esparso, levando a um considerável aumento na latência devido ao maior tráfego. Por outro lado, esse fenômeno não ocorre com a mesma intensidade quando há priorização de tráfego para a determinação de rotas. Considerando que o *ETSP com Rodadas* trata do tráfego desbalanceado, ele analisa essa vazão maior na

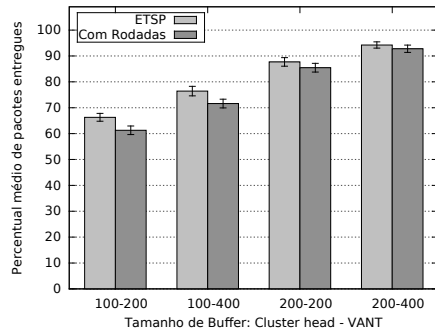
rede para a geração das rotas, lidando mais apropriadamente com a maior demanda.



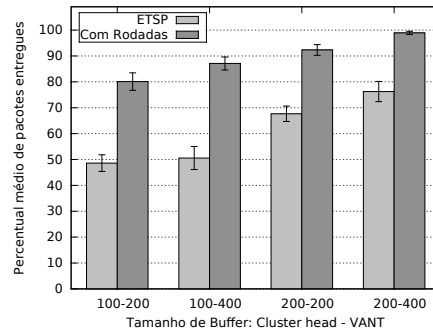
(a) 6 partições, fluxos esparsos.



(b) 6 partições, fluxos concentrados.



(c) 9 partições, fluxos esparsos.



(d) 9 partições, fluxos concentrados.

Figura 16 Número médio de pacotes entregues com sucesso para o modelo *Weighted*.

A Figura 16 mostra o comportamento do modelo quando o número de pacotes entregues é considerado. No gráficos (a) e (c) da figura, os cenários com fluxo esparsos para seis e nove partições são exibidos. Nesses cenários, a abordagem com priorização de tráfego entrega um número similar de pacotes quando comparada ao ETSP puro, para todas as combinações de *buffer*. É possível observar a influência do tamanho do *buffer* do *cluster head* na perda de pacotes. Quanto menor esse *buffer*, maior a perda.

Isso é explicado pela impossibilidade de armazenar todos os pacotes gerados entre duas visitas do VANT. Essa situação é amenizada quando o modelo *ETSP com Rodadas* é utilizado para priorizar áreas com tráfego concentrado (Figuras 16(b) e (d)).

Um efeito positivo do *ETSP com Rodadas* combinado com tráfego concentrado é uma redução drástica nas taxas de perda para situações onde o *buffer* do *cluster head* é pequeno. As perdas ocorrem pela saturação do *cluster head* e, utilizando-se priorização, esses *cluster heads* passam a ser visitados em intervalos de tempo menores.

Uma imagem geral dos resultados obtidos demonstra que para fluxos esparsos o *ETSP com Rodadas* opera tão bem quanto o ETSP sem priorização, no que se refere a uma baixa taxa de perdas. No entanto, nessa mesma situação o *ETSP com Rodadas* apresenta uma maior latência média. No caso de cenários com padrão de tráfego concentrado, o modelo proposto nesta seção apresenta redução na latência para todas as combinações de *buffer* ao mesmo tempo que aumenta o percentual de pacotes entregues (especialmente em redes saturadas).

5.3 Genetic

Esta seção apresenta as configurações e resultados de simulações para o último modelo proposto neste trabalho e definido na Seção 4.3. Trata-se de um modelo mais geral onde há a possibilidade de emprego de múltiplos VANTs.

5.3.1 Configuração das simulações

As configurações para simulação deste modelo estão apresentadas na Tabela 4. São utilizados cenários com quantidades diferentes de VANTs, assim como mais fluxos e partições. Como descrito na Seção 4.3.2, para se resolver este problema mais abrangente em um tempo computacionalmente viável foi utilizado algoritmo genético. Assim, para se obter soluções, o cálculo em cada cenário contou com 40 indivíduos evoluídos durante 50.000 gerações utilizando torneio de tamanho 3 e 0,7 como probabilidade tanto para o operador de *crossover* como para o de mutação.

Tabela 4 Parâmetros de configuração dos cenários simulados no modelo *Genetic*.

