



**LUIZ AUGUSTO GUIMARÃES COSTA**

**MIGRAÇÃO DE AGENTE SENSITIVO A POSIÇÃO  
GEOGRÁFICA EM REDES VEICULARES**

**LAVRAS - MG**

**2012**

**LUIZ AUGUSTO GUIMARÃES COSTA**

**MIGRAÇÃO DE AGENTE SENSITIVO A POSIÇÃO GEOGRÁFICA EM  
REDES VEICULARES**

Monografia apresentada ao Colegiado do  
Curso de Sistemas de Informação, para a  
obtenção do título de Bacharel em Siste-  
mas de Informação.

Orientador

Tales Heimfarth

**LAVRAS - MG**

**2012**

**LUIZ AUGUSTO GUIMARÃES COSTA**

**MIGRAÇÃO DE AGENTE SENSITIVO A POSIÇÃO  
GEOGRÁFICA EM REDES VEICULARES**

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

APROVADA em 11 de outubro de 2012.

RAPHAEL WINCKLER DE BETTIO 

WILIAN SOARES LACERDA 



TALES HEIMFARTH

(orientador/a)

**LAVRAS-MG**

**2012**

## RESUMO

O uso de agentes de software para celulares é uma abordagem promissora para implementar serviços e disseminar dados sobre redes *ad hoc*. Este trabalho apresenta uma análise dos agentes autônomos móveis com diferentes níveis de inteligência que lhes permitam fazer uso das informações geográficas, com diferentes níveis complexidade, em uma rede veicular *ad hoc*. Esta informação considera os nós atuais e futuras localizações, bem como a sua rota para chegar aos seus destinos, dependendo do nível de inteligência implementadas pelos agentes. Usando essa informação, os agentes decidem seu movimento de nó em nó durante conexões oportunistas, a fim de realizar seus objetivos relacionados à divulgação de dados e / ou ser-vice-disposição. A análise da proposta é feita no contexto de uma aplicação de rede de sensores, utilizando agentes móveis, que funcionam em cima de uma infra-estrutura  *Vehicular Ad Hoc Network*(VANET). Os resultados da simulação são apresentados e discutidos para apoiar as idéias propostas.

Palavras-chave: sensibilidade; Conscientização; Comunicação Oportunista; Agentes de Software; Redes Ad Hoc, VANETs.

## ABSTRACT

The use of mobile software agents is a promising approach to implement services and disseminate data over ad hoc networks. This paper presents an analysis of mobile autonomous agents with different levels of intelligence that allow them to make usage of the positioning information with different depth of complexity in a mobile ad hoc network aiming at efficient data dissemination. This information considers the nodes current and future locations, as well as their route to reach their destinations, depending on the level of intelligence implemented by the agents. Using this information, the agents decide their movement from node to node during opportunistic connections in order to accomplish their goals related to data dissemination and/or service provision. The analysis of this proposal is done in the context of a sensor network application, implemented by sensing services provided by the mobile agents, which run on top of an infrastructure-less Vehicular Ad hoc Network (VANET). Simulation results are presented and discussed to support the proposed ideas.

Keywords: Awareness; Opportunistic Communication; Software Agents; Ad Hoc Networks, VANETs.

## LISTA DE FIGURAS

Figura 1	Tipos de VANET (KUMAR; DAVE, 2011). . . . .	14
Figura 2	O Nó S envia um pacote para o nó D e a linha vermelha mostra o caminho da mensagem contornando a janela. . . . .	17
Figura 3	Google Maps API exemplo. . . . .	18
Figura 4	Exemplo de Modelo de Movimento Analítico. . . . .	20
Figura 5	Exemplo de Modelo de Movimento Simulado. . . . .	20
Figura 6	Exemplo de cenário e comportamento dos nós. . . . .	21
Figura 7	Migração do agente. . . . .	22
Figura 8	Exemplo de uma migração bem-sucedida de um Agente. . . . .	27
Figura 9	Exemplos para a avaliação realizada pelo algoritmo do nível de inteligência de caminho direto. . . . .	32
Figura 10	Exemplo de caminhos rota completa considerados pelo agente utilizando a inteligência Avançada. . . . .	34
Figura 11	Modelo do ambiente simulado representando um mapa quadrado dividido em quarteirões regulares. . . . .	38
Figura 12	Vista da cidade de Lavras utilizado no segundo modelo de movimento. . . . .	40
Figura 13	Média de migrações por agentes no modelo de movimento quadras. . . . .	43
Figura 14	Média de migrações por agentes no modelo de movimento Google Maps. . . . .	44

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>7</b>
<b>1.1</b>	<b>Objetivo .....</b>	<b>9</b>
<b>1.1.1</b>	<b>Objetivo Geral.....</b>	<b>9</b>
<b>1.1.2</b>	<b>Objetivos Específicos .....</b>	<b>10</b>
<b>1.2</b>	<b>Organização do trabalho.....</b>	<b>11</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO .....</b>	<b>12</b>
<b>2.1</b>	<b>Redes Ad Hoc.....</b>	<b>12</b>
<b>2.2</b>	<b>Redes Tolerantes a Interrupção (DTN).....</b>	<b>12</b>
<b>2.3</b>	<b>Redes Moveis Ad Hoc (MANET).....</b>	<b>13</b>
<b>2.4</b>	<b>Redes Veiculares Ad Hoc (VANET).....</b>	<b>13</b>
<b>2.5</b>	<b>Sistemas Multi-Agentes .....</b>	<b>15</b>
<b>2.6</b>	<b>Roteamento geográfico .....</b>	<b>16</b>
<b>2.7</b>	<b>Google Maps API .....</b>	<b>17</b>
<b>2.8</b>	<b>Modelo de Movimentos.....</b>	<b>19</b>
<b>3</b>	<b>CONTEXTUALIZAÇÃO DO PROBLEMA.....</b>	<b>21</b>
<b>4</b>	<b>METODOLOGIA .....</b>	<b>24</b>
<b>4.1</b>	<b>Classificação do Trabalho .....</b>	<b>24</b>
<b>4.2</b>	<b>Procedimentos metodológicos .....</b>	<b>24</b>
<b>4.3</b>	<b>GRUBiX e Simulação Orientada a Eventos.....</b>	<b>25</b>
<b>4.4</b>	<b>Solução Proposta.....</b>	<b>26</b>
<b>4.5</b>	<b>Nível de Inteligência do Agente.....</b>	<b>28</b>
<b>4.5.1</b>	<b>Nível de Inteligência Baseada no Destino .....</b>	<b>30</b>
<b>4.5.2</b>	<b>Nível de Inteligência Utilizando Caminho Direto .....</b>	<b>31</b>
<b>4.5.3</b>	<b>Nível de Inteligência Avançada .....</b>	<b>33</b>
<b>5</b>	<b>RESULTADO E DISCUSSÃO.....</b>	<b>37</b>
<b>5.1</b>	<b>Ambiente Simulado.....</b>	<b>37</b>
<b>5.1.1</b>	<b>Configuração da Simulação.....</b>	<b>39</b>
<b>5.2</b>	<b>Resultados das simulações utilizando o Modelo de Movimento em Quadras.....</b>	<b>42</b>
<b>5.3</b>	<b>Resultados das simulações utilizando o modelo de movimento baseado em dados geográficos do google maps. ....</b>	<b>43</b>
<b>6</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS .....</b>	<b>47</b>
	<b>REFERÊNCIAS.....</b>	<b>49</b>

