



**ADAUTO MARDEN MENDES**

**RESTABELECENDO A CONEXÃO EM REDES DTN  
UTILIZANDO UAVS PARA O TRANSPORTE DE  
PACOTES**

**LAVRAS - MG**

**2012**

**ADAUTO MARDEN MENDES**

**RESTABELECENDO A CONEXÃO EM REDES DTN UTILIZANDO  
UAVS PARA O TRANSPORTE DE PACOTES**

Monografia apresentada ao Colegiado do  
Curso de Sistemas de Informação do De-  
partamento de Ciência da Computação,  
para obtenção do título de Bacharel em  
Sistemas de Informação.

Orientador

Prof. Tales Heimfarth

**LAVRAS - MG**

**2012**

**ADAUTO MARDEN MENDES**

**RESTABELECENDO A CONEXÃO EM REDES DTN  
UTILIZANDO UAVS PARA O TRANSPORTE DE  
PACOTES**

Monografia de graduação apresentada ao  
Colegiado do Curso de Sistemas de  
Informação, para obtenção do título de  
Bacharel em Sistemas de Informação.

APROVADA em 18 de outubro de 2012.

RAPHAEL WINCKLER DE BETTIO

HERMES PIMENTA DE MORAES JÚNIOR



TALES HEIMFARTH (orientador/a)

**LAVRAS-MG**

**2012**

*Dedico esta monografia a meus pais Edilson Mendes e Delmira Cardoso Mendes,  
as minhas irmãs Brunna e Maria Cecília, a minha namorada Regina, a meus  
familiares e todos meus amigos.*

## **AGRADECIMENTOS**

A Deus, por abençoar cada passo meu. Aos meus pais Edilson Mendes Martins e Delmira Cardoso Mendes, minhas irmãs Brunna e Cecília e minha namorada e companheira Regina.

Aos amigos que fiz durante o curso, pela verdadeira amizade que construímos por todos os momentos que passamos durante esses cinco anos meu especial agradecimento.

Ao professor e orientador Tales Heimfarth por sua dedicação nas horas que mais precisei, pela sua confiança depositada em mim para conclusão deste projeto.

A todos os professores do curso de Sistemas de Informação, pela paciência, dedicação e ensinamentos disponibilizados nas aulas, cada um de forma especial contribuiu para a conclusão desse trabalho e conseqüentemente para minha formação profissional.

Por fim, gostaria de agradecer a todos, em especial a meus avôs Alfredo Mendes Ferreira e Sebastião Cardoso (*in memoriam*), pelo carinho e por todos os momentos que puder passar junto a vocês, minha tia Jú por todo seu tempo gasto comigo e a todos que contribuíram direta ou indiretamente para que esse trabalho fosse realizado meu eterno AGRADECIMENTO.

## **RESUMO**

Este trabalho tem como objetivo restabelecer a comunicação em uma Rede de Sensores Sem fio quando a mesma estiver com problemas de comunicação. Será utilizado veículos aéreos não tripulados(UAV) para carregar pacotes de uma região comprometida ao destino. A princípio, o UAV estará em trajetória aleatória até que identifique pontos isolados de falha na rede, e cada ponto encontrado é incluído na trajetória do UAV. Sendo assim, o UAV se desloca até essas regiões desconexas periodicamente, coletando os pacotes e enviando até o nó destino.

Palavras-chave: Redes de Sensores Sem Fio; Veículos Aéreos Não Tripulados; RSSF, UAV.

## **ABSTRACT**

This study aims to restore communication in a Wireless Sensor Network when it is with communication problems. Will use unmanned aerial vehicles (UAV) to load packages from a region committed to the destination. At first, the UAV will be in random path until you identify single points of failure in the network, and each point found is included in the trajectory of the UAV. Thus, the UAV moves to these regions disconnected periodically collecting and sending the packets to the destination node.

Keywords: Wireless Sensor Networks; Unmanned Aerial Vehicles, RSSF, UAV.



## LISTA DE FIGURAS

Figura 1	Exemplo de salto em RSSFs. A figura "a" representa a desconectividade em um salto, enquanto a figura "b" aborda a desconexão em mais de um salto. ....	14
Figura 2	store-carry-and-forward.(www.sciencedirect.com/science/) .....	14
Figura 3	Redes Ad hoc. [25] .....	18
Figura 4	Exemplo da camada de agregação da arquitetura DTN [45]. ....	23
Figura 5	Rede de sensores sem fio. [29] .....	26
Figura 6	Exemplo de nós sensores dispostos em uma rede. [30] .....	30
Figura 7	Utilizando Roteamento Geográfico para enviar do nó x ao nó D. [49] .....	32
Figura 8	Exemplo de Dead End em uma RSSF. [52]. ....	32
Figura 9	Imagem do Predador. [27]. ....	33
Figura 10	Imagem do Firebee. [27] .....	34
Figura 11	Imagem do Global Hawk. [28]. ....	35
Figura 12	Foto do AR Drone. [26] .....	36
Figura 13	Visualizador do GRUBiX. ....	39
Figura 14	Exemplos das ilhas de desconexão em uma RSSF. ....	40
Figura 15	Exemplo de nós que tem pacotes a serem enviados. ....	41
Figura 16	Eleição dos clustheads em ilhas desconexas. ....	42
Figura 17	Roteamento de pacote até o Clusthead. ....	44
Figura 18	Exemplo de UAVs enviando beacons na RSSF. ....	48
Figura 19	UAV percorrendo posições de uma lista usando o modelo ordem de chegada. ....	50
Figura 20	UAV utilizando o modelo mais próximo primeiro. ....	51
Figura 21	Geração de ilhas. ....	53
Figura 22	Gráfico: Quantidade de pacotes perdidos. ....	58
Figura 23	Gráfico: Tempo médio de Round do UAV. ....	60
Figura 24	Gráfico: Tempo médio de entrega de pacotes no destino. ....	62

## LISTA DE TABELAS

Tabela 1	Caracterização das falhas de acordo com sua causa.....	31
Tabela 2	<i>Parâmetros para a simulação do Restabelecimento da Conectividade de RSSFs Utilizando Nós Móveis para transportar pacotes.....</i>	57
Tabela 3	<i>Comparação: Quantidade de pacotes perdidos. ....</i>	58
Tabela 4	<i>Comparação: Tempo médio de Round UAV <b>m/s</b>. ....</i>	59
Tabela 5	<i>Comparação: Tempo médio de “round” do UAV do trabalho de restabelecimento de comunicação usando UAV [51]. ....</i>	60
Tabela 6	<i>Comparação: Tempo médio <b>m/s</b> de entrega de mensagens.....</i>	60
Tabela 7	<i>Comparação: Tempo médio gasto pelo pacote para chegar ao destino do trabalho de restabelecimento de comunicação usando UAV [51]. ....</i>	61

## SUMÁRIO

<b>1</b>	<b>Introdução e Motivação .....</b>	<b>12</b>
<b>1.1</b>	<b>Objetivos .....</b>	<b>14</b>
<b>1.1.1</b>	<b>Ojetivos Específicos .....</b>	<b>14</b>
<b>1.2</b>	<b>Organização do Trabalho .....</b>	<b>15</b>
<b>2</b>	<b>Metodologia.....</b>	<b>16</b>
<b>2.1</b>	<b>Procedimentos Metodológicos .....</b>	<b>16</b>
<b>3</b>	<b>Referencial Teórico .....</b>	<b>17</b>
<b>3.1</b>	<b>Redes Móveis Ad hoc .....</b>	<b>17</b>
<b>3.1.1</b>	<b>Definições .....</b>	<b>17</b>
<b>3.1.2</b>	<b>Aplicações de Redes Móveis Ad hoc .....</b>	<b>18</b>
<b>3.2</b>	<b>Redes Tolerantes a Interrupções (Delay Tolerant Networks - DTN).....</b>	<b>19</b>
<b>3.2.1</b>	<b>Características .....</b>	<b>21</b>
<b>3.2.2</b>	<b>Arquitetura de DTN .....</b>	<b>22</b>
<b>3.2.3</b>	<b>Aplicações.....</b>	<b>23</b>
<b>3.3</b>	<b>Redes de Sensores Sem Fio .....</b>	<b>24</b>
<b>3.3.1</b>	<b>Definições .....</b>	<b>24</b>
<b>3.3.2</b>	<b>Características .....</b>	<b>25</b>
<b>3.3.3</b>	<b>Aplicações de Redes de Sensores Sem Fio .....</b>	<b>27</b>
<b>3.3.4</b>	<b>Desafios em RSSFs.....</b>	<b>28</b>
<b>3.3.5</b>	<b>Nós Sensores.....</b>	<b>29</b>
<b>3.3.6</b>	<b>Tolerância a falhas em RSSF .....</b>	<b>29</b>
<b>3.4</b>	<b>Roteamento Geográfico .....</b>	<b>31</b>
<b>3.5</b>	<b>Veículos Aéreos não Tripulados (UAVs) .....</b>	<b>33</b>
<b>3.5.1</b>	<b>Tipos de UAVs .....</b>	<b>34</b>
<b>3.5.2</b>	<b>Missões utilizando UAVs .....</b>	<b>35</b>
<b>3.5.3</b>	<b>Pesquisas em UAVs.....</b>	<b>36</b>
<b>3.6</b>	<b>Simulação Orientada a Eventos .....</b>	<b>37</b>
<b>3.6.1</b>	<b>Definições Simulação Orientada a Eventos .....</b>	<b>37</b>
<b>3.7</b>	<b>Simulador Orientado a Eventos - GRUBiX.....</b>	<b>38</b>
<b>3.7.1</b>	<b>Definição .....</b>	<b>38</b>
<b>4</b>	<b>Algoritmo de restabelecimento da conexão .....</b>	<b>39</b>
<b>4.1</b>	<b>Descrição do Cenário .....</b>	<b>39</b>
<b>4.2</b>	<b>Algoritmo de eleição dos clustheads .....</b>	<b>40</b>
<b>4.3</b>	<b>Roteamento de Pacotes .....</b>	<b>43</b>
<b>4.4</b>	<b>Descoberta dos Clustheads.....</b>	<b>47</b>
<b>4.4.1</b>	<b>Round do UAV.....</b>	<b>49</b>

<b>4.5</b>	<b>Trajetória do UAV até o clusthead .....</b>	<b>49</b>
<b>4.5.1</b>	<b>Ordem de chegada .....</b>	<b>50</b>
<b>4.5.2</b>	<b>Mais Próximo Primeiro .....</b>	<b>51</b>
<b>4.6</b>	<b>Tamanho do Buffer utilizado nos clustheads .....</b>	<b>52</b>
<b>4.7</b>	<b>Algoritmo para geração das ilhas .....</b>	<b>52</b>
<b>5</b>	<b>Resultados e Discussões .....</b>	<b>56</b>
<b>5.1</b>	<b>Construção da simulação .....</b>	<b>56</b>
<b>5.2</b>	<b>Parâmetros da simulação .....</b>	<b>56</b>
<b>5.3</b>	<b>Resultados .....</b>	<b>57</b>
<b>6</b>	<b>Conclusão .....</b>	<b>63</b>

## 1 Introdução e Motivação

A utilização de Redes de Sensores Sem Fio (RSSFs)<sup>1</sup> surgiu da necessidade de monitorar ambientes de risco onde a presença humana é inviável [22]. As redes de sensores sem fio têm sido cada vez mais utilizadas em diversas aplicações, sendo elas científicas, comerciais e militares. Como exemplo dessas aplicações podemos citar o monitoramento para controle de riscos (incêndios, enchentes e abalos sísmicos), monitoramento de áreas de interesse militar, automação industrial, áreas de difícil acesso ou até inacessíveis a pessoas, como áreas de risco (de contaminação, com atividades vulcânicas, altas ou baixas temperaturas).

As condições em ambientes de monitoramento levam a falhas na rede devido a uma série de condições adversas, tais como interferências de comunicação. Estas falhas podem criar ilhas de nós isolados, que não podem informar os dados adquiridos por eles para outros nós. Este tipo de situação pode ser transitória ou permanente, dependendo das causas das falhas [56].

Contudo, segundo [17], nós sensores terrestres apresentam algumas limitações, principalmente em relação ao raio de comunicação de cada nó. Em situações como esta, o uso de sensores móveis pode provocar melhorias significativas. Os sensores móveis podem prover habilidades para que a rede possa se adaptar dinamicamente aos eventos ocorridos no ambiente, além de colaborar com o aumento da conectividade dentro da rede.

Em um ambiente monitorado por uma rede de sensores, a ocorrência de um ponto de desconexão pode afetar a comunicação e comprometer a estrutura desta rede. Surge assim a necessidade de restabelecer a conexão desta rede. A utilização de veículos aéreos não tripulados (UAVs)<sup>2</sup> é uma alternativa para resolução

---

<sup>1</sup>RSSFs :Redes de sensores sem fio.

<sup>2</sup>UAVs :Veículos aéreos não tripulados.

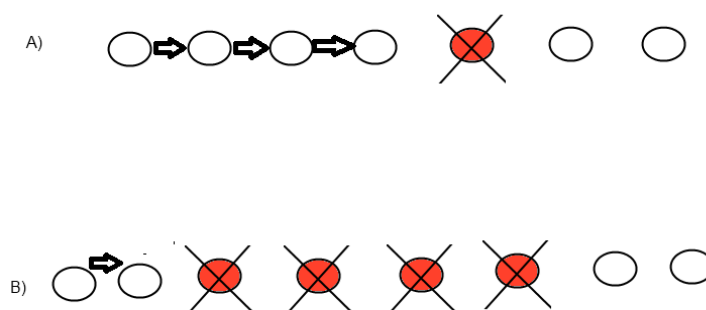
deste problema. Nesta situação, os UAVs tem a função de transportar os pacotes de regiões isoladas sem conectividade até regiões operantes.