Parâmetros de Simulação	
<i>Parâmetro</i>	<i>Valor</i>
Área de Simulação (km^2)	4×4
Número de Partições	24
Número de VANTs	4 e 8
Número de fluxos – NF	20 e 40
Pacotes por fluxo – PF	100
Vazão de cada fluxo ($\frac{packets}{s}$)	0.1
Número de pacotes enviados	NF \times PF
Tamanho de <i>buffer</i> para <i>cluster head</i> (pacotes) (packets)	100, 200
Tamanho de <i>buffer</i> para VANT (pacotes)	200, 400
Tipo de movimento	<i>Single / Multiple ETSP</i>

Duas variações do modelo de movimento são utilizadas para geração e avaliação de resultados. Na primeira, *Single ETSP*, calcula-se uma rota global, ou seja, a rede é formada por uma única região. Essa mesma rota é atribuída a todos os VANTs, mas com início de ciclo em pontos diferentes (a fim de evitar vários VANTs visitando o mesmo *cluster head* no mesmo tempo). Para esse movimento, não há necessidade de ponto de encontro

entre os VANTs, pois existe apenas uma região. Já no *Multiple ETSP*, a rede é dividida em n regiões para n VANTs, de forma que cada VANT é responsável por uma única região. O método de formação das regiões e escolha do ponto de encontro é o descrito na Seção 4.3.2.

5.3.2 Resultados

Nesta seção estão apresentados os resultados de simulações para as duas abordagens utilizadas para organizar um conjunto de VANTs de modo que cubram uma área com diversas partições desconexas. A Figura 17 apresenta os gráficos para quando a latência é analisada em cenários com vinte e quarenta fluxos de dados.

Os gráficos (a) e (b) da figura mostram a latência para a configuração de vinte fluxos utilizando as duas estratégias. Para ambas, o incremento no número de VANTs resultou em uma redução no tempo médio de entrega. Isso pode ser explicado tendo em vista que utilizando o *Single ETSP*, mais veículos levam a uma redução do intervalo de visita em cada *cluster head*, o que acarreta um menor tempo médio de espera de um pacote no *buffer*. O incremento no tamanho do *buffer* dos VANTs leva a uma ligeira redução na latência, pois há uma maior chance de nenhum pacote ser deixado no *cluster head* para ser transportado pelo próximo VANT. Já para o caso do *Multiple ETSP*, mais veículos implicam em menores regiões. Menores regiões levam a menores ciclos, portanto os pacotes de uma mesma região permanecem menos tempo no VANT. Também, a troca de pacotes entre regiões distintas no ponto de encontro dos VANTs ocorre em um intervalo menor (cujo limite superior é dado pelo maior ciclo). Por outro lado, o movimento utilizando um único ETSP é mais sensível ao aumento do número de VANTs, portanto

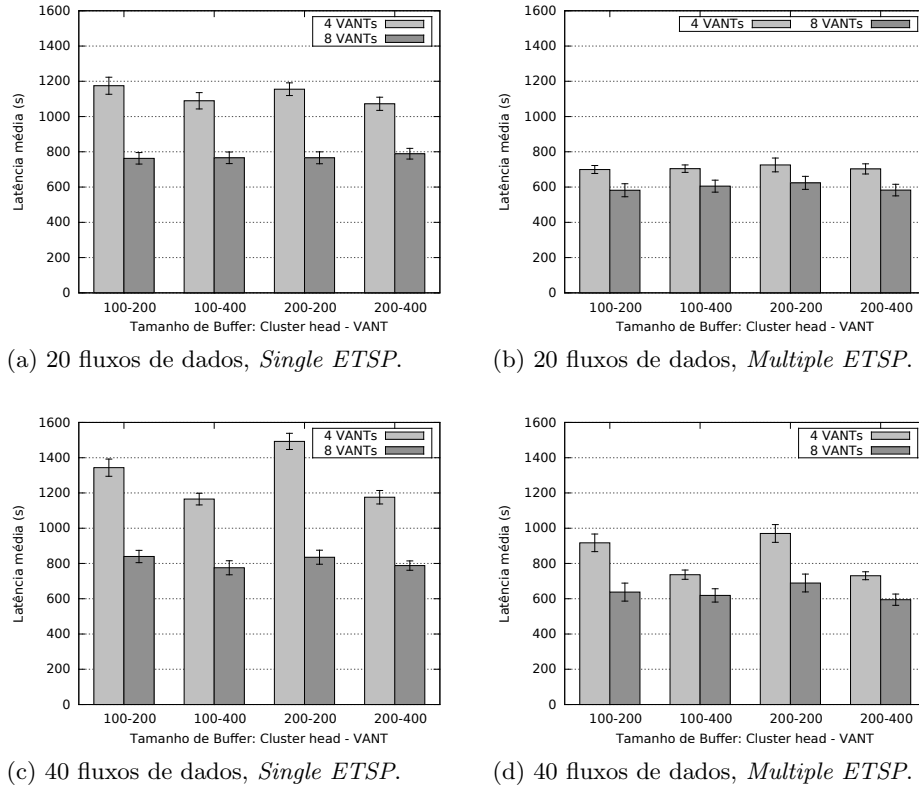


Figura 17 Latência média para entrega dos pacotes no modelo *Genetic*.

aumentá-lo traz melhorias, como pode ser visto na Figura 17(a).