## 1 INTRODUÇÃO

Comunicação sem fio permite novas aplicações desafiadoras que exploram a transferência de dados entre os nós móveis oportunistas. Redes oportunistas são redes onde a comunicação não é contínua, isto é, nunca deve existir um caminho pré-definido entre o nó de origem e o nó destino (PELUSI et al., 2006). Isto proporciona suporte para os cenários em que os dados são transportados a partir de uma fonte para um destino via nós móveis que levam fisicamente os dados a partir de um lugar para outro, sem a necessidade de uma infra-estrutura de rede fixa. Essas características podem ocorrer em uma série de aplicações tolerantes ao atraso (FALL, 2003). Entretanto, o aumento do uso de sistemas que exploram tal paradigma exige um maior número de nós que ajudam com o transporte de dados, com o objetivo de aumentar a probabilidade de que os dados eventualmente serão entregues. Neste ponto, a utilização dos nós mais promissores para transferências de dados, sendo aqueles que proporcionam uma maior probabilidade de conduzir os dados para o seu destino e com custos mais baixos apresentam uma importância notável.

A comunicação entre os nós móveis oferece adversidades diferentes, dependendo do tipo do nó que compõe a rede. Pequenas baterias dos dispositivos elétricos, tais como telefones celulares ou PDAs, têm restrições de consumo de energia importantes, daí o uso eficiente deste recurso é uma obrigação (WOLF, 2008). Por outro lado, a outros tipos de redes, tais como *redes veiculares ad-hoc* (VANETs) têm limitações em relação as exigências da qualidade do serviço, devido às altas velocidades com as quais os nós deslocam em tais redes, e, portanto, o intervalo de tempo que os nós tem para se comunicar é curto (ANDM; LARSSON, 2010). Esta conectividade intermitente, com janelas de tempo curto para se comunicar em VANETs resulta em ligações temporárias que não oferecem grande

confiabilidade, exigindo estratégias oportunistas para executar a comunicação entre os nós (CONTI et al., 2009).

Um agente de software móvel (LANGE; OSHIMA, 1999) é uma entidade de software capaz de mover ou migrar (CHESS et al., 1995), a partir de um nó hospedeiro na rede para outro. Ele inicia a sua execução, num nó, e pode continuar ou terminar em qualquer outro nó. Como parte essencial, um agente tem o código que define o seu comportamento e também dados que podem ser armazenados durante o movimento de um nó para o outro (CHESS et al., 1995). Uma característica importante é a autonomia que um agente pode ter em relação ao seu movimento em torno da rede (mobilidade) (HUHNS; SINGH, 1998). O uso de agentes de software móveis fornece uma abordagem modular para implementar serviços e/ou de transporte de dados encapsulados com comportamento inteligente o que permite a gestão da comunicação relacionada aos movimentos dos agentes entre os nós da rede.

Esse recurso oferece flexibilidade para implementar estratégias durante a execução do agente para superar as preocupações específicas apresentadas por determinados cenários em que a rede se destina ao trabalho, evitando estratégias pré-planejadas e inflexíveis que não são capazes de lidar com contextos de tempo de execução diferentes. Além disso, a utilização de agentes de software torna possível a injeção de novas aplicações na rede, de acordo com as necessidades dos utilizadores. Esse recurso oferece flexibilidade permitindo a instalação de um software comum nos nós (um middleware) (LEVIS; CULLER, 2002), e uma flexibilidade maior implantação de novas aplicações sob demanda.

As VANETs são redes em que os nós são móveis, dados trocados por eles podem ser entregues por uma mensagem ou mecanismo para transporte de dados que explora o paradigma *store-carry-and-forward* (ZHAO et al., 2004). Este

paradigma define que um nó recebe dados, armazena na memória e, em seguida, muda a sua posição, ou seja, ele vai encaminhar os dados anteriormente adquiridos para outro nó. Se os agentes estão a ser transmitidos entre os nós, é possível afirmar que este é um tipo de "agente de transporte", que é um conceito explorado (TEI et al., 2005). O transporte de agente usa o mesmo princípio de transporte de dados, mas em vez de apenas dados, um agente também transfere o seu comportamento. Uma característica interessante de usar agentes para transportar dados é que eles não são limitados a um protocolo de roteamento instalados nos nós, mas como parte de sua própria inteligência, eles podem fornecer a sua própria estratégia para mover através dos nós na rede.

Este trabalho explora uma abordagem em que os agentes de software móveis são utilizados para implementar serviços que precisam recolher e divulgar dados em uma rede ad hoc Veicular (VANET). Objetivo dos agentes de software é eventualmente atingir os nós que estão dentro da região alvo, em inglês *Target Region* (TR) através da migração do agente entre nós que se deslocam em uma área.

Serão desenvolvido três agentes de software com níveis de inteligência diferentes e comparar esses níveis de inteligência. A proposta é iniciada a partir de um mecanismo muito simples e evolui para os mais sofisticados, em que: 1) a informação de posicionamento muito básico, 2) investiga a utilização de informação de direção, e 3) um mecanismo que considera a rota completa do nó móvel.

## **1.1 Objetivo**

### **1.1.1 Objetivo Geral**

O objetivo geral deste trabalho é o desenvolvimento de algoritmos de transporte de agentes móveis em uma rede veicular movendo-se em uma cidade

até acessar um nó que esteja dentro da região alvo. Após encontrar a região alvo o agente deve permanecer a maior parte do tempo possível dentro dessa região.

### 1.1.2 Objetivos Específicos

Este trabalho tem os seguintes objetivos específicos:

1. Desenvolver algoritmos para transporte de agentes móveis utilizando:
  - Destino do nó móvel como referência ( nós sem *Global Positioning System* ou GPS).
  - Probabilidade de um determinado nó passar perto da região alvo (com GPS).
  - Rota Completa (com GPS).
2. Implementação dos algoritmos e técnicas no ambiente de simulação GRU-BiX. Um outro objetivo desse projeto é a implementação de cada um dos algoritmos no ambiente de simulação *open source* GRUBiX, desenvolvido pelo Grupo de Redes Ubíquas do Departamento de Ciência da Computação da Universidade Federal de Lavras, baseado no simulador Shox (LESSMANN et al., 2008).
3. Criação de um modelo de movimento baseado em quadras para a simulação dos algoritmos acima considerados. Nesse modelo os dispositivos irão percorrer quadras, quando os dispositivos chegarem no final da quadra ele deverá realizar uma escolha entre alterar seu movimento para esquerda, direita ou continuar em frente.
4. Criação de um modelo de movimento baseado em posições reais extraídas do aplicativo "Google Maps"(GOOGLE, 2012). Os movimentos serão base-

ados em mapas reais, então nesse modelo será considerado ruas sem saída, contramão e tudo que poderá ser considerado em uma pequena ou grande cidade.

## **1.2 Organização do trabalho**

O capítulo 1 apresenta um introdução sobre o assunto. No capítulo 2 é apresentação do referencial teórico. No capítulo 3 é apresentado a descrição geral do problema abordado. A metodologia desenvolvida é apresentada na seção 4. No capítulo 5 apresenta a descrição dos experimentos e os resultados obtidos. As conclusões do trabalho são apresentados no capítulo 6.