O trabalho continua uma pesquisa já realizada anteriormente, **Restaurando a conectividade de RSSFs através de nós móveis** [51]. Essa pesquisa tem como objetivo restabelecimento da conexão em redes de sensores sem fio em uma região onde os nós estavam uma distância de um salto de outro nó, como mostrado na figura 1 a, e o UAV servia de ponte para restabelecer a conexão. A solução proposta neste trabalho é o restabelecimento da conexão de redes tolerantes a atrasos e desconexões(DTN), onde os nós tem saltos maiores como mostrado na figura 1 b, utilizando o paradigma de *store-carry-and-forward* como mostrado na figura 2, que utiliza máquinas intermediárias para armazenar os dados destinados a outros participantes até que possam ser entregues, ou seja, segundo [44], esse paradigma, cada vez que um nó da rede DTN encontra outro, as mensagens que cada um carrega são trocadas segundo regras e políticas estabelecidas para maximizar a probabilidade de entrega dentro de um tempo máximo determinado, e respeitando-se a capacidade de cada um. As trocas de mensagens podem levar em conta fatores como: capacidade computacional, memória, raio de cobertura, velocidade dos nós, probabilidade de encontro do nó destino em função do histórico de encontros anteriores, encontros totalmente previsíveis e itinerário.

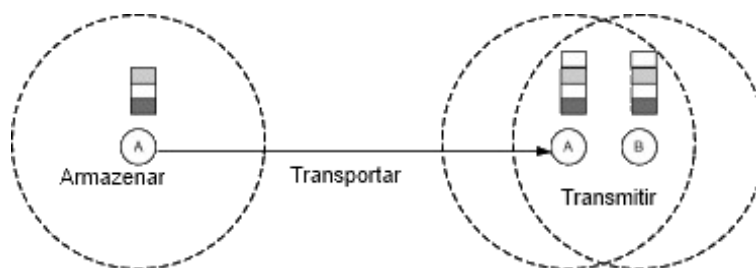
Para que este objetivo seja alcançado, serão utilizados UAVs como máquinas intermediárias, que atuando como <sup>3</sup>, entidades móveis e externas a uma RSSF, que ao se deslocarem pela área ocupada pela RSSF, coletam os pacotes em ilhas de sensores isoladas e desconexas e os entregam ao destino do pacote.

---

<sup>3</sup>*Data Mule* :Dispositivos móveis que armazenam e transmitem dados entre redes sem infraestrutura de transmissão de dados entre si.



**Figura 1:** Exemplo de salto em RSSFs. A figura "a" representa a desconectividade em um salto, enquanto a figura "b" aborda a desconexão em mais de um salto.



**Figura 2:** store-carry-and-forward.([www.sciencedirect.com/science/](http://www.sciencedirect.com/science/))

## 1.1 Objetivos

Este trabalho tem como objetivo estudar, adaptar e desenvolver algoritmos para restabelecimento da conexão em uma rede DTN quando existirem regiões desconexas distantes e isoladas.

### 1.1.1 Ojetivos Específicos

Este trabalho tem como objetivos específicos:

**Algoritmo de eleição de Clustheads:** Consiste no desenvolvimento de um algoritmo que elege um clusthead para cada ilha, sendo o mesmo responsável

por cada ilha. Qualquer pacote que um nó de uma ilha precise enviar para outra ilha, esse pacote tem que passar pelo seu clusthead para o mesmo entregar para o UAV, logo após o recebimento do pacote pelo UAV, esse pacote vai seguir para o nó destino.

**Algoritmo para Roteamento de Pacotes:** Adaptação de um algoritmo para rotear pacotes até o clusthead. Esse roteamento consiste em escolher melhor caminho entre um nó origem com o clusthead. Esse nó origem precisa mandar um pacote até o clusthead da sua ilha. O algoritmo calcula o nó vizinho mais próximo e envia, até chegar no clusthead. Este algoritmo também tem como objetivo criar uma rota para o UAV quando alguma desconexão acontecer. Esse UAV coleta o pacote em um clusthead, que precisa ser entregue no seu destino.

**Implementação e simulação dos algoritmos no ambiente de simulação GRUBIX:** Desenvolvimento e simulação dos algoritmos no ambiente de simulação GRUBIX, mantido pelo Grupo de Redes Ubíquas do Departamento de Ciência da Computação da Universidade Federal de Lavras.

## 1.2 Organização do Trabalho

O trabalho encontra-se organizado em cinco capítulos. No capítulo 1 apresenta uma introdução, a motivação, os objetivos e definição do problema estudado. A metodologia para realização do trabalho encontra-se no capítulo 2. No capítulo 3 podem ser encontradas as definições e bases teóricas para o entendimento do problema. O capítulo 4 demonstra os algoritmos usados para resolver o problema, no capítulo 5 os resultados obtidos e a conclusão no capítulo 6.



## **2 Metodologia**

Esta pesquisa é considerada uma pesquisa aplicada, visto que ela procura resolver um problema. Ao fim deste trabalho espera-se resultados construtivos para o problema tratado. Quanto aos seus objetivos a pesquisa é considerada exploratória, e visa restabelecer a conexão em RSSFs combinando conceitos existentes para resolução do problema. É considerada uma pesquisa experimental porque tem a aplicação de práticas e técnicas e visa buscar o desenvolvimento de algoritmos e técnicas de restabelecimento de conexão. A pesquisa é considerada quantitativa pois foram realizados testes, simulações e algumas avaliações.

Todo o trabalho desde o desenvolvimento, testes e simulações foi realizado no laboratório do GRUBiX, pois dessa forma todos os parâmetros do experimento puderam ser controlados. As simulações foram baseadas em eventos a partir do simulador GRUBiX, o que permitiu a realização de vários testes e a avaliação do algoritmo desenvolvido.

### **2.1 Procedimentos Metodológicos**

Todos os experimentos deste trabalho foram baseado em simulações. O uso da simulação é justificado pela dificuldade em aplicar os algoritmos em grandes quantidades de nós sensores e UAVs, por motivos financeiros e de tempo.

### 3 Referencial Teórico

Este capítulo apresenta a revisão bibliográfica que foi utilizada para realização deste trabalho.

#### 3.1 Redes Móveis Ad hoc

Nesta seção, será abordado as definições e aplicações de Redes Móveis *Ad hoc*.

##### 3.1.1 Definições

As redes *ad hoc* segundo [1] são redes que são implantadas sem uma infraestrutura fixa existente. Em uma rede *ad hoc*, os dispositivos móveis se comunicam entre si, transmitindo a mensagem através vários dispositivos. Neste caso, cada dispositivo móvel funciona como um usuário e um switch de roteamento, ao mesmo tempo. Normalmente, redes *ad hoc* são redes que se estabelecem em uma pequena área geográfica em uma situação de emergência.

Essas redes trouxeram um grande benefício, logo foram estendidas para lugares onde é muito difícil, ou até impossível a montagem de uma infra estrutura fixa. Como exemplos podemos citar florestas densas, desertos, e em atividades de sensoriamento onde existe o risco de morte para seres humanos [38].

A Figura 3 mostra três exemplos de uma Rede *Ad hoc*.

Essas redes possuem várias aplicações que vêm impulsionando seu desenvolvimento e sua utilização.



**Figura 3:** Redes Ad hoc. [25]

### 3.1.2 Aplicações de Redes Móveis Ad hoc

Essas redes podem ser construídas rapidamente, sem a necessidade de uma infraestrutura fixa, sendo assim, elas se tornam adequadas a situações e locais onde redes cabeadas não são viáveis. Dentre as possíveis aplicações das redes móveis ad hoc, podemos citar as seguintes [11]:

**Aplicações Militares:** as aplicações militares deram origem as redes ad hoc com a tarefa de proporcionar a comunicação de combatentes no campo de batalha, visto que tais ambientes não possibilitam a comunicação por meio de uma infraestrutura fixa;

**Redes de Sensores:** com o aumento da popularidade do uso de sensores no meio ambiente, temos a possibilidade de operar automaticamente a monitora-

ção e coleta de dados. Também podemos fazer a transmissão dessas informações sem que haja uma infraestrutura fixa. Esse tipo de rede será tratado mais detalhadamente adiante neste trabalho.

**Wireless Personal Area Network (WPAN):** rede onde vários dispositivos, como celulares, estão conectados a uma rede sem fio disponibilizada em um alcance pequeno, normalmente alguns metros.

**Veículos Aéreos Não Tripulados (UAVs):** muitas aplicações de redes ad hoc necessitam de componentes robóticos não tripulados. Em uma rede genérica, todos os nós são por definição capazes de trabalhar autonomamente. Nesse contexto, quando a mobilidade autônoma é inserida, tem-se a vantagem de poder trabalhar com redes e o movimento. Um exemplo bem simples e que é abordado nesse trabalho são os UAVs que podem ajudar na manutenção da conectividade de uma rede ad hoc.

Segundo [9] as redes ad hoc representam sistemas distribuídos complexos que incluem nós móveis sem fio podendo se organizar livre e dinamicamente de forma arbitrária, dentro de uma topologia de rede temporária.

### **3.2 Redes Tolerantes a Interrupções (Delay Tolerant Networks - DTN)**

Nos últimos anos, a Internet vem se difundindo rapidamente pelo mundo inteiro. Essa difusão ocorre principalmente através de redes e subredes conectadas por cabos sendo executadas sobre a pilha de protocolos TCP/IP. Entretanto, essa arquitetura não satisfaz alguns cenários, como por exemplo comunicações em redes entre sensores sem fio, redes móveis ou ainda redes em áreas rurais esparsas e interplanetárias [42]. Quando um desses fatores é comprometido, a comunicação pode se tornar inviável. Por essa razão, as redes TCP/IP necessitam de infraestruturas estáveis, o que nem sempre é possível em vários tipos de cenário, como:

- **Em regiões remotas** : onde não haja infraestrutura de telecomunicações permanentemente disponível;
- **Em redes de veículos** : onde não haja cobertura disponível ao longo de todas as rodovias e lugares visitados;
- **Em regiões de desastre** : onde a infraestrutura existente é total ou parcialmente destruída, como em terremotos, enchentes e outros;
- **Em redes de sensores sem fio** : onde os nós precisam economizar energia e, por isso, permanecem desligados periodicamente, causando o particionamento da rede e conectividade intermitente;
- **Em redes Ad-hoc móveis** : (Mobile Ad hoc NETWORKS - MANETs), onde a topologia da rede pode mudar constantemente, devido a grande mobilidade dos nós, provocando frequentes desconexões;
- **Em redes interplanetárias** : onde um corpo celeste pode interromper a comunicação por várias horas, ou onde a rotação de um planeta pode deixar a estação base periodicamente fora de alcance;

Por este motivo, há a necessidade de uma nova arquitetura para redes tolerantes a atrasos e desconexões (DTN),<sup>4</sup> que permita uma comunicação com longos atrasos e períodos sem conectividade entre os nós.

Segundo [42] a baixa densidade de nós, a conectividade intermitente, altas taxas de erros de comunicação, alta latência, limitações de banda e de energia dos nós surge ambientes conhecidos como *Challenged Networks*.

Redes (DTNs) permitem transmissões de dados entre dispositivos mesmo que eles estejam inalcançáveis durante um período [46]. São recomendadas solu-

---

<sup>4</sup>DTN :Delay and Disruption Tolerant Networks.

ções seguindo a arquitetura de redes tolerantes a atrasos e desconexões para permitir comunicação no caso de desconexão abordado neste trabalho.

### 3.2.1 Características

As principais características encontradas na DTNs estão relacionadas às desconexões e aos longos atrasos na entrega de mensagens:

- **os atrasos longos e/ou variáveis:**

Uma DTN pode ter atrasos de horas, ou mesmo de dias. O atraso fim-a-fim é encontrado através da soma dos tempos de atraso salto-a-salto e é formado por quatro componentes:

- Tempo de espera, que corresponde ao tempo de espera de cada nó pelo nó de destino ou pela chegada de um nó intermediário que possa encaminhar as suas mensagens;
- Atraso nas filas, que corresponde ao tempo de espera de cada nó na fila antes de uma mensagem corrente ser entregue;
- Atraso de transmissão da mensagem;
- Atraso de propagação, que corresponde ao tempo de propagação do sinal (latência).

- **Desconexões comuns:**

As perdas de conexão acontecem pelas constantes mudanças da topologia da rede e pelas condições ruins de comunicação, baixa densidade dos nós, quando os sensores estão hibernados para poupar energia, por negação de serviço como o ato do inimigo sujar a frequência (jamming) em operações

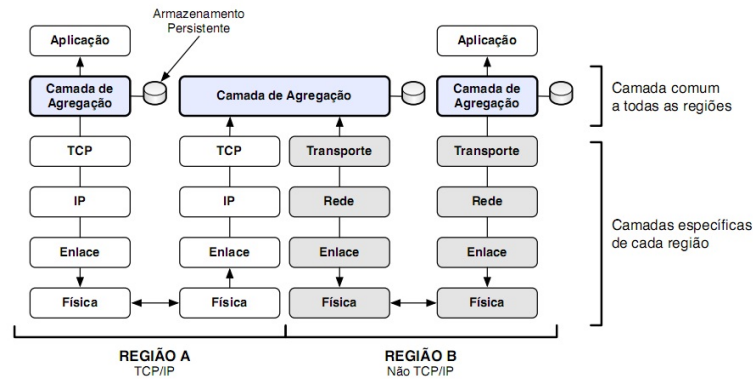
militares. Todos esses eventos podem gerar um caminho fim-a-fim inexistente entre um nó origem e um nó destino.