Outra importante conclusão que pode ser tirada quando se comparando as duas abordagens é a redução na latência no *Multiple ETSP* em relação ao *Single ETSP*. A explicação para esse fenômeno pode ser dada pelo menor ciclo que um VANT deve percorrer para servir sua região. Pacotes endereçados para partições na mesma região são prontamente servidos. Além disso, até mesmo aqueles pacotes para outras regiões são transportados mais rápidos pois eles são transferidos no ponto de encontro diretamente para o VANT responsável pela partição de destino. Isso evita uma viagem

do pacote por toda a área da rede, como ocorre no pior caso do *Single ETSP*. Um pacote quando utilizado-se o *Multiple ETSP* permanece menos no *buffer* do VANT. Isso é especialmente verdade para cenários com menos VANTs, tendo sido obtida uma redução na latência de até 41% para a combinação de 100 – 200 para *buffers*.

As Figuras 17(c) e (d) contêm os resultados para latência para cenários com quarenta fluxos de dados. Como esperava-se, a latência média aumenta para ambos os tipos de movimento. No entanto, esse resultado é mais substancial para situações com menos VANTs, pois um grande número de pacotes deve ser carregado por menos aviões. Assim, o tamanho do *buffer* do VANT tem um importante papel na latência: maiores *buffers* resultam em menores latências, pois nenhum pacote é deixado para ser transportado em uma próxima visita devido à saturação do *buffer* do VANT. O cenário com menor *buffer* para VANT e maior para *cluster head* com quatro VANTs tem o pior resultado: muitos pacotes são mantidos no *buffer* do *cluster head* aguardando uma próxima visita quando o *buffer* do avião está cheio. Comparando as duas estratégias, as mesmas conclusões mencionadas no parágrafo anterior são obtidas.

A Figura 18 apresenta o percentual de pacotes entregues com sucesso ao destino final para as duas estratégias apresentadas, utilizando vinte e quarenta fluxos de dados. Para vinte fluxos, são enviados no total 2.000 pacotes na rede e, para quarenta, são enviados 4.000 pacotes. Os gráficos (a) e (b) da figura trazem os percentuais de entrega para vinte fluxos. Para ambos os tipos de movimento e todas as configurações de *buffer*, apenas uma pequena taxa de perda foi verificada, mostrando que ambas as abordagens são adequadas para esses cenários. No caso do *Single ETSP*, para *buffers* menores

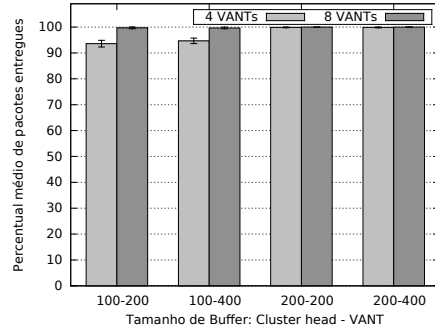
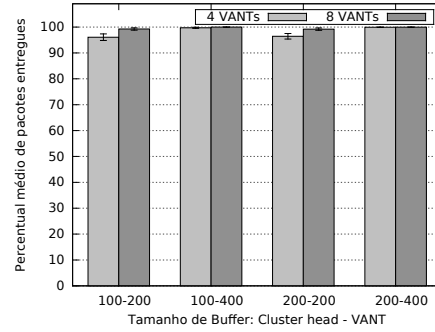
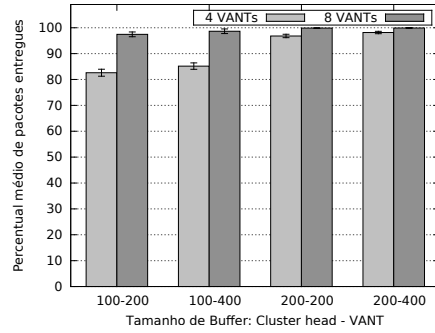
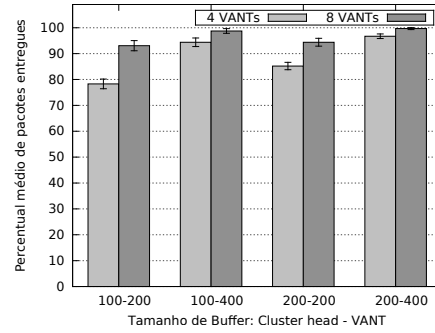
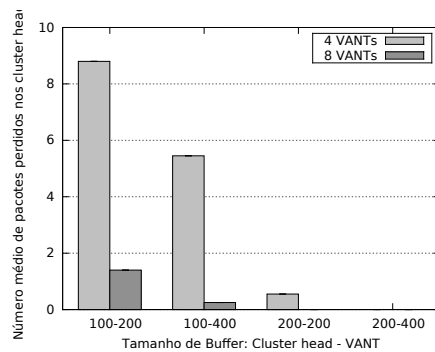
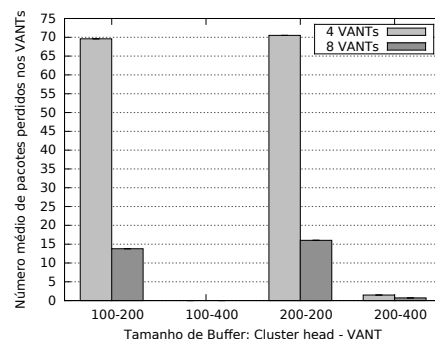
(a) 20 fluxos de dados, *Single ETSP*.(b) 20 fluxos de dados, *Multiple ETSP*.(c) 40 fluxos de dados, *Single ETSP*.(d) 40 fluxos de dados, *Multiple ETSP*.