## 2 REFERENCIAL TEÓRICO

O presente capítulo apresenta os conhecimentos necessários para a compreensão do presente trabalho.

### 2.1 Redes Ad Hoc

Uma rede ad hoc é definida como uma coleção de entidades móveis interligados por uma tecnologia sem fio, formando uma rede temporária sem a ajuda de qualquer administração ou qualquer tipo de apoio fixo (ZAFOUNE et al., 2007).

A rede ad hoc é um tipo descentralizado de rede, visto que não depende de uma infra-estrutura pré-existente, como roteadores em redes com fio ou pontos de acesso em redes sem fio (NIAZI; HUSSAIN, 2009). Outra característica é a capacidade das redes ad hoc adequarem a condição da conectividade da rede, distribuindo a responsabilidade de organização e controle da rede para os nós, podendo determinar se um nó vai transmitir ou não dinamicamente (REZENDE et al., 2008).

As redes ad hoc possuem características próprias que a tornam adequadas a situações onde não existe uma infra-estrutura de comunicação presente (NICULESCU; NATH, 2003). Elas também podem ser utilizadas em situações onde não seja possível utilizar nós críticos para organização ou controle da rede, pois o desempenho da rede ad hoc não é afetado se um nó falhar ou mesmo sair da rede (DRESSLER, 2008).

### 2.2 Redes Tolerantes a Interrupção (DTN)

Segundo (NICHOLS; HAMMONS, 2007), *Delay-tolerant networking* (DTN) são redes projetadas para trabalhar em ambientes com comunicação instável ou

temporária produzido por limitações ou anomalias, causando o menor impacto negativo possível na aplicação.

O autor (BURGESS et al., 2006) sugere que rotas instáveis podem ser causadas por problemas como: a alta mobilidade dos nós, a baixa densidade de nós, uma curta faixa para transmissão de dados ou interferências ambientais.

A DTN é uma arquitetura de rede muito utilizada atualmente na exploração do espaço ou de planetas distantes (SPYROPOULOS et al., 2010), por que o espaço é um ambiente hostil que proporciona um longo período de latência no sinal, como também existe uma grande interferência eletromagnética.

### **2.3 Redes Móveis Ad Hoc (MANET)**

A *Mobile Ad Hoc Networks* (MANET) é um categoria de redes ad hoc (REZENDE et al., 2008) onde os nós são livres para deslocarem enquanto se comunicam com outros nós. Por causa da natureza dinâmica das MANETs os dados que trafegam pelas MANETs podem ser corrompidos ou perdidos com facilidade (BAI et al., 2003).

Essa configuração proporciona uma frequente perda de conexão entre os nós (PELUSI et al., 2006), portanto as MANETs precisam de protocolos que possam estabelecer uma boa qualidade de serviço (BAI et al., 2003). Essas características móveis permitem que grupos de celulares possam se conectar sem precisar de uma infraestrutura fixa somente criando uma rede ad hoc (PIORKOWSKI et al., 2009).

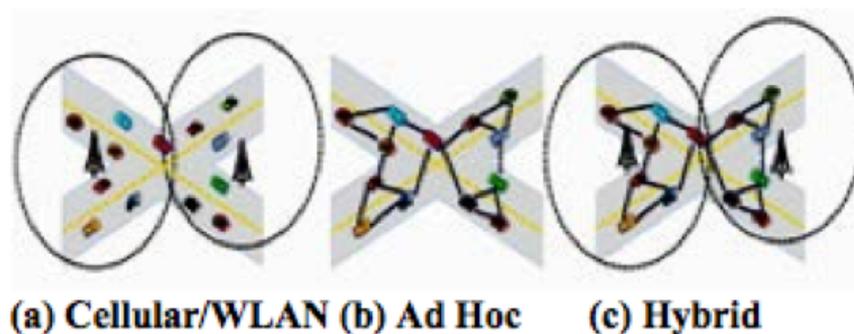
### **2.4 Redes Veiculares Ad Hoc (VANET)**

As *Veicular Ad Hoc Networks* (VANET) são um conjunto de veículos que possuem capacidade computacional e estão ligados em uma rede. As VANETs são um tipo de MANET que não possuem recursos limitados, então não é necessário

uma preocupação com uso de recursos como por exemplo energia, processamento e memória (LOULLOUDES et al., 2010).

A velocidade dos nós é outro fator importante no desenvolvimento de uma VANET (KUMAR; DAVE, 2011), partindo do princípio que veículos possuem uma velocidade muito maior de deslocamento do que dispositivos móveis.

O autor (KUMAR; DAVE, 2011) diz que as redes veiculares podem ser construídas de três maneiras. A primeira utilizando um modelo celular onde pontos fixos espalhados funcionam como ponte para conexão dos veículos. O segundo é uma rede onde as interseções dos veículos são pontes para transmissão dos dados. A terceira maneira para construir uma rede veicular é uma técnica híbrida com a rede Ad Hoc e celular onde existem os pontos fixos e a interseção para trafegar os dados entre os veículos. Esse modelos são exemplificados na figura 1.



**Figura 1:** Tipos de VANET (KUMAR; DAVE, 2011).

De acordo com (PIRAN et al., 2011), com a utilização de sensores nos veículos podem ser desenvolvidos aplicações de dois tipos:

- **Aplicação para Segurança:** são aplicações para reduzir os acidentes envolvendo veículos por exemplo monitorar velocidade, notificação de acidente, notificação de perigo na estrada e aviso de colisão.

- **Aplicação para Auxílio:** são aplicações de conveniência para o condutor do veículo por exemplo para notificações de roubo, de congestionamento, de multas de trânsito e informação sobre o tráfego.

O desempenho de uma VANET pode ser afetado pela topologia da rede que é alterada frequentemente (ANDM; LARSSON, 2010), prejudicando o desenvolvimento de aplicações em tempo real por possuir baixo nível de confiança.

Atualmente as redes VANETs são um desafio para a pesquisa e a engenharia. Sua grande flexibilidade e suporte a várias aplicações do mundo real (GIORDANO et al., 2010) foram atributos que motivaram o desenvolvimento deste trabalho.

## 2.5 Sistemas Multi-Agentes

Segundo (WOOLRIDGE, 2001), sistemas multi-agente são sistemas compostos por agentes inteligentes com capacidade de controlar o próprio comportamento e interagir com o ambiente. O autor (WOOLRIDGE, 2001) sugere que os agentes podem ser agente de software como também robôs ou qualquer outro tipo de recurso que realize tarefas.

De acordo com (WOOLRIDGE, 2001), os agentes em um sistema multi-agente tem algumas características importantes:

- **Autonomia:** os agentes são pelo menos parcialmente autônomos.
- **Vistas locais:** nenhum agente tem uma visão completa do sistema.
- **Descentralização:** não há nenhum agente designado para controlar.

O agente possui um conjunto de tarefas para serem executadas nos nós de uma rede. Os nós possuem uma certa capacidade para realizar essas tarefas e

também possuem uma quantidade limitada de recursos que podem ser utilizados pelos agentes. Para executar essas tarefas os agentes precisam percorrer a rede para poder acessar os recursos dos nós (SHEHORY et al., 1998).