Esse tipo de problema em uma rede tradicional seria resolvido acrescentando uma rede adicional, cabeamento ou hardware. Em uma rede DTN, que não possui uma infra-estrutura organizada, podemos adicionar apenas dois tipos de nós. A primeira opção seria nós fixos estrategicamente localizados na rede, porém como se trata de uma DTN onde a topologia está em constante mudança, essa solução não seria tão eficaz. A outra opção seria adicionar nós móveis a rede, visto que estes poderiam aumentar a capacidade reativas em áreas de maior demanda [53].

### **3.2.2 Arquitetura de DTN**

Segundo [45], a arquitetura DTN prevê a utilização da técnica de comutação de mensagens e o armazenamento persistente dos dados definindo uma sobre-camada (overlay) abaixo da camada de aplicação. Esta nova camada é denominada camada de agregação (Bundle Layer) e o protocolo de agregação é executado em todos os nós pertencentes à rede DTN, denominados nós DTN, da origem até o destino, à semelhança da camada IP. As “sub-redes” são denominadas redes regionais e a arquitetura em sobre-camada permite tornar a DTN totalmente independente das diversas redes regionais, permitindo que as aplicações se comuniquem através de múltiplas regiões. Como ilustrado na figura 4 abaixo, as camadas abaixo da camada de agregação são definidas de acordo com a conveniência do ambiente de comunicação de cada região, podendo ser específicas para cada região englobada pela DTN.

As unidades de dados da aplicação são mensagens de tamanhos variáveis que são enviadas pelas aplicações DTN (Application Data Units - ADUs). Na ca-



**Figura 4:** Exemplo da camada de agregação da arquitetura DTN [45].

mada de agregação acontece a transformação das mensagens em uma ou mais unidades de dados de protocolo (Protocol Data Units - PDUs) denominadas agregados (bundles), que são armazenados e encaminhados pelos nós DTN. Múltiplas cópias do mesmo agregado podem existir simultaneamente em diferentes partes da rede, tanto na memória local de um ou mais nós DTN quanto em trânsito entre os nós. Para evitar a associação a algum tipo de interatividade que é ineficiente em ambientes de longos atrasos e desconexões, foi escolhido o termo agregado em DTN, ao invés de transação. Nestes ambientes, por exemplo, o pedido de transferência de um arquivo pode ser enviado contendo os dados necessários para a autenticação do usuário (ex. login/senha), o nome do arquivo desejado e o diretório local onde o arquivo deve ser entregue [43].

### 3.2.3 Aplicações

Segundo [43] “O desenvolvimento de aplicações para DTNs deve levar em conta tanto a existência de cenários desconectados como a pouca interatividade. Os problemas mais comuns no desenvolvimento de aplicações em DTNs estão relacionados aos diferentes temporizadores da camada de aplicação. Ge-



almente, os temporizadores da aplicação são usados para os casos em que uma transação é enviada e a resposta não é obtida dentro do tempo esperado, nesses casos, uma nova transação é emitida. O exemplo mais comum é o acesso a páginas web, no qual o navegador espera um certo tempo por uma resposta às requisições enviadas pelo usuário. Após expirar o tempo, novas tentativas são realizadas até que seja retornada uma mensagem de sucesso ou de erro, quando não é possível o estabelecimento. Desta forma, torna-se inviável a utilização de servidores web em estações pertencentes a uma DTN, pois a conectividade não é permanente".

A entrega da mensagem realizada com sucesso é mais importante que qualquer outra métrica de desempenho, inclusive o atraso para aplicações DTN. A aplicação mais imediata de ser implementada em DTNs é o correio eletrônico (email), que possibilita que arquivos grandes sejam anexados. Outros tipos de aplicações DTNs também podem ser desenvolvidos, como transferência de arquivos, educação à distância, formulários eletrônicos, coleta de informações (votação, censo), vídeos, páginas web pessoais, jornais e revistas [43].

### **3.3 Redes de Sensores Sem Fio**

Esta seção aborda definições, características, aplicações e os principais desafios envolvendo as Redes de Sensores Sem Fio, visto que o problema na comunicação dessas redes é o principal problema abordado neste trabalho.

#### **3.3.1 Definições**

Segundo [33], as RSSFs podem ser vistas como um tipo especial de rede móvel ad hoc (MANET - Mobile Ad Hoc Network).

Do ponto de vista de organização, RSSFs e MANETs são iguais pelos seus elementos computacionais se comunicarem através de um protocolo de infra-

estrutura móvel. Os componentes das redes MANETs realizam tarefas distintas, enquanto que nas RSSFs seus componentes realizam tarefas em conjunto.

As Redes de sensores sem Fio (RSSFs) são compostas de nós sensores distribuídos em uma determinada região a ser monitorada como mostrado na figura 5.

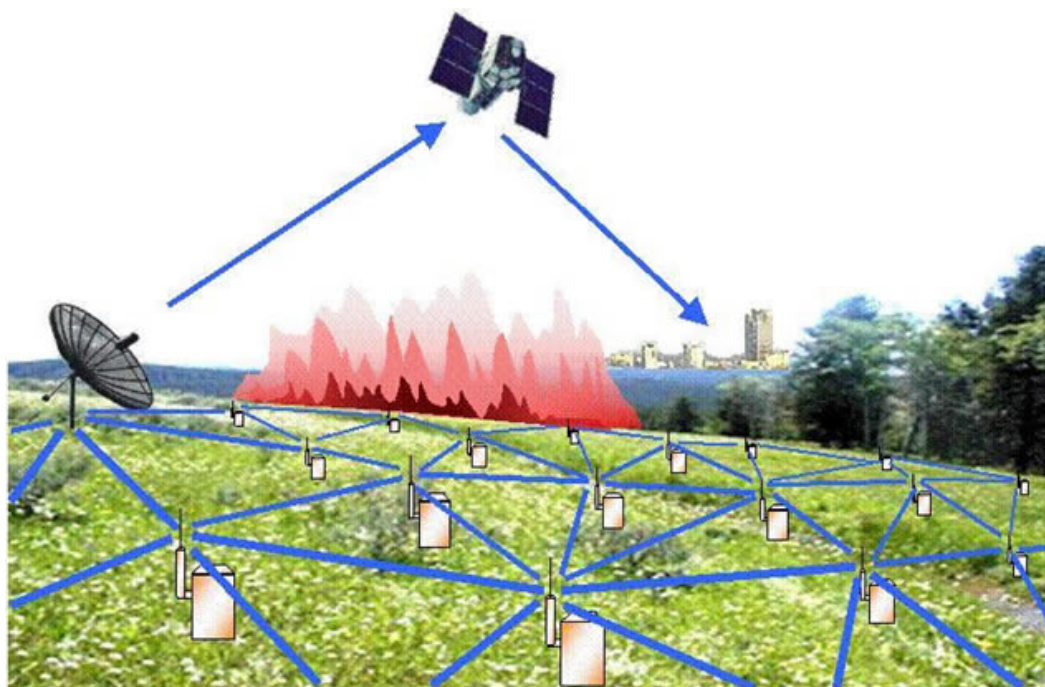
Nas redes de sensores, cada nó está equipado com um ou mais sensor, responsáveis por realizar o monitoramento. Como muitas vezes as baterias desses nós não são recarregáveis, as redes funcionam enquanto a alimentação estiver disponível ou deixa de funcionar quando a energia é desligada [1].

Tipicamente, as redes de sensores podem funcionar em dois modos. O primeiro de operação contínua e o segundo um modo de consulta. No modo de operação contínua, o nó sensor monitora continuamente o ambiente enviando as medições para um nó vizinho ou um nó base. Já no modo de consulta, o nó sensor permanece na maioria do tempo em ciclo de trabalho à espera de um sinal de um nó base ou um nó vizinho. Quando este nó recebe este sinal, ele coleta os dados do sensor, processa e envia para o nó solicitante [1].

### 3.3.2 Características

Segundo [33], as redes de sensores possuem várias características, dentre as quais podemos citar:

- **Dispersão relativa dos nós sensores** : característica comum também às redes ad hoc, onde a rede é formada por nós espalhados em uma região geograficamente grande. Este fato impossibilita a comunicação single-hop (salto único).
- **Homogeneidade** : Uma rede de sensor sem fio, diferente das redes ad hoc, possuem nós com as mesmas características.



**Figura 5:** Rede de sensores sem fio. [29]

- **Mobilidade de sensores :** Os nós sensores são classificados de acordo sua mobilidade em nós fixos ou nós móveis: podem ou não ser móveis de acordo o sistema que estão monitorando. Por exemplo, sensores colocados em uma floresta para medir temperatura e umidade são tipicamente estáticos e em um oceano para medir poluição são móveis.
- **Restrições de dados coletados :** Determina se os dados coletados pelos sensores têm algum tipo de restrição com um intervalo de tempo máximo para disseminação de seus valores em dada entidade de supervisão.
- **Quantidade de sensores:** As RSSFs podem conter de poucas a várias dezenas de sensores ou até mesmo milhares. Em redes com muitos sensores a escalabilidade torna-se uma questão muito importante.

- **Limitação de energia:** Na maioria dos casos as RSSFs são distribuídas em áreas remotas e de difícil acesso. A capacidade de energia do sensor limita-se a somente ao tempo de bateria disponível, pois a manutenção da rede nem sempre é possível.
- **Auto-Organização na rede:** A rede deve ser capaz de recuperar de possíveis falhas ou imprevistos, por exemplo, caso um nó venha a ser removido por fator qualquer, a rede tem que estar preparada para continuar com suas tarefas normalmente.
- **Tarefas colaborativas:** Quando existe necessidade de execução de tarefas complexas em RSSFs, um nó não tem capacidade suficiente para execução da tarefa. Neste caso, torna se necessário um comportamento colaborativo na rede, onde vários nós desempenham em conjunto as tarefas em questão.
- **Capacidade de resposta a consultas:** As consultas são feitas direcionadas para um nó, um conjunto de nós ou até mesmo para rede toda. As RSSFs devem ser capazes de responder a essas requisições e consultas.

### 3.3.3 Aplicações de Redes de Sensores Sem Fio

Houve um grande avanço tecnológico nas área de sensores, circuitos integrados e comunicação sem fio, levando assim a uma maior utilização dessas RSSFs.

Este tipo de rede pode ser aplicada no monitoramento, rastreamento, coordenação e processamento em diferentes contextos. Como exemplo podemos citar o monitoramento e controle das condições ambientais numa floresta ou em uma determinada área de interesse. Segundo [1], as RSSFs são usadas para coletar

ou rastrear determinadas informações, como medição de temperatura ou pressão, rastreamento de alvos, vigilância de fronteira e monitoração de produtos tóxicos.

As RSSFs possuem várias aplicações, dentre as quais podemos citar:

- No setor industrial no controle da graduação do etanol produzido por uma usina de álcool;
- No setor de segurança pública, com monitoramento do tráfego em rodovias e localização de veículos roubados;
- No setor médico no monitoramento de sinais vitais de pacientes (pressão, temperatura, glicemia);
- No setor ambiental com o monitoramento de animais, florestas, desastres com óleo e ou produtos químicos em oceanos e rios;
- No setor militar com o monitoramento de áreas inimigas, detecção de gases venenosos, radioatividade, explosões, entre outros;
- No setores domésticos com monitoramento de residências, nas questões climáticas, segurança e gerenciamento doméstico.

### 3.3.4 Desafios em RSSFs

De acordo com [10], muitos desafios são encontrados em RSSFs, dentre eles podemos citar:

- **Confiabilidade da comunicação sem fio:** Em muitos casos, especialmente com um número crescente de nós no mesmo raio de comunicação, a comunicação sem fio tende a não ser confiável. A principal causa deste fenômeno é a grande quantidade de colisões entre os pacotes trocados entre os nós. A

probabilidade de colisão é proporcional à densidade da rede, o tamanho e o tráfego gerado. Confiabilidade é a capacidade de prover garantia de entrega dos pacotes.

- **Baixa qualidade de comunicação:** Nós sensores operam e são dispostos em ambientes com condições ruins e de mau tempo. Nesse caso, os nós sensores devem ser bem projetados e tolerantes a falhas, pois em locais como esse a qualidade de comunicação pode ser instável.
- **Requisitos de tempo real:** Recai sobre a necessidade das aplicações, em alguns casos, necessitarem de informações em tempo real. Neste caso, torna-se fundamental que a rede de sensores seja capaz de prover informações confiáveis em tempo real.
- **Mobilidade espaço-temporal:** Mobilidade espacial refere-se à movimentação geográfica dos nós da rede, i.e. mudanças da localização dos nós ao decorrer do tempo.

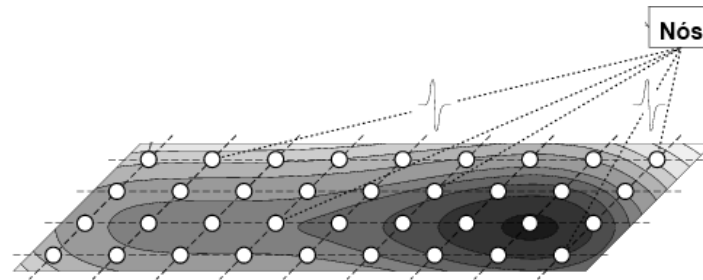
### 3.3.5 Nós Sensores

Os nós sensores, quando dispostos em uma rede de modo ad hoc formam assim uma rede de sensores. Esses nós coletam e processam informações localmente. Na figura 6 são mostrados exemplos de nós sensores [33].

### 3.3.6 Tolerância a falhas em RSSF

Podemos dividir as falhas em RSSFs em dois tipos:

- Silenciosas;
- Bizantinas; m impasse e dois ou mais processos ficam impedidos de continuar suas execuções, ou seja, ficam bloqueados.



**Figura 6:** Exemplo de nós sensores dispostos em uma rede. [30]

As falhas silenciosas são representadas pela não saída de resultados externos, e podem acontecer por fatores externos ou internos a rede, como fenômenos atmosféricos, bloqueio do processador (Deadlock)<sup>5</sup>, esgotamento de bateria e desastres naturais que representam bem o cenário onde acontecem as falhas do problema abordado. Resumindo, as falhas silenciosas são ocasionadas pela não recepção de pacotes ou erros de hardware nos nós sensores dentro de uma RSSF.