Figura 18 Número médio de pacotes entregues com sucesso para o modelo *Genetic*.

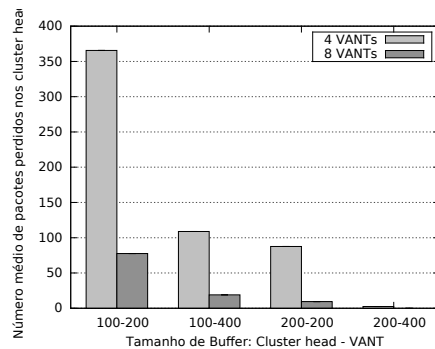
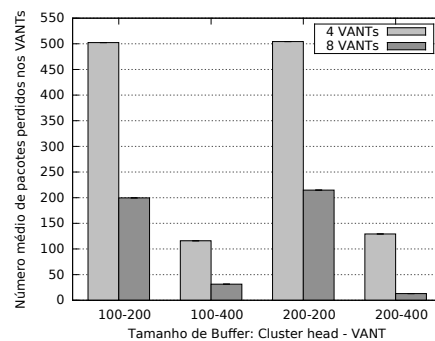
nos *cluster heads*, uma pequena taxa de perda foi observada. Isso é justificado pelo espaço insuficiente em *buffer* para armazenar todos os pacotes que chegam a um *cluster head* antes que um VANT o visite, considerando que o *Single ETSP* atribui uma mesma rota (grande) para todos os VANTs. Esse fenômeno, no entanto, não é observado no movimento *Multiple ETSP*.

Os resultados para quarenta fluxos estão nos gráficos (c) e (d) da Figura 18. Para todas as configurações e ambas as estratégias de movimento, uma taxa máxima de 20% de perda foi observada (*Multiple ETSP*, quatro VANTs, *buffers* de 100 – 200). Para outras configurações, obteve-se resul-

tados melhores. Por exemplo, para quatro VANTs e *buffers* em 100–400, o *Multiple ETSP* apresentou um percentual médio de entrega de 94% dos pacotes, enquanto que o *Single ETSP* pôde entregar 84% dos pacotes ao destino. Para oito veículos e *buffers* em 100–200, o *Single ETSP* foi capaz de entregar em média 96% dos pacotes, enquanto o *Multiple ETSP* entregou 92%.

(a) 20 fluxos de dados, *cluster heads*.

(b) 20 fluxos de dados, VANTs.

(c) 40 fluxos de dados, *cluster heads*.

(d) 40 fluxos de dados, VANTs.

Figura 19 Número médio de pacotes perdidos nos *cluster heads* e VANTs.