Para (FREITAS et al., 2011), existem duas estratégias para que o agente se locomova sobre a rede. A primeira é através da clonagem dos agentes, enquanto o segundo é chamado de migração dos agentes. Ambos os mecanismos utilizam o mesmo agente que se deslocam pelos nós em uma rede. No entanto, eles se distinguem pelo fato de o agente do primeiro cria uma cópia de si mesmo e este clone é enviado para outro nó, enquanto no segundo, o próprio agente é transmitido para outro nó.

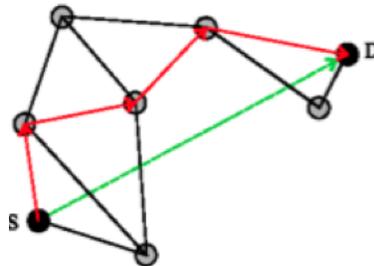
A utilização de um sistema multi-agente sobre VANETs é interessante por que a topologia da rede está em constante mudança, sendo assim é necessário que as escolhas sejam realizadas de acordo com a organização da rede (FREITAS et al., 2011).

Essas características de sistemas multi-agentes citadas anteriormente tornam a utilização das VANETs mais flexível (SHEHORY et al., 1998).

## **2.6 Roteamento geográfico**

Roteamento Geográfico é um conceito de roteamento que se baseia em informações de posição geográfica. Essa organização é proposta para redes sem fio e com base no princípio de que a fonte envia uma mensagem para a posição geográfica do destino em vez de usar o endereço de rede (SHU et al., 2010).

A utilização de roteamento geográfico em um VANET é sugerida em (FREITAS et al., 2011) partindo do princípio que o veículo é maior e possui mais energia facilitando o transporte de um Sistema de posicionamento global(GPS).



**Figura 2:** O Nó S envia um pacote para o nó D e a linha vermelha mostra o caminho da mensagem contornando a janela.

Uma importante exigência para a implementação do roteamento geográfico é que cada nó possa determinar sua própria localização e que o emissor esteja ciente da localização do destino. Essas informações são importantes para que os dados transmitidos possam chegar ao destino (GUISE; PROCTOR, 2011).

Segundo (TAYLOR, 2006), existem duas técnicas para realizar um roteamento geográfico:

- **Encaminhamento Geográfico:** é um método de roteamento pela distância geográfica. Os nós simplesmente transmitem pacotes de dados para o seu vizinho mais próximo até o local de destino especificado no pacote. No entanto, este esquema simples pode levar a zonas mortas que são zonas onde o nó não tem conectividade para região do alvo.
- **Roteamento por Perímetro:** é usado para rotear pacotes em torno das janelas encontrados na rede, assim utilizando alguns algoritmos de grafos para poder contornar essas janelas e atingir os nós que estão no destino.

## 2.7 Google Maps API

Google Maps (antigo Google Local) é uma aplicação web que fornece serviços de mapeamento que pertencem ao Google, gratuito para uso não-comercial,

que alimenta serviços de mapas incluídos em muitos web-sites (MILLER, 2006). Essa plataforma incluindo as ferramentas do Google Maps, Google Ride Finder, o Google Transit e mapas embutidos em sites de terceiros através do Google Maps API. Ele oferece mapas de ruas, um planejador de rotas para viajar a pé, de transporte público, carro, bicicleta e um sistema localizador para vários países ao redor o mundo (MILLER, 2006).



**Figura 3:** Google Maps API exemplo.

Nesse trabalho utilizou a API do Google Maps (GOOGLE, 2012) que fornece uma interface para que seja possível acessar os dados geográficos do Google Maps (GOOGLE, 2012).

Esses serviços da web usam solicitações HTTP (*Hypertext Transfer Protocol*) para especificar URLs (*Uniform Resource Locator*), passando parâmetros de URL como argumentos para os serviços. Normalmente, esses serviços retornam dados nas solicitações HTTP no formato JSON (*JavaScript Object Notation*) ou XML (*Extensible Markup Language*) para análise e/ou processamento por seu aplicativo (GOOGLE, 2012).

Abaixo um exemplo da construção da URL para solicitar serviços ao Google Maps (GOOGLE, 2012).

<http://maps.googleapis.com/maps/api/geocode/json?latlng=40.714224,-73.961452>

## 2.8 Modelo de Movimentos

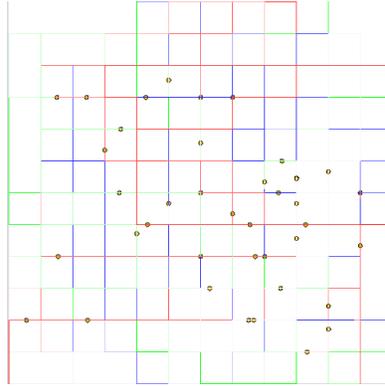
Para realizar as simulações nesse trabalho foram utilizados modelos de movimentos, para simular o comportamento dos nós.

Os modelos de movimentos representam o movimento de componentes móveis, a mudança de sua localização, velocidade e aceleração ao longo do tempo. Tais modelos são frequentemente utilizados para fins de simulação, quando novas técnicas de navegação ou de comunicação são investigados (NICHOLS; HAMMONS, 2007).

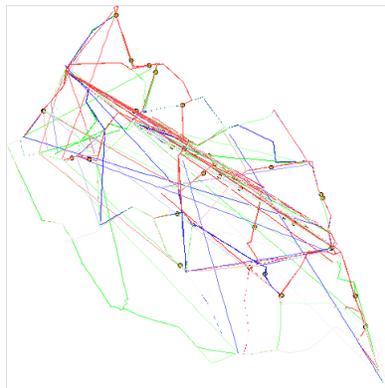
A utilização de modelos de movimentos para realizar simulações em VANET servem para simular os padrões de comportamento dos componentes da rede, por que utilizar componentes reais é inviável (FREITAS et al., 2011).

O autor (SUN; SAUVOLA, 2002) sugere duas abordagens para construir um modelo de movimento:

- **Analítico:** esse modelo de movimento é construído por uma fórmula matemática que define os padrões de comportamento dos elementos representados no modelo.
- **Simulado:** é construído através da observação do comportamento dos nós no ambiente que se deseja criar a simulação.



**Figura 4:** Exemplo de Modelo de Movimento Analítico.



**Figura 5:** Exemplo de Modelo de Movimento Simulado.

### 3 CONTEXTUALIZAÇÃO DO PROBLEMA

O cenário de aplicação apresentado neste trabalho é uma rede veicular móvel. Para modelar o movimento da cidade foram utilizados o modelo de movimento Manhattan e outro baseado em requerimentos enviados ao Google Maps que será apresentado posteriormente. Os nós, com a ajuda de agentes de software, realizam coletas de dados em áreas específicas do mapa utilizado, que são chamados de regiões-alvo (TRs). Várias TRs podem ser definidas em um mapa e diferentes agentes de software representam as diferentes aplicações que podem ser implantadas no sistema. A figura é um exemplo do cenário a ser utilizado nesse trabalho.