Nas falhas bizantinas ocorrem as saídas com resultados inconsistentes, podendo estes resultados estarem corretos ou incorretos, porém este não é o foco do trabalho.

As falhas podem ser categorizadas quanto a sua extensão e persistência como mostrado na tabela 1 [22].

- **Persistência:** A falha é classificada como persistente quando mostra se um nó falho voltará a funcionar depois de um determinado tempo (Transiente) ou se a falha é permanente. A falha é considerada transiente quando a mesma durar minutos, caso contrário é considerada permanente.

<sup>5</sup>*Deadlock* : Um impasse e dois ou mais processos ficam impedidos de continuar suas execuções, ou seja, ficam bloqueados.

Causa da falha	Persistência	Extensão
Fatores móveis de interferência	transiente	isolada
Fenômenos atmosféricos	Permanentes	Agrupadas
Desastres naturais	Permanentes	Agrupadas
Bloqueio do processador	Permanente	isolada
Ataques de colisão	Ambas	isolada
Quebra acidental	Permanente	isolada

**Tabela 1:** Caracterização das falhas de acordo com sua causa

- **Extensão:** A falha é classificada como de extensão quando mostra a quantidade de nós afetados. A complexidade da falha depende do número de nós afetados e da vizinhança do nó falho. As falhas podem ser isoladas ou agrupadas.

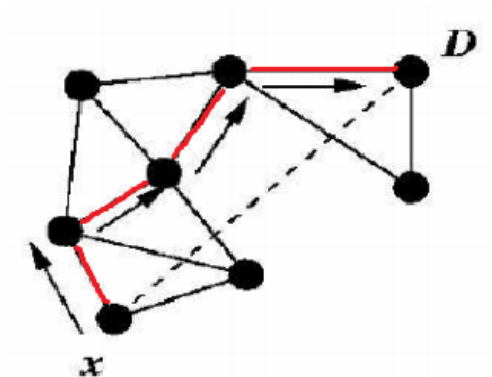
### 3.4 Roteamento Geográfico

Segundo [50], o Roteamento Geográfico é baseado em informações de posição geográfica. Os protocolos de roteamento geográfico utilizam informações geográficas obtidas através de um sistema de GPS (Global Positioning System) ou através de um sistema local, válido somente para os sensores da rede. Como exemplo, pode-se citar os protocolos *GeoMote GEAR*, entre outros [48]. O roteamento geográfico visa encaminhar os pacotes até o destino, utilizando como endereço, a localização geográfica dos nós. Dessa maneira, é possível rotear os pacotes de maneira escalável, pois cada nó armazena apenas os estados dos seus vizinhos [49].

A figura 7, mostra o caminho utilizado por um pacote roteado do nó x ao nó D.

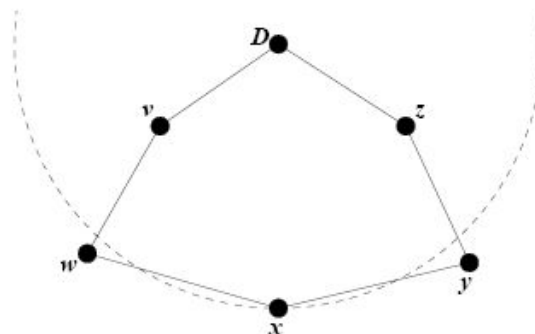
O protocolo *Geographic forwarding* utiliza o vizinho mais próximo do destino como próximo salto. Essa abordagem gulosa muitas vezes nos leva a si-





**Figura 7:** Utilizando Roteamento Geográfico para enviar do nó  $x$  ao nó  $D$ . [49]

tuações onde o nó não possui nenhum vizinho mais próximo do seu destino do que ele próprio. Essa situação é chamada dead-end e está apresentada na Figura 8 segundo [52].



**Figura 8:** Exemplo de Dead End em uma RSSF. [52]

Sendo  $x$  o nó vizinho mais próximo do nó  $D$ , porém o caminho possível passa pelos nós  $w$  e  $y$  que estão mais distante.

Segundo [49], os protocolos de roteamento geográfico vêm se tornando cada vez mais importantes em ambientes sem fio, principalmente em ambientes

onde haja variação da topologia, como redes ad-hoc. Dentre as principais vantagens de se usar o roteamento geográfico, temos o fato de que a necessidade de armazenar estados é mínima, um nó conhece apenas a sua localização e de seus vizinhos (em alguns casos de seus vizinhos dois saltos distantes), o que aumenta em muito a escalabilidade.

### 3.5 Veículos Aéreos não Tripulados (UAVs)

Os veículos Aéreos Não Tripulados recebem várias denominações, como VRP<sup>6</sup>, VANT<sup>7</sup>, UAV<sup>8</sup>, dentre muitas outras. UAVs são veículos aéreos que não necessitam de um operador humano, podendo voar de forma autônoma ou sendo controlados remotamente. Com sua eficácia do seu uso em ambientes militares, surgiu interesse na utilização desses para outros fins [6].



**Figura 9:** Imagem do Predador. [27]

Segundo [2] UAVs são muito importantes em diversas aplicações dentre elas pode-se citar: vigilância, monitoramento e detecção de desastres (incêndios, erupções, dentre outros). Cada aplicação possui sua respectiva restrição, como o tempo e as condições climáticas do ambiente. Uma importante aplicação de UAVs seria o monitoramento de florestas em busca de focos de incêndio, ou até de áreas

<sup>6</sup>VRP : Veículo Remotamente Pilotado.

<sup>7</sup>VANT : Veículo aéreo Não Tripulado.

<sup>8</sup>UAV : Unnamed Aerial Vehicle.

de desmatamento. Na figura 10 é mostrada uma imagem de um UAV chamado de *Firebee*, usado para monitoramento.



**Figura 10:** Imagem do Firebee. [27]

Neste trabalho os UAVs serão agentes de transporte de pacotes para suprir a falha de comunicação em regiões onde existem nós inoperantes. A próxima seção abordará alguns tipos de UAVs.

### 3.5.1 Tipos de UAVs

Os UAVs podem ser divididos em dois grupos [16]:

- UAVs de grande porte;
- UAVs de pequeno porte.

**UAVs de grande porte:** UAVs de grande porte têm sistemas de lançamento e recuperação que podem ser separados dos seus sistemas de controle e exploração de dados. Os UAVs apresentam mecanismos que podem ser controla-

dos separadamente, como por exemplo, coleta das informações e comunicação via satélite.

Dentre os UAVs de grande porte podemos citar como exemplo:

- Predator;
- Global Hawk.



**Figura 11:** Imagem do Global Hawk. [28]

**UAVs de pequeno porte:** Este tipo de UAV é mais comumente utilizado para monitoramento. Os dados captados por estes UAVs são enviados geralmente de volta para o operador para ser processados em tempo real.

Dentre os UAVs de pequeno porte podemos citar como exemplo:

- Raven;
- Pointer;
- AR Drone.

### 3.5.2 Missões utilizando UAVs

Os UAVs são usados em várias situações de monitoramento de desastres. Como exemplo dessas situações pode-se citar a utilização de UAVs controladas por

três operadores (um piloto, um diretor de vôo, e um especialista da missão) para inspecionar edifícios que desabaram após o furacão Katrina. Outro exemplo que podemos citar da utilização de UAVs em monitoramento de desastres seria após o terremoto em L'Áquila, onde os UAVs foram equipados com câmeras para avaliar toda região afetada. Estas são algumas situações onde temos o uso de UAVs em monitoramento de desastres [2].

### 3.5.3 Pesquisas em UAVs

Como as RSSFs apresentam limitações, os nós móveis como na figura 12 vem como uma forma de diminuir essa limitação da rede. Existe uma melhora significativa na integração de nós móveis a uma RSSF [17].

Neste contexto, a utilização de nós móveis para realizar a comunicação de um nó a outro, quando houver falha de comunicação entre eles surge como alternativa para resolução do problema proposto por este trabalho.



**Figura 12:** Foto do AR Drone. [26]

## **3.6 Simulação Orientada a Eventos**

Esta seção mostra algumas definições de Simulação Orientada a Eventos e do simulador utilizado neste trabalho.

### **3.6.1 Definições Simulação Orientada a Eventos**

Segundo [14], na simulação orientada a evento, um sistema é modelado pela definição das mudanças que ocorrem no tempo de evento do modelador é determinar os eventos que podem causar a mudança no estado do sistema e assim a lógica associada a cada tipo de evento pode ser desenvolvida. A simulação do sistema é produzida pela execução da lógica associada a cada evento em uma sequência ordenada no tempo.

Na simulação orientada a eventos existe um procedimento associado com cada tipo de evento no sistema. O simulador ciclicamente escala eventos, atualiza o relógio para o próximo evento a ocorrer e executa o procedimento associado ao evento. A execução no simulador envolve retirar o evento de uma lista global de eventos e processá-lo por um componente do sistema. O componente executará sua lógica e poderá gerar novos eventos. O tempo não é absoluto, mas uma quantidade virtual. Os eventos podem ser agendados para um determinado instante e esses serão ordenados por tempo de ocorrência. A ocorrência de um evento afeta o estado da simulação. Também são atualizados os contadores de estatísticas que permitem a geração dos relatórios da simulação [51].

A simulação orientada a eventos utiliza uma lista onde os eventos são armazenados por ordem cronológica de acontecimento. Os eventos vão sendo disparados no meio simulado de forma sequencial fazendo com que a representação do tempo seja apenas em função do atributo tempo de ocorrência dos eventos. Caso estes eventos não possuam uma variável explícita de tempo de aconte-

cimento, eles devem ser organizados simplesmente por suas ordens de acontecimentos [51].

### 3.7 Simulador Orientado a Eventos - GRUBiX

Nesta seção será definido o simulador utilizado neste trabalho (GRUBiX).

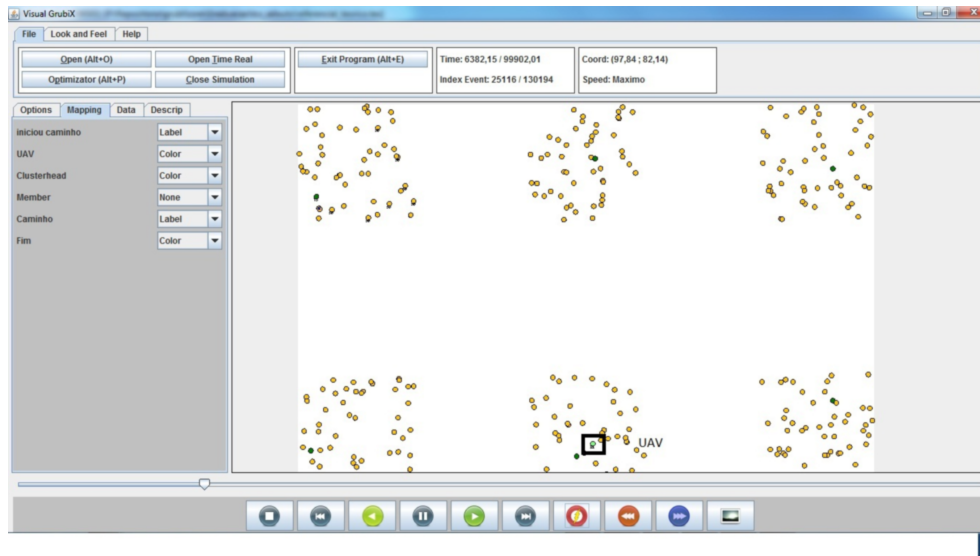
#### 3.7.1 Definição

O GRUBiX [15] é um ambiente de simulação Open Source mantido pelo Grupo de Redes Ubíquas do Departamento de Ciência da Computação da Universidade Federal de Lavras (GRUBi - UFLA). O simulador GRUBiX é identificado como um ambiente de simulação orientado a eventos e utiliza a linguagem de programação Java. O ambiente de simulação GRUBiX é uma extensão de um conhecido simulador para rede móveis ad hoc, o ShoX [ShoX: A scalable ad hoc network simulator].

O simulador GRUBiX tem como tarefa de avaliação e comportamento da rede. O GRUBiX tem uma versão estável e suas simulações dão origem a eventos que simulam o comportamento da rede [32].

Cada evento no simulador Grubix tem uma marca indicando quando ele deve ser processado e este evento pode ser definido em qualquer ponto de tempo.

A figura 13 mostra o visualizador do simulador GRUBiX mostrando o cenário abordado neste trabalho com vários nós. Os nós de cor verde escura representa os Clustheads, o UAV está marcado abaixo na figura e tem-se um fluxo de mensagem marcado com uma *label* de valor 50 em cada nó.