Os pacotes perdidos nos resultados vêm de saturação do *buffer* de *cluster heads* (para ambos os tipos de movimento) ou, quando considerando especificamente o *Multiple ETSP*, perdas também podem ocorrer devido à

saturação dos VANTs quando da troca de pacotes entre eles. Para o último caso, os números de perdas são apresentados na Figura 19. No gráfico (a) da figura, a perda gerada nos *cluster heads* para vinte fluxos é exibida. Como esperado, uma maior perda foi observada com quatro VANTs e *buffer* pequeno para *cluster heads*. Também como esperado, quando analisa-se as perdas geradas na troca de pacotes entre VANTs, as maiores taxas ocorrem cenários com quatro VANTs com *buffers* menores (200). Isso é justificado pelo fato de que com menos aviões, cada um deles é responsável por mais partições e conseqüentemente no momento da troca tendem a receber mais pacotes. Assim, com pouco espaço para armazenar novos pacotes, ocorrem mais perdas.

Já as Figuras 19(c) e (d) contém os resultados para quarenta fluxos de dados. As observações realizadas para vinte fluxos também são válidas para estes resultados. No entanto, com quarenta fluxos os números são ampliados pois, por exemplo, até quinhentos pacotes são perdidos pelo VANTs quando estes possuem *buffers* pequenos. Outro resultado significativo é a taxa de perdas nos *cluster heads* com *buffers* pequenos (100) e quatro aviões também com *buffers* pequenos. Mais de 350 pacotes são perdidos nessa configuração. Aumentando ou o *buffer* dos *cluster heads* ou dos VANTs consegue-se uma notável melhora. Quando o *buffer* dos *cluster heads* é aumentado, um intervalo maior para coleta dos pacotes é tolerado, pois o *cluster head* passa a poder armazenar mais pacotes sem gerar perdas. Esse intervalo maior é útil quando os VANTs possuem *buffers* pequenos e não conseguem transportar todos os pacotes em uma única visita. Já quando apenas o *buffer* dos VANTs é aumentado, a redução da taxa de perda nos *cluster heads* é considerável, pois os pacotes não precisam aguardar vários

ciclos para serem coletados (o VANT tem espaço suficiente em cada ciclo para coletar todos os pacotes que aguardam transporte).

Tabela 5 Número médio de pacotes trocados entre os VANTs.

Pacotes trocados entre VANTs		
Fluxos	Número de VANTs	
	4	8
20	42%	46.4%
40	50.3%	50.2%

Considerando que um certo número de pacotes é endereçado para partições da mesma região da origem e outros para diferentes regiões, também é apresentado o percentual de pacotes que é trocado entre os VANTs, ou seja, destinados a partições de outras regiões. Esses números estão na Tabela 5. Para todos os cenários, entre quarenta e cinquenta por cento dos pacotes são trocados no ponto de encontro dos VANTs. Considerando que o número de partições externas à região é muito maior que as internas, esse resultado demonstra o agrupamento de partições realizado pelo algoritmo genético, em que a função *fitness* recompensa a formação de regiões com maior comunicação interna (Equação 14).

5.4 Considerações Finais

Com base nos resultados apresentados, para cada um dos modelos, pode-se demonstrar como o padrão de tráfego é um fator decisivo na escolha do modelo de movimento mais adequado para uma dada rede.

Enquanto a priorização proporcionada pelo modelo *Weighted* leva, no melhor caso para tráfego concentrado, a uma taxa de entrega de até 99% dos pacotes, no ETSP puro essa taxa não supera 76%. Por outro lado, nos cenários com fluxos esparsos, o *Weighted* tem uma entrega similar ao

ETSP puro, observando-se que na média uma distribuição uniforme dos fluxos faz com que não haja áreas com tráfego muito superior às demais. Já para a latência, quando o tráfego é concentrado, priorizar áreas com maior comunicação trouxe redução de até 63% no tempo médio de entrega dos pacotes.

Quando o número de partições na rede é alto (no caso avaliado, 24 partições), a utilização de mais VANTs, em que cada um é responsável por algumas partições, apresentou melhores resultados comparando-se aos cenários com menos partições e um único VANT. A taxa de entrega para 8 VANTs em *Multiple ETSP*, 24 partições e 20 fluxos, no pior caso (100–200), foi de 96%, enquanto para o ETSP puro com 9 partições e 20 fluxos foi de 66% e no *Weighted* (sob as mesmas condições) foi de 61%. Já para a latência com 8 VANTs em *Multiple ETSP* e 20 fluxos de dados, o ganho médio foi de 39% para a combinação *buffers* de 200–200 quando comparado ao ETSP puro e de 48% em relação ao *Weighted* e *buffers* em 100–200. Nesses comparativos apresentados para o modelo *Genetic / Multiple ETSP* foram utilizados apenas cenários com fluxos esparsos, dadas as condições de simulação do modelo *Genetic*, em que não há necessariamente concentração do tráfego.