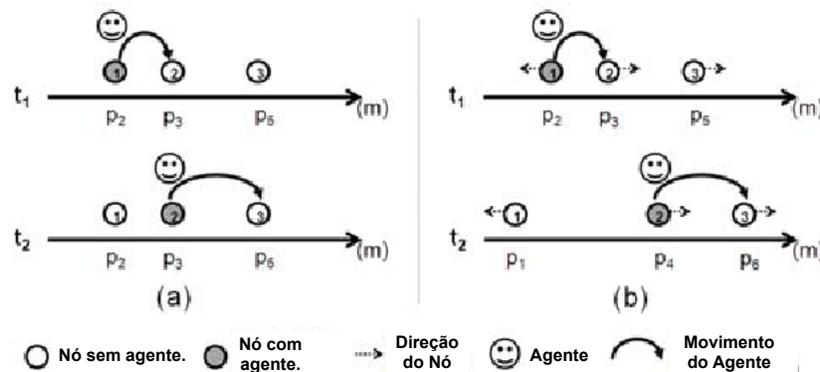


**Figura 6:** Exemplo de cenário e comportamento dos nós.

A coleta de dados é um serviço implementado por agentes de software. O desempenho deste serviço é ajustado para as necessidades específicas que compõem uma missão com objetivo de ser realizada pelos agentes. Os nós móveis fornecem suporte para os agentes que executam os seus serviços, isto é, os nós devem oferecer recursos de computação, dispositivos e sensores que podem ser utilizados pelos agentes para a aquisição de dados.

As missões representadas e implementadas pelos agentes de software podem ser enviadas a TRs diferentes para responder às necessidades de serviço e de usuários. Isto requer que os agentes de software migrem de alguns nós para os

outros, de acordo com as necessidades de mobilidade e da mobilidade dos nós, a fim de alcançar o destino alvo, ou seja, as TRs correspondentes e realizarem as suas respectivas tarefas.



**Figura 7:** Migração do agente.

Um cenário ilustrativo para a aplicação apresentada acima é de uma frota de táxis em uma grande cidade, como São Paulo, Rio de Janeiro ou Belo Horizonte, fornecendo o suporte à plataforma básica. São realizados diferentes tipos de pedidos em diferentes partes da cidade. Estes pedidos são atendidos por agentes de software que migram entre os táxis de acordo com o encontro dos nós. Pode-se pensar em um negócio em que as empresas de táxi cobrem uma taxa para fornecer suporte à plataforma com base para os agentes de software. O uso de agentes de software móveis torna possível a injeção de diversas aplicações no sistema, como por exemplo, os agentes que têm a missão de coletar dados sobre poluentes em uma área com alta concentração de indústrias, ou para coletar dados sobre o fluxo de tráfego e congestionamentos.

Outras aplicações possíveis podem ser definidas como em situações onde a infra-estrutura de comunicação convencional está fora de ordem, devido a eventos catastróficos, por exemplo, as unidades móveis das equipes de emergência podem

formar uma rede para coletar informações durante uma catástrofe. Novamente, a utilização de agentes de software representa uma vantagem, uma vez que permite a inserção de novos pedidos de dados sob demanda, os quais podem se mover através da rede de acordo com os pedidos dos utilizadores.

No cenário apresentado acima, o problema a ser resolvido é como realizar de forma eficiente as migrações dos agentes entre os nós em uma rede com conectividade intermitente, de modo que os agentes eventualmente atinjam os nós dentro de suas respectivas TRs.

O tempo em que os agentes são acolhidos por nós localizados dentro destas TRs também tem que ser maximizado, enquanto que o número de migrações tem de ser minimizado. Note-se que o problema vai além de um mecanismo de encaminhamento de definição geográfica, tendo dois aspectos principais a serem decididos. O primeiro aspecto é a ligação entre os nós intermitente requer uma abordagem comunicação oportunista, explorando as ligações pontuais estabelecidas entre os nós, mas observando o contexto de execução atual sistema. O segundo aspecto refere-se aos requisitos da aplicação, definido na relação das TRs para cada agente, e os recursos disponíveis para realizar as tarefas. Os requisitos da aplicação podem variar muito de acordo com as necessidades do usuário, e em relação aos recursos disponíveis. Um exemplo é uma VANET em que todos os nós não têm acesso aos mesmos recursos, alguns nós possuem informações geográficas (isto é, ter acesso ao GPS) enquanto que outros não.

## **4 METODOLOGIA**

### **4.1 Classificação do Trabalho**

O trabalho, quanto à sua natureza, é considerado como de Pesquisa Aplicada, pois visa o tratamento de um problema concreto. Espera-se ao fim, que a própria pesquisa apresente resultados sólidos em se tratando da resolução do problema apresentado.

Quanto aos objetivos, se classifica como pesquisa exploratória, visto o objetivo de combinar práticas existentes (algoritmos auto-organizáveis) em busca da resolução de um problema.

No que se refere aos procedimentos, é considerada como pesquisa experimental, novamente por se apresentar como aplicação de métodos e técnicas. Para a realização do trabalho, se fará uso de ensaios e estudos de laboratório. Onde, a partir do simulador GRUBiX, poderão ser feitas as simulações, testes e avaliações dos algoritmos.

Ainda em relação às práticas metodológicas, a pesquisa também se classifica como quantitativa. Serão realizados testes, simulações e avaliações de forma quantitativa. Ao fim serão comparados os valores numéricos de cada experimento.

### **4.2 Procedimentos metodológicos**

Os experimentos realizados neste trabalho basearam-se em simulação. O uso de simulação justificou-se pela complexidade em se aplicar o algoritmo proposto em uma grande quantidade de veículos, questões financeiras e relacionadas ao tempo para se testar o algoritmo em uma grande variedade de cenários também foram avaliadas. Outro fator importante no uso do simulador é que uma cidade

possui um ambiente que proporciona obstáculos que podem atrapalhar uma melhor avaliação de cada nível de inteligência.

A próxima seção introduz os conceitos referentes ao simulador utilizado, bem como os conceitos de simulação orientada a eventos.

### **4.3 GRUBiX e Simulação Orientada a Eventos**

Para desenvolvimento do algoritmo proposto neste trabalho, foi utilizado o simulador de redes sem fio GRUBiX, desenvolvido no Departamento de Ciência da Computação, pelo grupo de pesquisas GRUBi, da Universidade Federal de Lavras (??).

Este simulador é uma evolução do projeto Shox (??). Assim como o Shox, o simulador GRUBiX caracteriza-se como um ambiente de simulação orientado a eventos. De forma simplificada, em um ambiente de simulação orientado a eventos, a linha do tempo de uma simulação é representada por uma lista de eventos, em que cada novo evento é inserido nesta lista. A posição em que cada evento é inserido nesta lista varia de acordo com o momento em que cada evento deve ocorrer.

Adicionalmente, esses simuladores oferecem um arcabouço de ferramentas para que se possa desenvolver toda a pilha de protocolos presente em um nó sem fio. Essa disponibilidade permite que cada camada da pilha de protocolos seja personalizada de acordo com as necessidades de cada aplicação.

Assim como o Shox, o simulador GRUBiX utiliza a linguagem de programação Java. O uso de uma linguagem orientada a objetos, nesse caso Java, indica que as camadas da pilha de protocolos sejam modeladas como objetos. O uso da orientação a objetos permite que cada camada seja personalizada através de mecanismos de herança. Portanto, basta que a nova camada personalizada herde as

funcionalidade de uma camada base para que se tenha a possibilidade de alterar o comportamento padrão da camada.

O desenvolvimento deste trabalho caracteriza-se por um algoritmo sobre a camada de aplicação da pilha de protocolos dos nós sem fio.