**Figura 13:** Visualizador do GRUBiX.

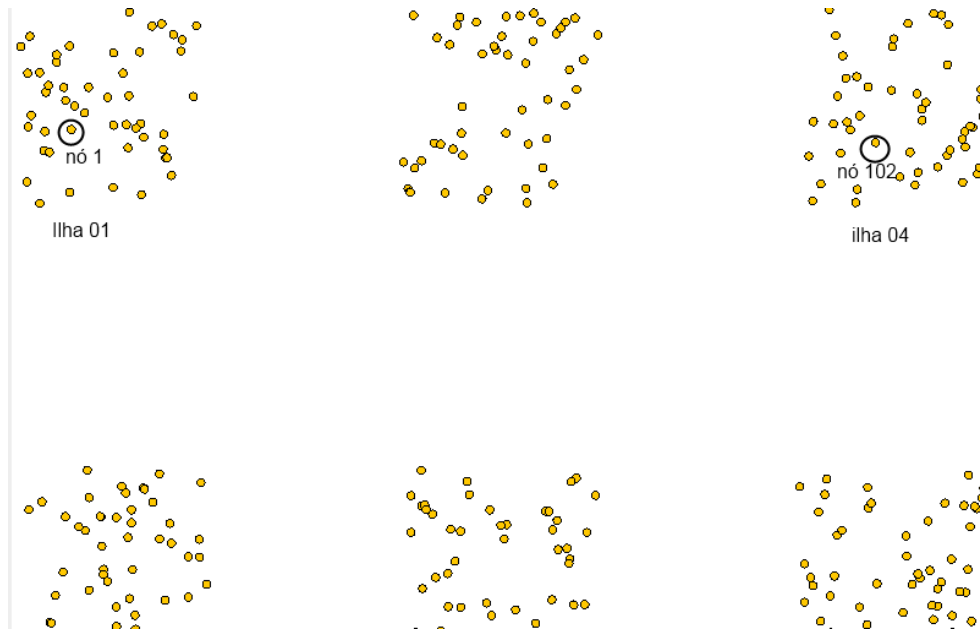
## 4 Algoritmo de restabelecimento da conexão

Este capítulo discute o funcionamento dos algoritmos desenvolvidos e adaptados para o restabelecimento de conexão utilizando UAVs. As próximas seções descrevem o problema tratado e apresentam a solução proposta para a resolução deste problema.

### 4.1 Descrição do Cenário

Tem-se o problema de desconexão em uma rede de nós distantes, onde os pacotes em uma região isolada (Ilha 01), como mostrado na figura 14, tem o nó 01 com pacotes de destino para o nó 102 da (Ilha 04) e não consegue chegar a seu destino devido a esta desconexão. Este é o principal problema abordado neste trabalho.





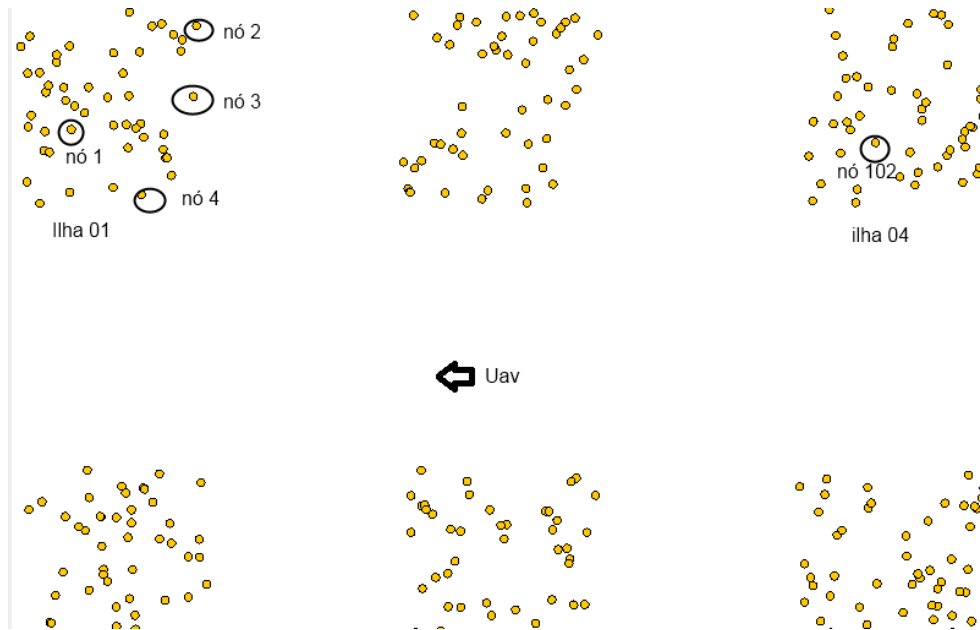
**Figura 14:** Exemplos das ilhas de desconexão em uma RSSF.

## 4.2 Algoritmo de eleição dos clustheads

O algoritmo de eleição dos clusthead em cada ilha surgiu da necessidade de centralizar as mensagens a serem enviadas em cada região isolada, afim de otimizar a rota do UAV.

Neste trabalho é tratada a necessidade do envio de pacotes por mais de um nó, ou seja, mais de um fluxo de mensagem. Considerando uma situação onde vários nós isolados precisam enviar mensagens uns aos outros, o tempo da trajetória do UAV aumentaria, uma vez que vários nós deveriam ser visitados, como exibido na figura 15.

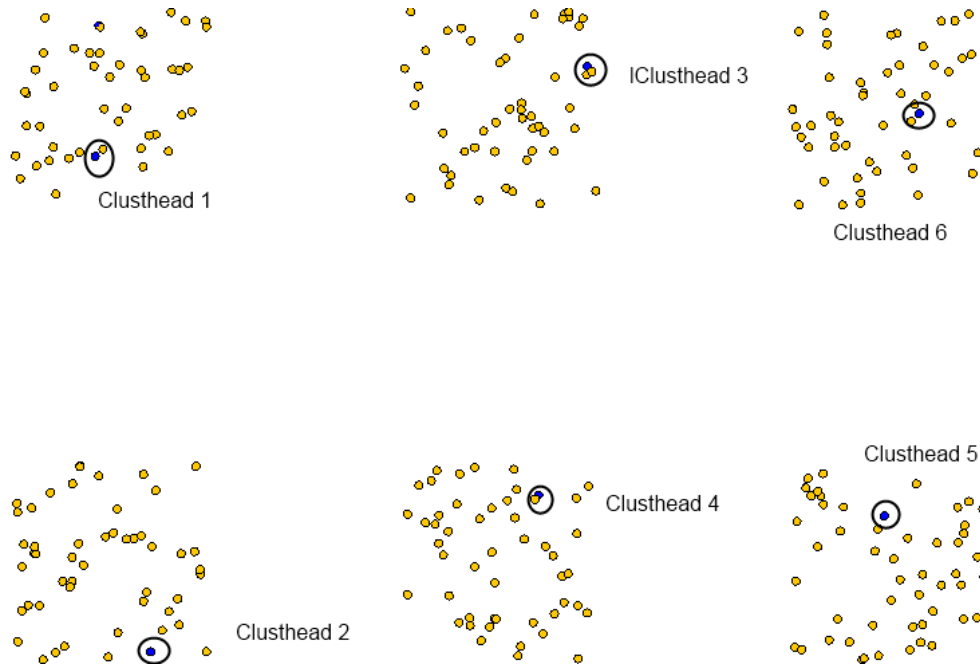
Devido ao problema retratado na figura 15, elege-se em cada ilha um clusthead, como na figura 16, responsável por receber todas as mensagens e ar-



**Figura 15:** Exemplo de nós que tem pacotes a serem enviados.

mazenar em um buffer, até o momento que o UAV identifica esse nó e recebe os pacotes a serem enviados até o destino.

O algoritmo da eleição do clusthead, cria se um temporizador como mostrado na linha 4 do código 1, que é uma quantidade de tempo dada para cada nó. No momento em que o temporizador de cada nó é inicializado, são feitas as eleições de um clusthead, verificando a posição de cada nó na ilha, selecionando como clusthead de cada ilha o nó melhor localizado, ou seja, um nó com quantidade maior de vizinhos. Quando este nó é eleito, seu temporizador é finalizado. No momento que a eleição é feita, todos os nós vizinhos são avisados da escolha do clusthead como mostrado na linha 21, e da sua respectiva posição na ilha e as demais eleições são finalizadas.



**Figura 16:** Eleição dos clustheads em ilhas desconexas.

**Listing 1:** Algoritmo, em pseudo código, que elege os clustheads.

---

```

1 Se(wuc for do tipo TestForClusterHeadWUC)
2   {
3     Se (!electionDone) {
4       Se (Math.random() < 0.01) {
5         ChangeToClusterHead();
6       }
7       WakeUpCall wk = new TestForClusterHeadWUC(sender, 300);
8       sendEventSelf(wk);
9     }
10  }
11
12 Metodo ChangeToClusterHead()
13   inicio

```

```

14     SimulationManager.logNodeState(IdDoNode, "Clusterhead", "int
        ", "90");     UavController.GetInstance().
        registerNewRoundPosition(this.node.getPosition());
15     electionDone <- verdadeiro;
16     clusterHead <- verdadeiro;
17     ClusterHeadInfo.ClusterHeads.put(IdDoNode, IdDoNode);
18     ElectionWUC election <- novo ElectionWUC(sender, 10);
19     this.sendEventSelf(election);
20     fim

```

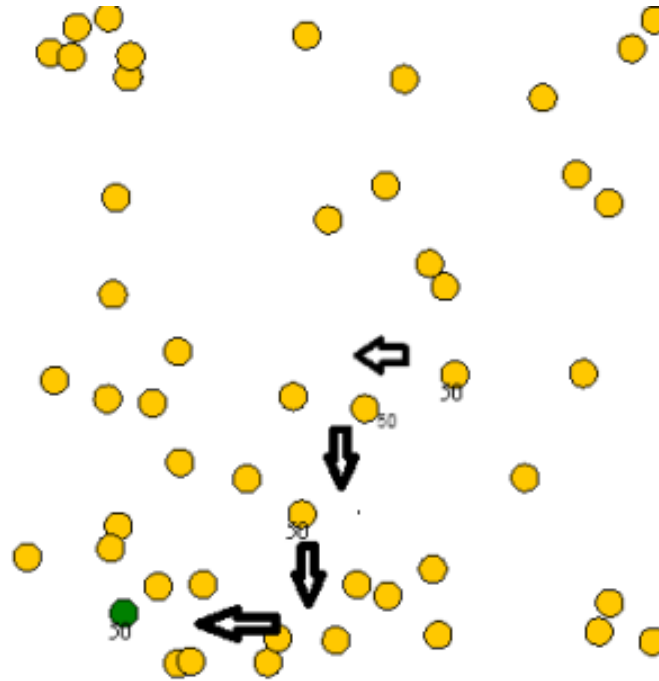
---

### 4.3 Roteamento de Pacotes

Neste trabalho, para realizar o roteamento de mensagens nas ilhas geradas foi utilizado o roteamento geográfico.

Foi realizada uma adaptação do algoritmo 2 de roteamento geográfico. Neste algoritmo, foi criado estados do pacote para determinar em qual região o pacote se encontra. Um nó, ao receber um pacote, recebe o estado *Inside*, determinando que o pacote se encontra no roteamento de nós da ilha e cria uma mensagem de requisição de posição e a envia a todos os seus vizinhos mostrado na linha 13 do algoritmo. Quando o nó recebe a mensagem de requisição de posição, cria uma mensagem de resposta com sua posição e a envia de volta ao nó que fez a requisição. Ao receber as mensagens de resposta de todos os vizinhos, o nó calcula qual deles é o nó mais próximo do clusthead, e envia a mensagem a este nó, que recebe o estado atual de *Clusthead* que por sua vez repete todo o processo até chegar ao clusthead como mostrado na figura 17.

A mensagem, no momento que chega no clusthead, altera o estado do pacote na linha 51 para *UAV* para determinar que o pacote tem como atual destino o UAV. O algoritmo, na linha 43, também verifica neste momento se o buffer de



**Figura 17:** Roteamento de pacote até o Clusthead

tamanho determinado não esta cheio. Caso esteja, o pacote é descartado. Caso contrário, na linha 52, ele armazena o pacote até a chegada do UAV. Quando o clusthead recebe uma alerta do UAV, na linha 57, o estado do pacote é alterado para *OUTSIDE*, identificando que o pacote tem como atual destino seu próprio destino. O pacote é armazenado na lista de mensagens do UAV e a posição do clusthead é armazenado na sua lista de posições da sua ronda. Quando o UAV estiver próximo ao destino do pacote, ele envia o pacote para o nó destino. O pacote chega no destino enviado e o UAV continua sua ronda. Caso contrário o pacote fica armazenado na sua lista de mensagens até este destino estiver ao alcance. Enquanto isso o UAV continua sua ronda pelas posições adicionadas na sua lista.