6 Conclusão

Este trabalho propôs um conjunto de métodos para restabelecimento da comunicação entre partições desconexas de uma RSSF. Para tanto é utilizado o conceito de *message ferrying* e VANTs para transportar fisicamente os pacotes entre partições distintas.

Para organizar a rede e prover um enlace entre as partições e o VANT, um *cluster head* é eleito em cada partição. Assim, todos os pacotes direcionados a nós de outras partições são enviados para o *cluster head*, onde aguardam para serem transportados pelo VANT.

Três modelos de movimento foram propostos para definição de rotas de VANTs. O primeiro trata-se de um modelo básico que utiliza um ciclo hamiltoniano mínimo entre os *cluster heads* para determinar a rota de um VANT. Já o segundo modelo, com base no primeiro, também considera o tráfego presente nas partições para determinar a melhor rota para um VANT. Assim, partições com maior tráfego devem ser visitadas com maior frequência. Por fim, um modelo mais geral para múltiplos VANTs e priorização de tráfego foi apresentado. Esse modelo tenta minimizar tanto o tamanho do ciclo realizado por cada VANT, mas também manter o máximo possível de tráfego interno dentro de cada região.

Foram realizadas simulações para cada um dos modelos de movimento em diferentes cenários. Esses cenários diferem em número de partições, área, tamanho para *buffers*, número e padrão de distribuição de fluxos de dados, entre outras características.

Comparando-se os modelos, aquele que considera o tráfego para determinar as rotas foi capaz de entregar, no melhor cenário, 99% da quantidade de pacotes em cenários com tráfego esparsos (uniformemente distri-

buído) quando comparado à versão sem priorização. No entanto quando o tráfego é concentrado em uma área parcial da rede, a priorização proveu um aumento de até 87% na taxa de entrega de pacotes e, ao mesmo, reduziu a latência média, para um dado cenário, de 523s para 203s.

Os resultados, para o modelo com múltiplos VANTs, foram obtidos com base em duas variações do modelo. Uma delas, *Single ETSP*, atribui a mesma rota a todos os VANTs do sistema, enquanto a outra (*Multiple ETSP*) gera regiões distintas, sendo uma atribuída para cada VANT. Pelos resultados demonstrara-se que, apesar de ambas as variações serem capazes de entregar no mínimo 80% parte dos pacotes, a latência para o *Multiple ETSP* foi significativamente menor. Esse efeito foi acentuado para cenários com tráfego mais intenso e menos VANTs, em que houve redução de até 41% na latência.

Uma visão geral dos resultados evidencia o impacto de características de comunicação próprias da rede e, também, de capacidades de armazenamento tanto dos nós sensores como dos VANTs. De modo geral, *buffers* maiores para os *cluster heads* implicam menores taxas de perda e, para os VANTs, tendem a reduzir a latência na entrega. Além disso, um maior número de VANTs reduz a latência, seja por conta de menores ciclos ou menor intervalo de visitação a cada *cluster head*.

Muitas RSSFs devem ter seu tempo de vida estendido o máximo possível. Entretanto, isso é um objetivo afetado pelos particionamentos que a rede sofre. Assim, este trabalho apresentou e avaliou técnicas para minimizar o efeito de tais desconexões, aumentando o tempo de vida de aplicações tolerantes a atrasos.

6.1 Trabalhos futuros

Além dos modelos de movimento apresentados neste trabalho, outros podem ser propostos.

Uma possibilidade é o desenvolvimento de um modelo em que alguns VANTs sejam responsáveis por estabelecer a comunicação entre os demais, eliminando a necessidade de um ponto de encontro entre eles. Também, pode-se utilizar, ao invés de um ponto de encontro, uma interseção entre regiões, em que uma dada região é compartilhada entre dois VANTs, sendo ela o ponto de troca.

Todo o trabalho desenvolvido utilizou nós sensores sem mobilidade e apenas os VANTs são capazes de restabelecer a comunicação entre partições desconexas. Em um cenário em que os nós (ou outros dispositivos quaisquer) forem móveis, é possível explorar padrões de mobilidade para aumentar a conectividade da rede. Assim, protocolos de roteamento oportunista podem ser utilizados para explorar a movimentação intrínseca dos nós da rede, juntamente com VANTs para prover um enlace entre partes da rede muito ou totalmente desconexas.

Por fim, também é possível que não apenas a quantidade e padrão de distribuição do tráfego da rede sejam utilizados como peso para a determinação de rotas. Necessidades próprias de qualidade de serviço de aplicações distribuídas que executam em uma rede podem ser pesos, a fim de que tais aplicações se mantenham operacionais mesmo na presença de particionamentos na rede.