#### **4.4 Solução Proposta**

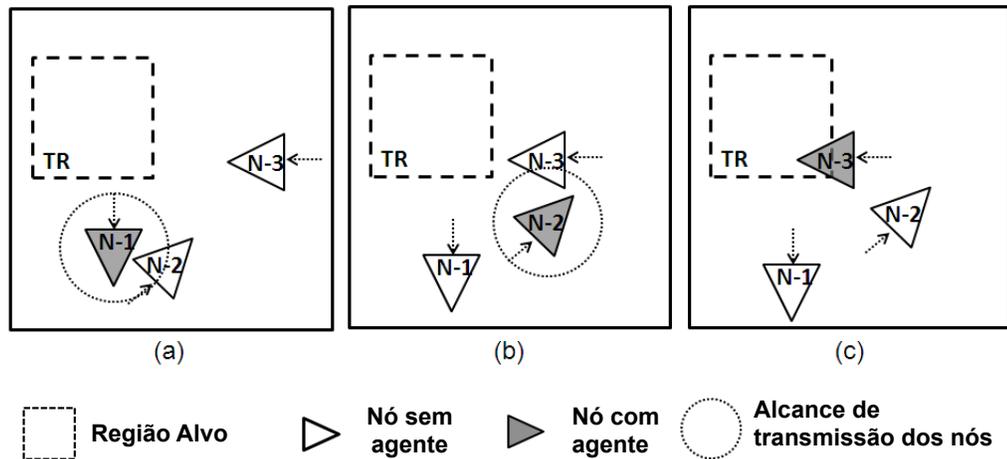
Observando o cenário apresentado no capítulo anterior e o problema derivado deste cenário, a solução proposta é explorar as informações de contexto geográfico para realizar migrações dos agentes inteligentes entre os nós, em que os agentes têm o objetivo de chegar a nós de que os levam para as respectivas TRs solicitadas em cada missão.

A utilização de agentes de software móveis é justificada, em primeiro lugar por a suposição de que os nós da rede não estão ligados às missões executadas pelos agentes(FREITAS TALES HEIMFARTH, 2012), isto é, eles não mudam os seus movimentos devido às necessidades de uma missão específica.

Os nós possuem a única função básica de transmitir os agentes entre eles, de acordo com as necessidades dos agentes. Além disso, vários agentes com missões diferentes podem entrar ou sair da rede, solicitando os dados de TRs diferentes. Esta forma ad hoc e dinâmica de operação requer flexibilidade para injetar novas missões na rede, o que não pode ser feito a partir do início do tempo de execução do sistema. Agentes de software móveis são capazes de proporcionar tal flexibilidade e ao mesmo tempo executar a migração inteligente entre os nós, sem a desvantagem de ser restrito a um mecanismo de encaminhamento de dado que está instalado nos nós que compõem a rede, ou seja, os agentes podem usar diferentes estratégias e diferentes critérios para executar seus movimentos. Essa é uma característica explorada nesse trabalho. Além disso, o uso de agentes oferece

ainda a possibilidade de o mesmo agente adaptar-se de acordo com o contexto de execução atual e alterar o mecanismo de decisão usada para mover-se entre os nós.

A figura 8 mostra um exemplo no qual um primeiro agente está sendo transportado pelo nó N-1 que se move fora da TR que o agente deve executar a sua missão. Além de mover para fora da TR, nó N-1 afasta-se da TR (Figura 8a), isto é, numa direção contrária em relação ao que o que é desejado. Durante o seu movimento, o nó N-1, no entanto encontra um outro nó, N-2, que se move na direção que pode levar o agente mais perto de TR. Neste momento, uma decisão de migrar ou não a N-2 tem de ser tomada. Assumindo-se que a decisão é positiva, o agente migra para N-2 e segue com o nó no trajeto dele. Eventualmente N-2 encontra um outro nó, N-3, que se move na direção de TR (Figura 8b). Como a N-2 não está se movendo em direção de TR, o agente decide migrar para o nó de N-3, e, finalmente, o agente consegue chegar à sua TR (Figura 8c).



**Figura 8:** Exemplo de uma migração bem-sucedida de um Agente.

A figura 8 mostra um exemplo de migração de sucesso de um agente a partir de um nó de fora da TR que não estava em movimento em relação à região

(N-1) a um outro nó que eventualmente transporta o agente a sua TR (N-3). Esta migração do agente foi realizada por meio de dois saltos (primeiro a partir de N-1 e N-2 e, em seguida, a partir de N-2 N-3), explorando o conceito de transporte de agente (TEI et al., 2005) como explicado anteriormente.

Como apresentado na Figura 8, nós N-1 e N-3 conheceram um outro nó durante os seus movimentos e nó N-2 encontra os outros dois, mas em um cenário real, com uma maior densidade de nós, a conexão entre os nós acontecem com muito mais frequência. Neste caso, os agentes não devem apenas migrar para um nó, mas em vez disso, eles têm de considerar se vale ou não migrar para um nó considerando os custos e benefícios de uma tal ação. Assim, há uma necessidade de ter uma boa política e uma idéia sobre o que considerar quando se decidir sobre a migração ou não.

O objetivo dos agentes de software é eventualmente atingir os nós que estão dentro da TR através da migração do agente através dos nós que se deslocam no sentido desta área, tal como no exemplo apresentado na Figura 8. Neste sentido, as informações de contexto estão relacionadas com as posições geográficas dos nós, bem como as indicações dos seus movimentos seguintes.

Os fatos acima mencionados motivaram o desenvolvimento da proposta das três diferentes abordagens para executar a migração do agente (FREITAS TALE HEIMFARTH, 2012), com base em diferentes níveis de inteligência para explorar os diferentes níveis de informação de contexto.

#### **4.5 Nível de Inteligência do Agente**

A primeira condição que um agente tem que considerar é se o nó atual está dentro ou fora da TR. No primeiro caso, o agente está em um nó que ele pode desempenhar a sua missão e não precisa migrar para outros nós (FREITAS TA-

LES HEIMFARTH, 2012). Caso contrário, ele tem que esperar que o nó atual chegue à TR para realizar a sua missão ou tentar migrar para outro nó, quando encontrar um outro nó que possui uma avaliação melhor. Com base nessas considerações, o comportamento necessário para que o agente possa analisar se pode executar a missão ou deve migrar para outro nó pode ser definida como apresentada na código abaixo.

**Listing 1:** Algoritmo, em pseudo código, que define o comportamento do agente.

---

```

1 While (true)
2   While (Inside_TR)
3     perform(Mission);
4   End_While
5   While (!Inside_TR)
6     If Meeting_Node != null then
7       Worth = evaluateMigration(Meeting_Node);
8       If Worth == true then
9         migrate(Meeting_Node);
10      End_If
11    End_If
12  End_While
13 End_While

```

---

O código 1 acima mostra que o agente está realizando a sua missão dentro da TR (linhas 02 a 04) ou espera por uma comunicação com outro nó, enquanto o nó atual está fora da TR (linhas 05 a 12). Quando o nó atual encontra outro nó, ele avalia se vale a pena ou não passar para este nó (linha 07), e se assim for, ele migra para o nó de reunião (linha 09).

O método *evaluateMigration(Meeting\_Node)* (linha 7 da Lista 1), é responsável pela decisão sobre a migração do agente para um outro nó conectado. Para implementar esse método, o agente tem de adquirir a informação geográficas

necessária sobre o nó atual e o outro nó conectado, para que uma decisão possa ser tomada. As informações dos pedidos do agente e como ele usa essa informação define seu nível de inteligência. Esta proposta considera três níveis de inteligência distinguidos pela informação de que os agentes usam para tomar a decisão sobre como migrar ou não para um nó de reunião, que são detalhados a seguir.