**Listing 2:** Algoritmo, em pseudo código, para realizar o roteamento de pacotes.

```

1 Metodo lowerSAP(Packet packet) throws LayerException
2     inicio
3         Se (packet instanceof ReplyPosPack) faça
4             inicio
5                 ReplyPosPack rp <- (ReplyPosPack) packet;
6                 neigh.add(rp.node);
7             fim
8
9         Se (packet instanceof RequestPosPack) faça
10            inicio
11                RequestPosPack rp <- (RequestPosPack) packet;
12                ReplyPosPack rpp <- novo ReplyPosPack(sender, rp.
                    getSender().getId(), this.node);
13                sendPacket(rpp);
14            fim
15
16        Se (packet instanceof GRNetPacket) faça
17            inicio
18                msgCount + 1;
19                GRNetPacket pak <- (GRNetPacket) packet;
20                Packet p <- pak.getEnclosedPacket();
21                Se (p.getReceiver().asInt() = this.getId().asInt() ) faça
22                    inicio
23                        sendPacket(p);
24                        SimulationManager.logNodeState(this.node.getId(), "Fim
                            ", "int", "100");
25                    fim
26                Senão faça
27                    inicio
28                        Se (this.node.getId().asInt() > 300) faça
29                            inicio

```

```

30
31         Se (pak.actualDestiny = <- ActualDestiny.UAV)
           inicio
32         Position pos <- (SimulationManager.getAllNodes()
           ).get(p.getReceiver()).getPosition();
33         PacketEnvelop pk <- novo PacketEnvelop(p,pos);
34         pk.actualDestiny <- ActualDestiny.OUTSIDE;
35         EnvList.add(pk);
36         SimulationManager.logNodeState(this.node.getId(),
           "Caminho do UAV", "int", "300");
37         SendNextNode(pk);
38         FillPlacesToVisit(pos);
39         UavController.GetInstance().
           registerNewOncePosition(pos);
40         fim
41     fim
42     Senão faça
43     Se (EnvList.size() > 15) faça
44     inicio
45         Statistics.getInstance().RegisterLoss();
46     fim
47     Senão faça
48     inicio
49         Position pos <- (SimulationManager.getAllNodes()).
           get(p.getReceiver()).getPosition();
50         PacketEnvelop pk <- novo PacketEnvelop(p,pos);
51         pk.actualDestiny <- pak.actualDestiny;
52         EnvList.add(pk);
53         SendNextNode(pk);
54     fim
55     fim
56     fim

```

```

57     Se (packet instanceof BeaconNetPacket) Senão
58     inicio
59         Se (packet.getSender().getId().asInt() = 301) faça
60         inicio
61             for (PacketEnvelop env: EnvList)
62             inicio
63                 env.blocked <- false;
64             fim
65         fim Senão faça
66         inicio
67             sendPacket(packet.getEnclosedPacket());
68         fim
69     fim
70 fim

```

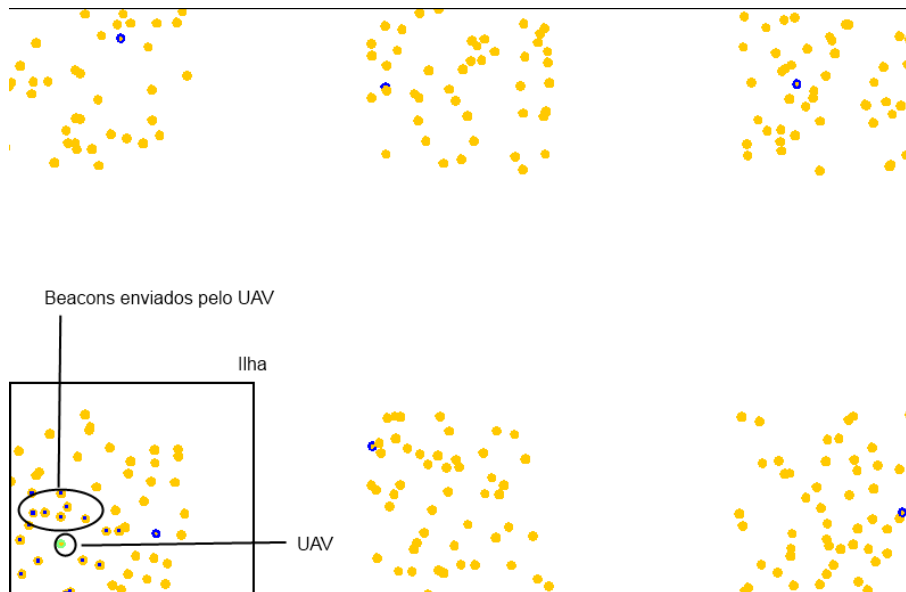
---

#### 4.4 Descoberta dos Clustheads

A identificação da posição de cada clusthead em uma ilha, representa cada ponto de desconexão na rede. Esse reconhecimento permite que o UAV restabelece a conexão nestes pontos desconexos.

Em períodos de tempo, o UAV envia mensagens de reconhecimento (beacons) como mostrado na figura 18 para todos os nós dentro do seu raio de alcance. Os nós que estão dentro deste raio são alertados da presença do UAV nas proximidades, inclusive os Clustheads. Sendo assim, se alguma mensagem estiver no buffer do clusthead, o UAV recebe esses pacotes e adiciona a posição do clusthead na sua lista de ronda. Sendo assim, o UAV voltará àquela posição periodicamente para restabelecer a conexão do nó fixo, e armazena na sua lista de mensagens os pacotes recebidos que serão entregues quando o destino estiver ao alcance do UAV.





**Figura 18:** Exemplo de UAVs enviando beacons na RSSF

Com as posições inseridas nesta lista, o UAV cria uma trajetória para visitar esses pontos periodicamente.

O seguintes métodos são utilizado pelo UAV para realizar a trajetória até um clusthead:

- **Random Walk:**

O Randon Walk, utilizado neste trabalho, funciona da seguinte maneira: O UAV está percorrendo uma trajetória aleatória, e durante o percurso ele envia mensagens de reconhecimento (beacons). Assim que um nó, no caso o clusthead, recebe essa mensagem, o clusthead verifica se tem mensagem no buffer. Existindo mensagens, o clusthead envia a mensagem para o UAV, que também adiciona essa posição do clusthead na sua lista de ronda, e fica responsável de enviar essa mensagem até seu destino. E determinados períodos de tempos ele realiza a ronda nas posições armazenadas na sua lista de ronda.

- **Visão Global:**

Neste trabalho foi utilizado o método visão global, no qual o UAV inicialmente está em uma trajetória aleatória. Os fluxos de mensagens são criados, e enquanto cada clusthead é gerado na simulação, sua posição é adicionada a lista de posição para ser utilizada na rota do UAV.

Com este método, temos vantagens no restabelecimento da conexão de uma RSSF utilizando UAV, visto que, enquanto o UAV realiza sua trajetória os pontos são adicionados na lista, porém ele não é um método realista e não foi utilizado nas simulações deste trabalho.

#### **4.4.1 Round do UAV**

O UAV pode se encontrar em dois estados. Existe o estado no qual o UAV realiza uma busca de pontos desconexos. Sendo assim sua trajetória seguirá um dos métodos de descobertas apresentados anteriormente.

Quando o UAV encontra-se em estado de round, os diferentes pontos da lista de posições do UAV serão visitados um a um, de acordo com um dos modelos apresentados anteriormente.

No capítulo a seguir serão mostrados os resultados e discussões das simulações realizadas com os algoritmos tratados neste capítulo e algumas comparações de resultados feitas com outro trabalho que tratava o restabelecimento de conexão utilizando UAVs porém em nós fixos próximos.

#### **4.5 Trajetória do UAV até o clusthead**

A definição da trajetória que o UAV seguirá na simulação é importante, visto que, essa trajetória influenciará diretamente na simulação. Existem dois mo-

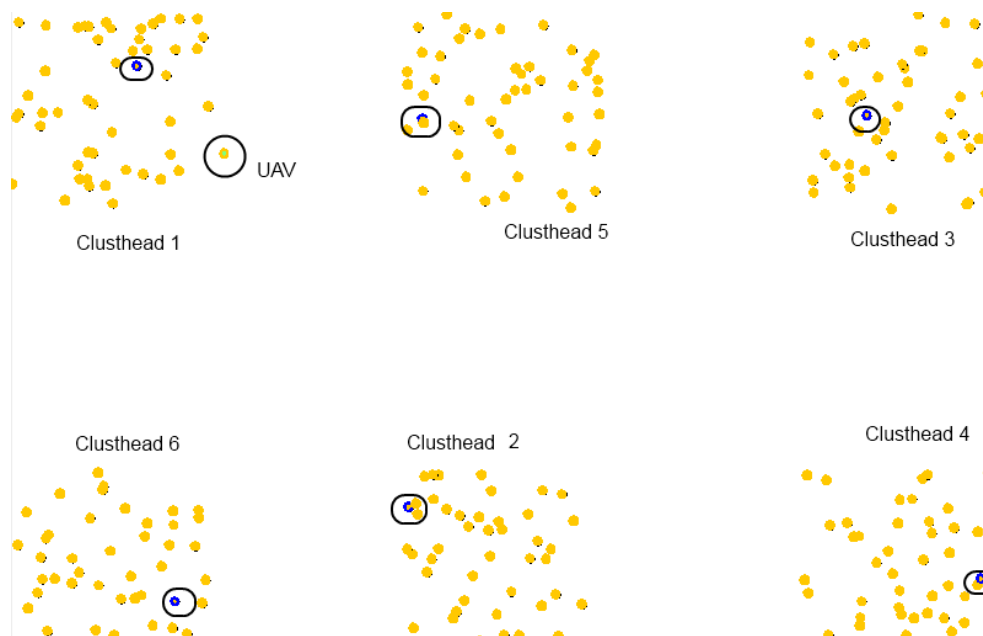
delos adaptados no trabalho para decidir qual posição será visitada primeiro pelo UAV.

#### 4.5.1 Ordem de chegada

O algoritmo da ordem de chegada faz com que o UAV visite os locais de acordo com a sequência das posições adicionadas a sua lista.

É um modelo que tem como vantagem sua simplicidade, pois basta adicionar a posição na lista, dispensando cálculos adicionais.

A Figura 19 representa um UAV visitando as posições de sua lista, utilizando o modelo ordem de chegada, onde o UAV visita o primeiro clusthead adicionado a sua lista, sendo o clusthead 1 e em sequência o clusthead 2, clusthead 3, 4, 5 e finalmente o clusthead 6.



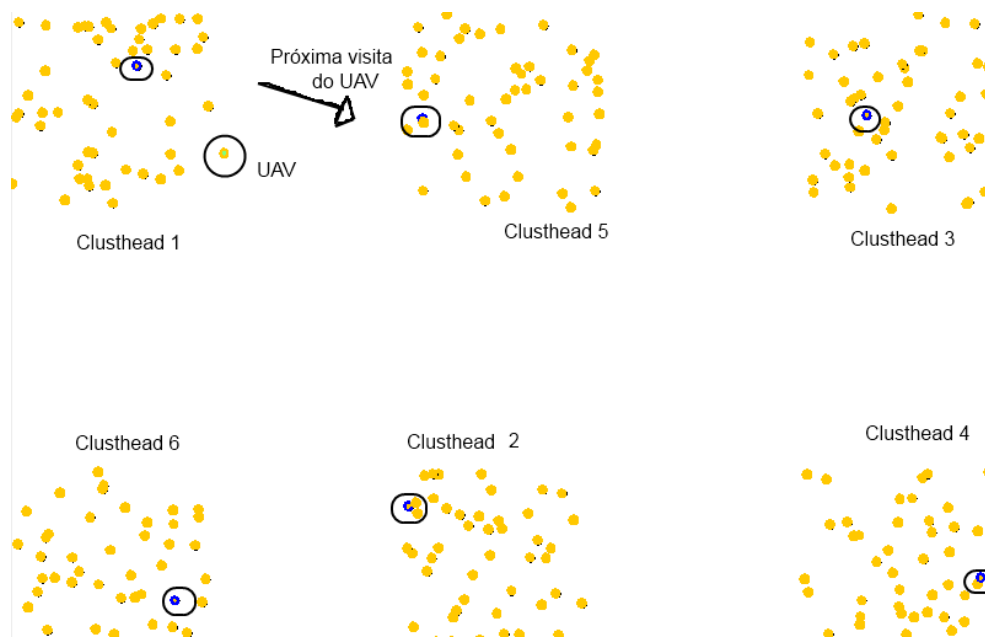
**Figura 19:** UAV percorrendo posições de uma lista usando o modelo ordem de chegada.

#### 4.5.2 Mais Próximo Primeiro

Este modelo diz que o UAV visita o local mais próximo primeiro e em seguida a posição mais próxima seguinte.

Este modelo requer cálculos para determinar as posições mais próximas. Sua vantagem é a otimização da rota do UAV.

A figura 20 mostra o UAV utilizando o modelo mais próximo primeiro, onde ele visita primeiramente o clustead 5, sendo seu nó mais próximo, apesar do clusthead ter sido adicionado na 5 posição da lista, logo em seguida ele visita o clusthead 3, 4, 2 e finalmente o clusthead 6.



**Figura 20:** UAV utilizando o modelo mais próximo primeiro.

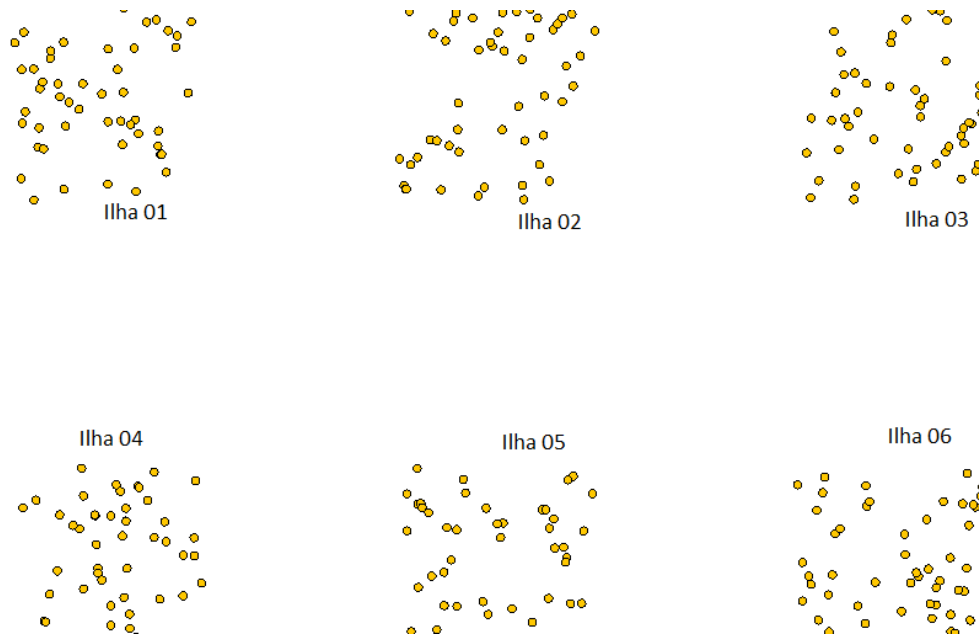
#### **4.6 Tamanho do Buffer utilizado nos clustheads**

O tamanho do buffer utilizado em cada clusthead influencia diretamente os resultados das simulações, sendo assim o tamanho do buffer é de grande importância no trabalho. Quanto maior for o buffer de cada clusthead, menor será a perda de mensagens e conseqüentemente mais satisfatórios serão os resultados. Com um buffer maior, o clusthead tem a capacidade de aguardar um UAV por mais tempo, evitando assim descarte de pacotes.