REFERÊNCIAS

AL-TURJMAN, F.; HASSANEIN, H.; IBNKAHLA, M. Optimized wireless sensor network federation in environmental applications. In: GLOBAL TELECOMMUNICATIONS CONFERENCE, 54., 2011, Houston. **Proceedings...** Houston: IEEE, 2011. p. 1-5.

ATAY, N.; BAYAZIT, B. Mobile wireless sensor network connectivity repair with k-redundancy. In: CHIRIKJIAN, G. et al. (Ed.). **Algorithmic foundation of robotics VIII**. Berlin: Springer, 2009. p. 35-49. (Springer Tracts in Advanced Robotics, 57).

AZZAZ, S.; SAIDANE, L. A.; MINET, P. Repairing sensors strategies in fault-tolerant wireless sensor networks. In: ACM SYMPOSIUM ON PERFORMANCE EVALUATION OF WIRELESS AD HOC, SENSOR, AND UBIQUITOUS NETWORKS, 8., 2011, New York. **Proceedings...** New York: ACM, 2011. p. 113-116.

BEKTAS, T. The multiple traveling salesman problem: an overview of formulations and solution procedures. **Omega**, Elmsford v. 34, n. 3, p. 209-219, 2006.

BERBEGLIA, G.; CORDEAU, J. F.; LAPORTE, G. Dynamic pickup and delivery problems. **European Journal of Operational Research**, Amsterdam, v. 202, n. 1, p. 8-15, 2010.

BONE, E.; BOLKCOM, C. **Unmanned aerial vehicles: background and issues for congress**. Washington: The Library of Congress, 2003. 52 p.

ERMAN, A. et al. Enabling mobility in heterogeneous wireless sensor networks cooperating with uavs for mission-critical management. **Wireless Communications**, New York, v. 15, n. 6, p. 38-46, 2008.

GARCIA-HERNANDEZ, C. F. et al. Wireless sensor networks and applications: a survey. **International Journal of Computer Science and Network Security**, Seoul, v. 17, n. 3, p. 264-273, 2007.

HANSMANN, U. **Pervasive computing: the mobile world**. New York: Springer, 2003. 448 p. (Springer Professional Computing).

HEIMFARTH, T.; ARAUJO, J. P. de. Using unmanned aerial vehicle to connect disjoint segments of wireless sensor network. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS, 28., 2014, Victoria. **Proceedings...** Victoria: IEEE, 2014. p. 907-914.

HEIMFARTH, T.; ARAUJO, J. P. de; GIACOMIN, J. C. Unmanned aerial vehicle as data mule for connecting disjoint segments of wireless sensor network with unbalanced trac. In: INTERNATIONAL SYMPOSIUM ON OBJECT/COMPONENT/SERVICE-ORIENTED REAL-TIME DISTRIBUTED COMPUTING, 17., 2014, Reno. **Proceedings...** Reno: IEEE, 2014. p. 246-252.

HEIMFARTH, T. et al. Evaluation of a genetic programming approach to generate wireless sensor network applications. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS, 28., 2014, Victoria. **Proceedings...** Victoria: IEEE, 2014a. p. 775-782.

HEIMFARTH, T. et al. Gb-mac: a backbone based low latency protocol for wsns. In: ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING, 29., 2014, New York. **Proceedings...** New York: ACM, 2014b. p. 692-698.

HEIMFARTH, T. et al. Utilização de múltiplos veículos aéreos não tripulados para o estabelecimento de comunicação em redes de sensores sem o particionadas. In: WORKSHOP DE SISTEMAS DISTRIBUÍDOS AUTONÔMICOS, 4., 2014, Florianópolis. **Anais...** Florianópolis: WOSIDA, 2014c. p. 25-28.

HEIMFARTH, T.; GIACOMIN, J. C.; ARAUJO, J. P. de. Aga-mac: adaptive geographic anycast mac protocol for wireless sensor networks. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS, 29., 2015, Gwangju. **Proceedings...** Gwangju: IEEE, 2015. p. 373-381.

HOLLAND, J. H. **Adaptation in natural and artificial systems.** Ann Arbor: University of Michigan, 1975. 183 p.

HROMKOVIC, J. **Algorithmics for hard problems:** introduction to combinatorial optimization, randomization, approximation, and heuristics. 2nd ed. Berlin: Springer-Verlag, 2004. 544 p.

HUNJET, R.; HUI, P. Maintaining connectivity in mobile adhoc networks using distributed optimisation. In: COMMUNICATIONS AND INFORMATION SYSTEMS CONFERENCE, 2011, Caberra. **Proceedings...** Caberra: MilCIS, 2011. p. 1-6.