#### 4.5.1 Nível de Inteligência Baseada no Destino

Para o primeiro nível de inteligência, chamado de *Inteligência Baseada no Destino*, o agente é apenas capaz de saber se o nó hospedeiro está dentro ou fora da TR, e se o próximo destino de um nó está dentro da TR ou não. Representa, por exemplo, uma situação em que o agente só tem acesso a um mapa de curso e sem acesso a informações geográficas ou de posicionamento adicional fornecida por exemplo uma unidade GPS (Global Positioning System), tal como a posição exata ou o trajeto seguido pelo nó móvel ou a sua direção.

No método *evaluateMigration(Meeting\_Node)* a implementação do *nível de inteligência Baseado no Destino*, realiza a seguinte sequência de passos: 1) se o nó atual tem um destino dentro da TR, o agente continua neste nó, ou seja, decide não migrar, 2) se o nó atual não tem o seu destino, como uma posição dentro de TR, no caso em que o destino de um nó de encontro está dentro da TR, o agente decide migrar para o nó de encontro, de outro modo o agente continua no nó atual. O código 2 apresenta este algoritmo, em que a avaliação do destino do nó atual é apresentado na linha 01, enquanto que a avaliação do destino do nó de reunião é apresentado na linha 05.

**Listing 2:** implementação do *evaluateMigration(Meeting\_Node)* para o nível de inteligência baseado no destino.

---

```

1  If (Current_Node.getDestination() == Inside_TR) then
2      decision = false;
```

```
3 End_If
4 Else
5     If (Meeting_Node.getDestination() == Inside_TR) then
6         decision = true;
7     End_If
8     Else
9         decision = false;
10    End_Else
11 End_Else
12 return decision;
```

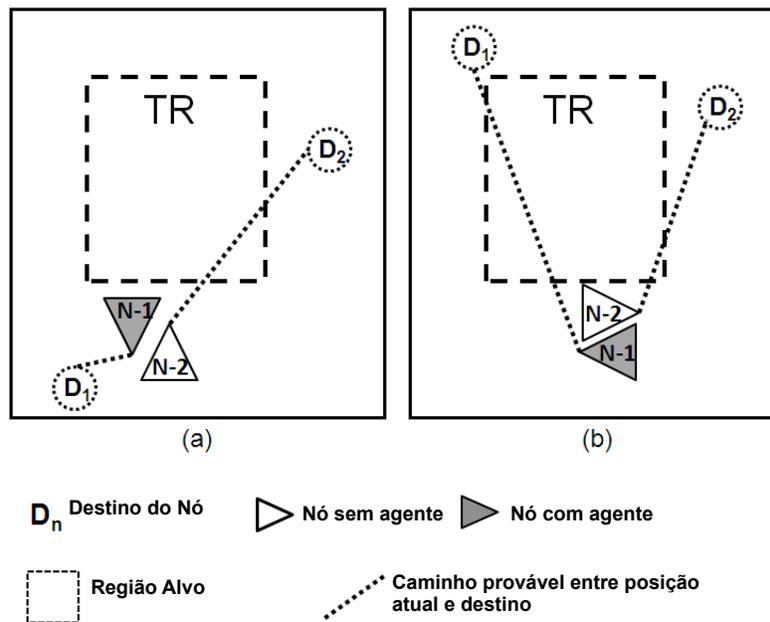
---

#### 4.5.2 Nível de Inteligência Utilizando Caminho Direto

O segundo nível de inteligência é uma evolução da inteligência utilizado pelo primeiro. Para este nível, considera-se que o agente possui não só a informação disponível a partir do mapa simples, mas também a posição exata do respectivo nó hospedeiro, bem como os destinos deste nó e dos nós conectados. Este é um caso em que o agente é assume ter acesso a um sistema de posicionamento, tais como GPS, mas não tem informação sobre as rotas, ou seja, a partir de um mapa detalhado.

O mecanismo de inteligência utilizado neste nível é baseada na probabilidade de um nó passar através da TR (FREITAS TALES HEIMFARTH, 2012). Para calcular a probabilidade, a avaliação da via direta, que liga a posição corrente e os nós de destino é realizada e conseqüentemente é chamado de *Nível de Inteligência utilizando Caminho Direto*. A probabilidade é calculada através da criação de um caminho direto de ligar a posição atual e a posição de destino e avaliar o comprimento do caminho que este permanece dentro da TR. Quanto maior o comprimento deste caminho dentro da TR, maior é a probabilidade de o agente selecionar este nó para ficar ou para migrar. Se a probabilidade de um nó conectado

é maior do que para o nó atual, o agente migra, caso contrário, ele permanece no nó atual. A figura 9 apresenta dois exemplos da aplicação desta estratégia.



**Figura 9:** Exemplos para a avaliação realizada pelo algoritmo do nível de inteligência de caminho direto.

Observe que na figura 9a, a linha que liga a posição atual do nó de N-1, que é atualmente o agente de transporte, não atravessa a TR, enquanto a linha correspondente para o nó N-2 tem uma porção que está dentro do TR. Seguindo a estratégia de raciocínio descrito, o agente irá migrar a partir do nó N-1 para o nó N-2. Na Figura 9b, um exemplo complementar é apresentado. Nesta situação, as linhas que ligam as posições atuais e os destinos dos dois nós têm porções dentro do TR, mas o comprimento do trajeto no interior da área de nó de N-1 é maior do que aquele para o nó de N-2. Como resultado, o agente que está atualmente em nó N-1 não migrará para o número do nó N-2. A lista 3 mostra o algoritmo

implementado no método *evaluateMigration(Encontro\_Node)* para o nível de inteligência utilizando caminho direto.

**Listing 3:** implementação do *evaluateMigration (evaluateMigrate(Meeting\_Node))* para o nível de inteligência utilizando caminho direto.

---

```

1  prob_Current = Calculate_Prob (Current_Node.getDestination ())
2  prob_Meeting_Node = Calculate_Prob (Current_Meeting_Node .
    getDestination ())
3  If (prob_Current > prob_Meeting_Node) then
4      decision = false;
5  End_If
6  Else
7      decision = true;
8  End_Else
9  return decision;

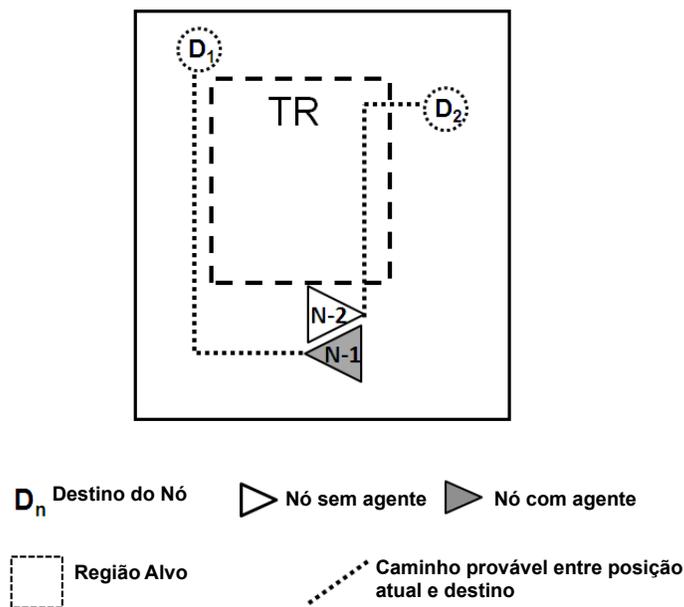
```