#### **4.7 Algoritmo para geração das ilhas**

Foi desenvolvido um algoritmo para gerar conjuntos de nós fixos isolados, chamados de ilhas, afim de simular pontos de desconexão distantes. Esse algoritmo gerou 6 ilhas como mostrado na figura 21, de 50 nós cada, isoladas utilizando 300 nós fixos, cada nó com um identificador. Cada ilha tem sua posição geográfica delimitada, ou seja, no momento que é gerada uma ilha, os 50 nós serão distribuídos aleatoriamente neste espaço que delimita a ilha.

O algoritmo 3 seleciona na sua primeira condição na linha 5, que dentre a quantidade total de nós, o primeiro grupo de nós serão posicionados na primeira ilha. Na linha 13, ele determina que o segundo grupo de nós serão distribuídos na segunda ilha. Na linha 22 ele determina que, o terceiro grupo de nós, serão distribuídos na terceira ilha. Na linha 31, ele determina que o quarto grupo de nós, serão distribuídos na quarta ilha. Na linha 39, ele determina que o quinto grupo de nós serem dispostos na quinta ilha. Na linha 48, ele distribui o grupo de nós restante na sexta ilha.



**Figura 21:** Geração de ilhas.

**Listing 3:** Algoritmo, em pseudo código, para geração das ilhas.

---

```

1  Função newPosition(Node node) inicio
2      numerico x;
3      numerico y;
4      Position realPos <- vazio;
5      Se (node.getId().asInt() < nodeCount / 6) faça
6          inicio
7              x <- random.nextDouble() * (0.2 * xSize);
8              x <- Math.floor(x * 100) / 100;
9              y <- random.nextDouble() * (0.3 * ySize);
10             y <- Math.floor(y * 100) / 100;
11             realPos <- new Position(x, y);
12             fim
13         Senão faça Se (node.getId().asInt() < nodeCount / 3) faça
14             inicio
15                 x <- random.nextDouble() * (0.2 * xSize) + (0.4 * xSize);

```

```
16         x <- Math.floor(x * 100) / 100;
17         y <- random.nextDouble() * (0.3 * ySize);
18         y <- Math.floor(y * 100) / 100;
19         realPos <- new Position(x, y);
20
21     fim
22     Senão faça Se (node.getId().asInt() < nodeCount / 2) faça
23     incio
24         x <- random.nextDouble() * (0.2 * xSize) + (0.8 * xSize);
25         x <- Math.floor(x * 100) / 100;
26         y <- random.nextDouble() * (0.3 * ySize);
27         y <- Math.floor(y * 100) / 100;
28         realPos <- new Position(x, y);
29
30     fim
31     Senão faça Se (node.getId().asInt() < ((nodeCount / 6) * 4))
32         faça
33         incio
34             x <- random.nextDouble() * (0.2 * xSize);
35             x <- Math.floor(x * 100) / 100;
36             y <- random.nextDouble() * (0.3 * ySize) + (0.7 * ySize);
37             y <- Math.floor(y * 100) / 100;
38             realPos <- new Position(x, y);
39         fim
40     Senão faça Se (node.getId().asInt() < (nodeCount / 6) * 5)
41         faça
42         incio
43             x <- random.nextDouble() * (0.2 * xSize) + (0.4 * xSize);
44             x <- Math.floor(x * 100) / 100;
45             y <- random.nextDouble() * (0.3 * ySize) + (0.7 * ySize);
46             y <- Math.floor(y * 100) / 100;
47             realPos <- new Position(x, y);
```

```
46
47     fim
48     Senão faça Se (node.getId().asInt() <<- nodeCount) faça
49     inicio
50         x <- random.nextDouble() * (0.2 * xSize) + (0.8 * xSize);
51         x <- Math.floor(x * 100) / 100;
52         y <- random.nextDouble() * (0.3 * ySize) + (0.7 * ySize);
53         y <- Math.floor(y * 100) / 100;
54         realPos <- new Position(x, y);
55
56     fim
```

---



## 5 Resultados e Discussões

Este capítulo apresenta os resultados e discussões acerca dos resultados obtidos nas simulações realizadas com objetivo de simular o restabelecimento de uma rede RSSF utilizando UAVs para o transporte de pacotes em regiões desconexas distantes.

### 5.1 Construção da simulação

Os testes e experimentos foram realizados para avaliar o desempenho dos algoritmos desenvolvidos no trabalho. As métricas usadas para medir este desempenho foram:

- **Porcentagem de pacotes perdidos**
- **Tempo médio do UAV para realizar Round**
- **Tamanho do "Buffer" para armazenamento das mensagens**
- **Tempo médio do pacote para chegar ao destino**

### 5.2 Parâmetros da simulação

Todos os experimentos foram realizados com objetivo de avaliar o comportamento das RSSF em caso de desconexão. Foram realizadas 20 simulações no simulador do GRUBiX. O cenário da simulação foi composto por 300 nós fixos e um nó móvel (UAV). Essa simulação avaliou o comportamento do UAV ao receber um pacote em um ponto de desconexão distante e na entrega do pacote para seu destino.

Neste cenário da simulação foi utilizado o algoritmo *Random Walk* para o monitoramento da RSSF. Foi colocado o UAV em rota aleatória. A cada vez

que ele encontra um clusthead que precisa enviar mensagens, a posição deste nó é salva na lista de round. Sendo assim, em determinados momentos, ele realiza uma rota nas posições que foram adicionadas na lista de round e verifica se este nó tem alguma mensagem a ser transportada.

Na simulação foram distribuídos 300 nós sensores terrestres (com raio de comunicação de 20 metros) dispostos em 6 ilhas isoladas afim de simular a falta de conexão das RSSF em uma área total de 300m x 200m.

A Tabela 2 mostra os valores utilizados para as simulações.

Parâmetro	Valor
Campo(Área)	300 m x 200 m
Raio de comunicação dos nós	20 metros
Número de nós terrestres	300
Tamanho do buffer de cada nó	3 a 15 mensagens
Número de nós móveis(UAVs)	1
Velocidade do nó móvel	10Km/h
Número de fluxos	2 e 4
Número de pacotes por fluxo	298
Duração de cada simulação	3000 segundos

**Tabela 2:** *Parâmetros para a simulação do Restabelecimento da Conectividade de RSSFs Utilizando Nós Móveis para transportar pacotes*

### 5.3 Resultados

Foram feitas avaliações e comparações nas simulações realizadas. Abaixo são apresentadas tabelas comparativas com estes resultados. Alguns resultados foram comparados com a outra pesquisa [51].

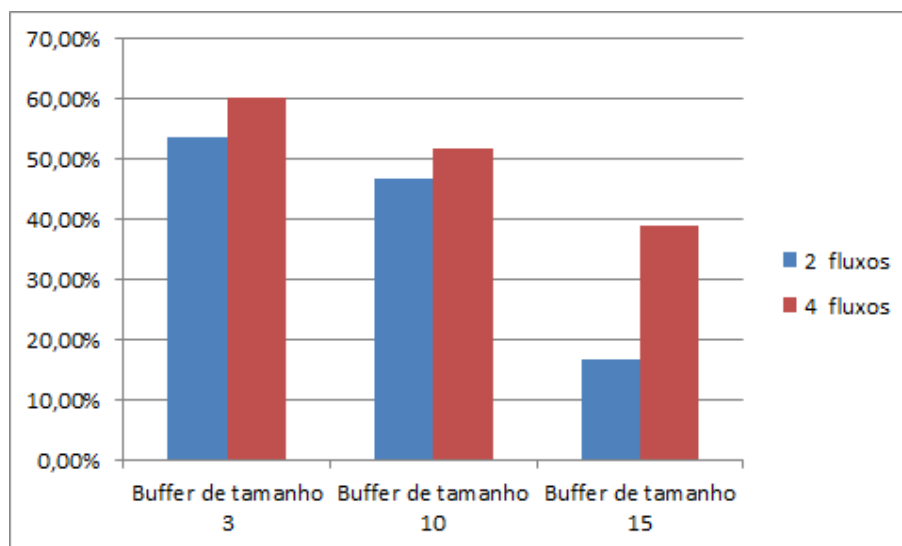
Nessas tabelas temos o buffer e a quantidade de fluxos utilizada em cada simulação. Estes resultados foram obtidos utilizando o algoritmo **FCFS** do inglês *First Come First Served* para visita das posições dos clustheads.

A Tabela 3 mostra as perdas médias de mensagens nas simulações realizadas. Esta tabela relaciona os resultados da perda de mensagens em 2 e 4 fluxos de mensagens, além de comparar a quantidade de buffer em cada fluxo.

Buffer	2 fluxos	4 fluxos	2 fluxos	4 fluxos
3 mensagens	323,75	724,5	53,8%	60,3%
10 Mensagens	281,25	561,75	46,8%	51,7%
15 Mensagens	99,75	467,78	16,6%	38,9%

**Tabela 3:** Comparação: Quantidade de pacotes perdidos.

No gráfico da Figura 22 pode-se perceber a quantidade média de pacotes perdidos. Neste gráfico podemos verificar que a medida que aumenta o tamanho do buffer dos nós, diminui-se a perda de mensagens, nas simulações com 2 e 4 fluxos de mensagens.



**Figura 22:** Gráfico: Quantidade de pacotes perdidos.

A Tabela 4 mostra os resultados do tempo de round do UAV com 2 e 4 fluxos de mensagens deste trabalho.

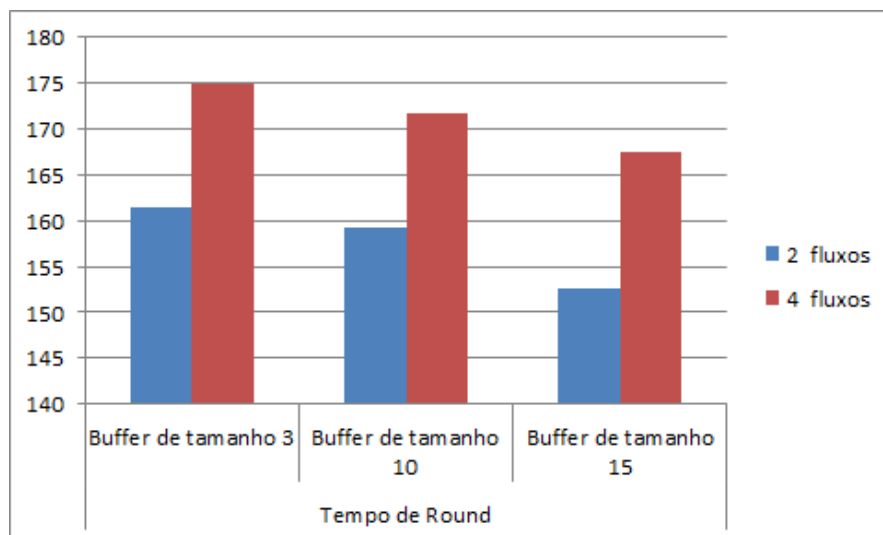
Buffer	2 fluxos	4 fluxos
3 mensagens	161,47	175,02
10 Mensagens	159,27	161,27
15 Mensagens	152,52	167,55

**Tabela 4:** Comparação: Tempo médio de Round UAV *m/s*.

O gráfico da Figura 23 mostra os resultados referentes ao Round do UAV. Pode-se concluir com estes resultados que mais uma vez o aumento do buffer contribuiu para diminuição do tempo médio do round. Isto é explicado uma vez que o UAV ao visitar o nó que tem mensagens a serem transportadas, recebe uma quantidade de mensagens referente ao seu buffer, ou seja, se o buffer tem tamanho 3, ele recebe 3 mensagens e transporta essas mensagens até seus destinos.

A Tabela 5 mostra os resultados do tempo de round do UAV em 2 e 4 fluxos de mensagens, com buffer de tamanho 3, simulados no trabalho [51]. Comparando estes resultados com o da tabela 4 deste trabalho percebemos que o round do trabalho [51] foi menor, isso é explicado porque naquele trabalho os nós desconexos estavam distribuídos em pontos próximos, de apenas um salto, já neste trabalho o UAV realiza um round maior, porque precisa percorrer várias regiões desconexas com salto de nós distantes.

A Tabela 6 mostra os resultados do tempo médio de entrega de mensagens no destino deste trabalho.



**Figura 23:** Gráfico: Tempo médio de Round do UAV.

Algoritmo	Dois Fluxos	Quatro Fluxos
FCFS	90,16	143,09

**Tabela 5:** Comparação: Tempo médio de “round” do UAV do trabalho de restabelecimento de comunicação usando UAV [51].

Buffer	2 fluxos	4 fluxos
3 mensagens	90,41	106,7
10 Mensagens	73,62	86,96
15 Mensagens	69,5	89,26

**Tabela 6:** Comparação: Tempo médio m/s de entrega de mensagens.

A Tabela 7 mostra os resultados do tempo médio de entrega de mensagens no destino no trabalho [51].

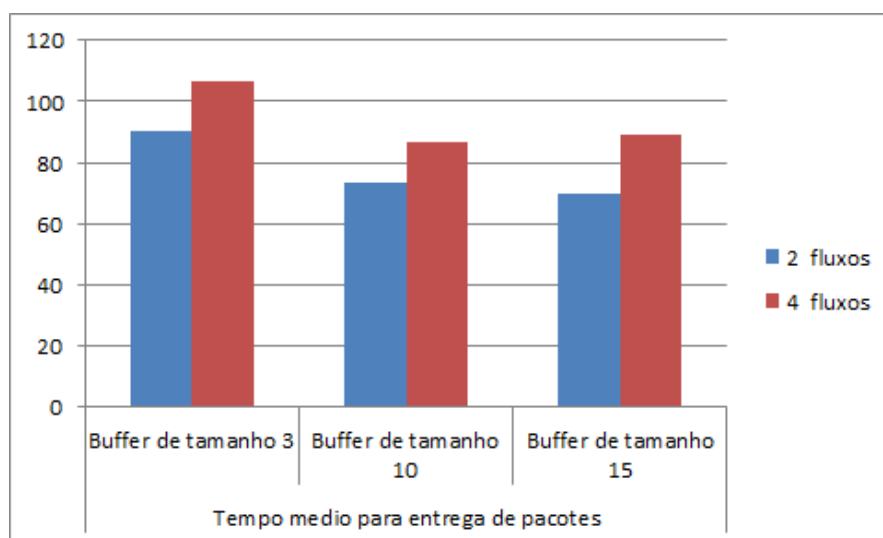
Algoritmo	Dois Fluxos	Quatro Fluxos
FCFS	11,32	73,78

**Tabela 7:** Comparação: Tempo médio gasto pelo pacote para chegar ao destino do trabalho de restabelecimento de comunicação usando UAV [51].