JOSHI, Y.; YOUNIS, M. Mobility-based internetworking of disjoint segments. In: BIENNIAL SYMPOSIUM ON COMMUNICATIONS, 27., 2014, Kingston. **Proceedings...** Kingston: QBSC, 2014. p. 193-197.

KARL, H.; WILLIG, A. **Protocols and architectures for wireless sensor networks.** New York: J. Wiley, 2005. 526 p.

KARP, B.; KUNG, H. T. Gpsr: greedy perimeter stateless routing for wireless networks. In: ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, 6., 2000, New York. **Proceedings...** New York: ACM, 2000. p. 243-254.

KIMENCE, S.; BEKMEZCI, I. Weighted relay node placement for wireless sensor network connectivity. **Wireless Networks**, New York, v. 20, n. 4, p. 553-562, 2014.

KONSTANTINIDIS, A.; YANG, K. Multi-objective k-connected deployment and power assignment in wsns using a problem-specific constrained evolutionary algorithm based on decomposition. **Computer Communications**, Amsterdam, v. 34, n. 1, p. 83-98, Jan. 2011.

KUROSE, J.; ROSS, K. **Computer networking:** a top-down approach. Boston: Pearson Education, 2012. 864 p. (Always Learning).

LEE, S.; YOUNIS, M. Optimized relay placement to federate segments in wireless sensor networks. **IEEE Journal on Selected Areas in Communications**, Piscataway, v. 28, n. 5, p. 742-752, June 2010.

LESSMANN, J.; HEIMFARTH, T.; JANACIK, P. Shox: an easy to use simulation platform for wireless networks. In: INTERNATIONAL CONFERENCE ON COMPUTER MODELING AND SIMULATION, 20., 2008, Washington. **Proceedings...** Washington: IEEE Computer Society, 2008. p. 410-415.

LI, N.; HOU, J. Improving connectivity of wireless ad hoc networks. In: ANNUAL INTERNATIONAL CONFERENCE ON MOBILE AND UBIQUITOUS SYSTEMS: NETWORKING AND SERVICES, 2., 2005, San Diego. **Proceedings...** San Diego: MobiQuitous, 2005. p. 314-324.

LLORCA, J.; MILNER, S.; DAVIS, C. Mobility control for joint coverage-connectivity optimization in directional wireless backbone networks. In: MILITARY COMMUNICATIONS CONFERENCE, 2007, Orlando. **Proceedings...** Orlando: IEEE, 2007. p. 1-7.

LOUREIRO, A. A. F. et al. Redes de sensores sem fio. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 21., 2003, Natal. **Anais...** Natal: Sociedade Brasileira de Computação, 2003. p. 1-48.

MARINHO, M. et al. Using cooperative mimo techniques and uav relay networks to support connectivity in sparse wireless sensor networks. In: INTERNATIONAL CONFERENCE ON COMPUTING, MANAGEMENT AND TELECOMMUNICATIONS, 2013, Ho chi Minh. **Proceedings...** Ho chi Minh: ComManTel, 2013. p. 49-54.

OLIVEIRA, H. E. de et al. An adaptive routing protocol based on xed hubs for opportunistic networks. In: INTERNATIONAL CONFERENCE ON INNOVATIVE MOBILE AND INTERNET SERVICES IN UBIQUITOUS COMPUTING, 9., 2015, Blumenau. **Proceedings...** Blumenau: IMIS, 2015. 1 CD-ROM.

UNITED STATES OF AMERICA. Secretary of Defense. **Unmanned Aircraft Systems (UAS) Roadmap**. Washington, 2005. 213 p.

VALAVANIS, K. P. **Advances in unmanned aerial vehicles: state of the art and the road to autonomy: intelligent systems, control and automation: science and engineering.** New York: Springer, 2007. 544 p.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. **Computer Networks**, Amsterdam, v. 52, n. 12, p. 2292-2330, 2008.

ZHAO, W.; AMMAR, M.; ZEGURA, E. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: ACM INTERNATIONAL SYMPOSIUM ON MOBILE AD HOC NETWORKING AND COMPUTING, 5., 2004, New York. **Proceedings...** New York: ACM, 2004. p. 187-198.

ZHAO, Z.; GAO, C.; DUAN, F. A survey on autonomic computing research. In: ASIA-PACIFIC CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND INDUSTRIAL APPLICATIONS, 2009, Wuhan. **Proceedings...** Wuhan: PACIIA, 2009. v. 2, p. 288-291.