Comparando os resultados das tabelas 7 [51], onde mostra o tempo médio de entrega de mensagens em nós de um salto e a tabela 6 que apresenta os resultados de entrega de mensagens em regiões de saltos distantes e desconexas. Mais uma vez, é percebido que este trabalho gasta um tempo maior de entrega de mensagens, porque seus nós estão isolados com saltos maiores, sendo assim, existe uma distância maior de um nó desconexo a outro nó conexo.

O gráfico da Figura 24 mostra os resultados referentes ao tempo médio de entrega de mensagens até o destino. Pode-se concluir que a variação do tamanho do buffer nos 2 e 4 fluxos de mensagens ajuda a diminuir o tempo médio de entrega de pacotes.

Pode-se concluir através das análises das tabelas e gráficos que a utilização dos UAVs para transportar mensagens obteve sucesso, visto que além de conseguir transportar as mensagens em uma RSSF comprometida, consegue-se diminuir a taxa de perda de pacotes com o aumento do buffer dos nós. Com um aumento maior ainda do tamanho do buffer pode obter uma vantagem melhor nos resultados dos pacotes perdidos e conseqüentemente o tempo de entrega dos pacotes. Embora os resultados apresentados neste trabalho, sejam maiores que no trabalho [51], isto já era esperado, uma vez que a distância de salto dos nós desconexos são maiores. Em testes preliminares feitos nas simulações, a utilização de um buffer de tamanho



**Figura 24:** Gráfico: Tempo médio de entrega de pacotes no destino.

maior do que os simulados neste trabalho trouxe melhores resultados. No próximo capítulo serão apresentadas as conclusões deste trabalho.

## 6 Conclusão

O trabalho desenvolveu técnicas para o uso de UAVs para o transporte de pacotes em redes de sensores sem fio, onde existiam ilhas desconexas. Foram realizados testes e experimentos computacionais para comprovar tanto a eficácia como eficiência dos métodos desenvolvidos.

Os resultados dos testes e simulações comprovaram que a utilização dos UAVs, para transporte de mensagens, ajuda no restabelecimento da conectividade em RSSFs e que, como visto anteriormente nas análises feitas nos gráficos, o aumento do buffer dos nós diminui na perda de pacotes, fazendo com que o UAV consiga transportar uma quantidade maior de mensagens. Pode-se comprovar que um aumento do buffer para 15 pacotes em 2 fluxos de mensagens gerou uma perda aproximada de 17% do valor total de pacotes, e em 4 fluxos de mensagens, uma perda aproximada de 40% da quantidade total de pacotes.

Os resultados obtidos foram comparados com [51], conclui que a distância entre os nós desconexos altera diretamente os resultados do round do UAV.

Este trabalho reafirmou que o uso de UAVs pode aumentar a conectividade das redes de sensores sem fio, afirmando assim as vantagens no uso do UAV no trabalho.

Nos trabalhos futuros poderão ser combinados mais de um UAV monitorando a RSSF. A comunicação entre dois UAVs em uma RSSF ajudaria a otimizar as rotas realizadas pelos mesmos.



## Referências

- [1] ABOELAZE, M., AND ALOUL, F. Current and future trends in sensor networks: A survey, 2005.
- [2] QUARITSCH, M. AND KRUGGL, K. AND WISCHOUNIG-STRUCL, D. AND BHATTACHARYA, S. AND SHAH, M. AND RINNER, B. Networked UAVs as aerial sensor network for disaster management applications: A survey, 2005.
- [3] ANDERS LINDGREN. The quest for a killer app for opportunistic and delay tolerant networks (invited paper), 2009.
- [4] BASU, P., AND REDI, J. Coordinated flocking of uavs for improved connectivity of mobile ground nodes. *Military Communications Conference. MILCOM 2004. IEEE Vol. 3* (2004), 1628–1634.
- [5] BEARD, R. W., AND MCLAIN, T. W. Coordinated target assignment and intercept for unmanned air vehicles. *Robotics and Automation, IEEE Transactions on* 18, 6 (dec 2002), 911–922.
- [6] BONE, E., AND BOLKCOM, C. Unmanned aerial vehicles:background and issues for congress. Tech. rep., Congressional Research Service - The Library of Congress, 2003.
- [7] CHEE-YEE, AND KUMAR, S. P. Sensor networks: Evolution, opportunities, and challenges. *Proceeding of the IEEE Vol.91 No.08* (Jan. 2003).
- [8] CORREA, U., CODAS, A., AND MONTEZ, C. Redes locais sem fio: Conceito e aplicacoes.
- [9] CORREA, U., CODAS, A., AND MONTEZ, C. Redes locais sem fio: Conceito e aplicacoes.

- [10] DRESSLER, F. Self-Organization in Sensor and Actor Networks.
- [11] CORSON, S., AND MACKER, J. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations, 1999.
- [12] CULLER, D., ESTRIN, D., AND SRIVASTAVA, M. Introduction: Overview of sensor networks. *IEEE Computer Vol. 37, No. 8* (2004).
- [13] DE A. BARRETO, G., ARAUJO, A. F. R., AND DE O. ROSA, M. Algoritmo de busca heuristica usando redes neurais competitivas para planejamento otimo de trajetoria de um robo movel.
- [14] DE MELLO, B. A. Modelagem e simulacao de sistemas.
- [15] DESENVOLVEDORES. *Manual do Simulador GrubiX*, 2011.
- [16] DREW, J. G., SHAVER, R., LYNCH, K. F., AMOUZEGAR, M. A., AND SNYDER, D. Unmanned aerial vehicle end-to-end support considerations. Tech. rep., RAND CORP SANTA MONICA CA, 2005.
- [17] ERMAN, A., HOESEL, L., AND HAVINGA. Enabling mobility in heterogeneous wireless sensor networks cooperating with uavs for mission-critical management. *Wireless Communications, IEEE 15, 6* (december 2008), 38–46.
- [18] FODOR, K., AND VIDACS, A. Efficient routing to mobile sinks in wireless sensor networks. In *Proceedings of the 3rd international conference on Wireless internet (ICST, Brussels, Belgium, Belgium, 2007)*, WICON '07, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 32:1–32:7.

- [19] GARCIA-HERNANDEZ, C. F., GONZALEZ, P. H. I., GARCIA-HERNANDEZ, J., AND PEREZ-DIAZ, J. A. Wireless sensor networks and applications: a survey. *IJCSNS International Journal of Computer Science and Network Security Vol.7 No.3* (2007), 264–273.
- [20] HAMMOURI, O. M., AND MATALGAH, M. M. Voronoi path planning technique for recovering communication in uavs. In *Proceedings of the 2008 IEEE/ACS International Conference on Computer Systems and Applications* (Washington, DC, USA, 2008), AICCSA '08, IEEE Computer Society, pp. 403–406.
- [21] HEIMFARTH, T., AND FREITAS, E. P. Experimental analysis of a wireless sensor network setup strategy provided by an agent-oriented middleware.
- [22] BERNDT, A . Introdução a Redes de Sensores sem Fio (RSSF).
- [23] HEINEN, F. J. Sistemas de controle hibrido para robos moveis autonomos. Master's thesis, Universidade do Vale do Rio dos Sinos - UNISINOS Centro de Ciencias Exatas e Tecnologicas Mestrado em Computação Aplicada - PIPCA, 2002.
- [24] HENRIQUES, R., BACAO, F., AND LOBO, V. Uav path planning based on event density detection. *2009 International Conference on Advanced Geographic Information Systems Web Services d, 1* (2009), 112–116.
- [25] SAMIR, DA S. em <http://www.miqueiasreale.com.br/?p=1014>.
- [26] ROBOTSHOP. AR.Drone iPhone Controlled Quadrocopter Unveiled at CES2010. em <http://www.robotshop.com/gorobotics/the-news/ardrone-iphone-controlled-quadrocopter-unveiled-at-ces2009>.

- [27] ROBERT, V. Como funciona o Predator UAV. em <http://ciencia.hsw.uol.com.br/predator.htm/>.
- [28] JOHN, P. RQ-4A Global Hawk Tier II+ HAE UAV. em <https://www.fas.org/irp/program/collect/globalhawk.htm/>.
- [29] PEDRO, PINTO. Rede de Sensores sem fios, Conheça esta tecnologia. em <http://pplware.sapo.pt/windows/software/redes/rede-de-sensores-sem-fios-conhece-esta-tecnologia/>.
- [30] WELLINGTON, W. F. SARMENTO. Os desafios da integração de Identificação por Rádio Frequência e Redes de Sensores Sem-fio no contexto da Computação Ubíqua. em <http://thinkingandthebrain.blogspot.com.br/2010/04/os-desafios-da-integracao-de.html>.
- [31] LANGER, R. A. Estudo e implementacao de metodos para planejamento de trajetorias e controle de robos moveis nao holonomicos. Master's thesis, Pontificia Universidade Catolica do Parana - PUCPR, 2007.
- [32] LESSMANN, J., HEIMFARTH, T., AND JANACIK, P. Shox: An easy to use simulation platform for wireless networks. *UKSIM '08: Proceedings of the Tenth International Conference on Computer Modeling and Simulation* (2008), p. 410–415.
- [33] LOUREIRO, A., NOGUEIRA, J. M., AND RUIZ., L. B. Redes de sensores sem fio. *Tutoriais do Simposio Brasileiro de Redes de Computadores*. (2003).
- [34] MEDEIROS, F. L. L., AND DA SILVA, J. D. S. Grafos de visibilidade aplicados a representacao computacional de ambientes de navegacao aerea.

- [35] MHATRE, V. P., ROSENBERG, C., KOFMAN, D., MAZUMDAR, R., AND SHROFF, N. A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Transactions on Mobile Computing* 4 (January 2005), 4–15.
- [36] NETO, A. A. Geracao de trajetorias para veiculos aereos autonomos nao-tripulados, 2008.
- [37] NETO, R. F. T., AND DOS SANTOS COELHO, L. Planejamento de rotas para robos de inspecao usando uma nova abordagem swarm intelligence.
- [38] PERKINS, C. E. *Ad hoc networking*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
- [39] PINHEIRO, P. G., DE MELO, A. A. N., DE V. FILHO, J. E., AND FURTADO, V. A representacao do tempo em simulacoes computacionais.
- [40] RAO, A., RATNASAMY, S., PAPADIMITRIOU, C., SHENKER, S., AND STOICA, I. Geographic routing without location information.
- [41] RICHARDS, A., BELLINGHAM, J., TILLERSON, M., AND HOW, J. Co-ordination and control of multiple uavs.
- [42] POERSCH, A. Localização de Recursos em Redes de Emergência Tolerantes a Atrasos e Desconexões.
- [43] OLIVEIRA, C. Uma proposta de roteamento probabilístico para redes tolerantes a atrasos e desconexões.
- [44] CLAUDIO, SA DE ABREU. Modelo matematico para DTN epidêmica.
- [45] GUSTAVO, LUIZ DUARTE. Redes Tolerantes a Atrasos e Desconexões.

- [46] ALEXANDRE KAMINAGAKURA, DIOGO DE ANDRADE E LUIZ CARLOS PESSOA ALBINI. Analisando o uso de intermediarios para entrega de mensagens em redes DTN.
- [47] PINHEIRO, P. G. A representacao do tempo em simulacoes computacionais..
- [48] ETIENNE C. R. DE OLIVEIRA E CÉLIO V. N. DE ALBUQUERQUE. Avaliação de Protocolos de Roteamento para Redes Ad hoc e RSSF Aplicados à TV Digital Interativa e Cidades Digitais.
- [49] OTTO CARLOS, M., LUIS HENRIQUE, M. Roteamento Geográfico.
- [50] OSHU, LEI AND ZHANG, YAN AND YANG. TPGF: geographic routing in wireless multimedia sensor networks.
- [51] MOIZES, P. Restaurando a conectividade de RSSFs através de nós móveis.
- [52] BRAD, K., AND H. T. KUNG. GPSR:Greedy perimeter stateless routing for wireless networks.
- [53] BRENDAN, B., OLIVER BROCK BRIAN , NEIL LEVINE. MV Routing and Capacity Building in Disruption Tolerant Networks.
- [54] SOD. Uas roadmap 2005. Tech. rep., Office of The Secretary of Defense, 2005.
- [55] STEERE, D. C., BAPTISTA, A., MCNAMEE, D., PU, C., AND WALPOLE, J. Research challenges in environmental observation and forecasting systems. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00)* (2000), pp. 292–299.

- [56] TAHERKORDI, A., AND TALEGHAN, M. Dependability considerations in wireless sensor networks applications. *Journal of Networks Vol. 1, 6* (2006).
- [57] WOLF, D. F., DO VALLE SIMOES, E., OSORIO, F. S., AND JUNIOR, O. T. *Robotica movel inteligente: Da simulacao as aplicacoes no mundo real*. 2008.
- [58] ZAMBALDE, A. L., DA SILVA E PADUA, C. I. P., AND ALVES, R. M. *O documento cientifico em ciencia da computacao e sistemas de informacao*.