---

#### 4.5.3 Nível de Inteligência Avançada

No terceiro nível de inteligência o agente considera que as abordagens anteriores podem não apresentar um desempenho bom o suficiente em alguns casos. Por exemplo, o caminho direto pode falhar nos casos em que o nó assume um percurso que não coincide com a a probabilidade de passar pela TR calculada, ou se desvia muito da linha reta traçada entre a posição atual e o seu ponto de destino. Esta é a situação apresentada no exemplo apresentado na figura 9b. Neste exemplo, o nó N-1 não pode passar no interior da TR para chegar ao seu destino, enquanto que o nó N-2 pode ser um caminho que efetivamente passa através TR. Esta situação é representada na figura 10. A inteligência caminho direto não pode considerar essa hipótese, porque ele não tem informações sobre a rota que os nós vão seguir.

Observando o tipo de problema apresentado acima, o terceiro tipo de nível de inteligência possui o conhecimento da rota, considerando a rota completa e posições dos nós atuais para seus destinos. Este recurso permite que o agente possa calcular o caminho mais curto a partir da posição atual para a TR ou mesmo para posições mais próximas da TR.



**Figura 10:** Exemplo de caminhos rota completa considerados pelo agente utilizando a inteligência Avançada.

O cálculo do caminho mais curto permite que o agente decida migrar para um nó, que possui um trajeto mais direto para o TR, mesmo no caso em que o nó hospedeiro também se mova no sentido da TR. Além disso, a capacidade de analisar destinos mais perto da TR representa uma melhoria em relação às outras estratégias. Isto porque para os nós que não estão se movendo em direção à TR, mas podem ter destinos mais próximos a ele e não tem probabilidade de passar no interior da TR, o agente usando a *rota avançada* pode considerar a possibilidade

de migrar para eles, enquanto isso não ocorrer nos dois níveis de inteligência anteriormente apresentados. Neste caso, por exemplo, com o *nível de inteligência do caminho direto*, o agente iria ficar em seu nó atual. Por outro lado, para o desenvolvimento do *nível de inteligência avançado*, o agente iria considerar a rota completa de cada nó, tanto aquele em que o agente está alojado e o nó conectado, a fim de definir qual vai passar para mais perto do TR. Selecionando o nó que passaria mais perto da TR aumenta a oportunidade de o agente encontrar outros nós que estão se movendo para a TR.

O código 4 apresenta o algoritmo utilizado pelo agente utilizando o *nível de inteligência avançado* para implementar o *evaluateMigration(Meeting\_Node)*. Linha 1 testa os casos em que ambos os nós têm destinos, dentro ou fora TR. Neste caso, a decisão de mudar ou não para o nó de encontro considera o caminho mais curto em relação a TR (linhas 02-07). Caso contrário, se um dos nós tem destino dentro TR e os outros fora TR, a decisão será a de migrar se o nó encontrado é o que se move em direção a TR (linhas 10 - 12), ou para não migrar se o nó atual é o único que se move em direção ao TR (linhas 13-15).

**Listing 4:** *evaluateMigrate(Meeting\_Node)* implementação para o nível de Inteligência Avançada.

---

```

1  If (( Current_Node.getDestination() == Inside_TR && Meeting_Node.
      getDestination() == Inside_TR ) ||
2   ( Current_Node.getDestination() != Inside_TR && Meeting_Node.
      getDestination() != Inside_TR )) then
3   If ( Current_Node.path(TR) < Meeting_Node.path(TR) ) then
4     decision = false;
5   End_If
6   Else
7     decision = true;
8   End_Else

```

```
9 End_If
10 Else
11     If (Meeting_Node.getDestination() == Inside_TR) then
12         decision = true;
13     End_If
14     Else
15         decision = false;
16     End_Else
17 End_Else
18 return decision;
```

---

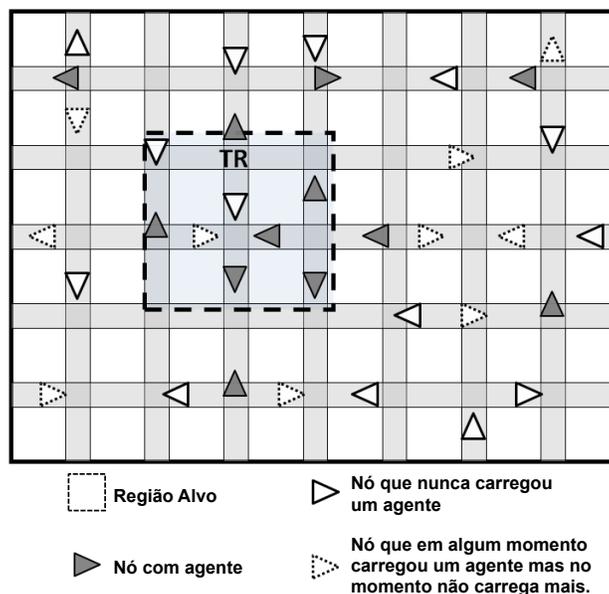
## 5 RESULTADO E DISCUSSÃO

A avaliação dos diferentes níveis de inteligência apresentados acima foi realizada por meio de duas métricas: 1) número de saltos realizadas por agente em cada simulação e 2) Percentagem da simulação em tempo durante o qual os agentes ficam dentro da TR. A primeira métrica fornece uma visão sobre a sobrecarga em termos de uso dos recursos de comunicação, enquanto a segunda fornece informações sobre a eficiência de cada algoritmo em controlar os agentes para a direção da TR e mantê-los dentro. É importante notar que a primeira métrica considera a sobrecarga devido apenas à migração do agente em si e é por isso que os resultados são apresentados em número de migrações em vez de bytes transmitidos por migração. Isto é explicado pelo fato de que, no presente trabalho, considera-se que os dados de transmissão dos agentes durante a sua migração são de tamanho semelhante para todos os agentes, que é uma suposição razoável. Considerando que em um uso real desta abordagem, eles só teriam dados processados por algoritmos que eles iriam implementar como serviços, por exemplo, agregação de dados, e não transportar grandes quantidades de dados brutos.

### 5.1 Ambiente Simulado

As simulações foram realizadas, utilizando uma rede de sensores veicular (VSN) como estudo de caso, em que os nós sensores móveis da VSN são uma frota de táxis. Nesta aplicação, os táxis se deslocam em uma cidade para responder as solicitações dos clientes. Durante a movimentação pela cidade, existe alguma probabilidade dos táxis passarem pela área de interesse, ou seja, as TRs, assim o agente pode aproveitar os táxis como transporte enquanto for conveniente, de acordo com a localização do seus TRs.

Foram utilizados dois ambientes diferentes como cenários para as experiências realizadas. A primeira é uma área de quadrado que representa um mapa de uma cidade, dividido em blocos, em que uma TR é definida. A Figura 11 apresenta esse ambiente e a área destacada indica a TR. Observe que nessa imagem do ambiente simulado apresentado na Figura 11 existem nós que possuem um agente, enquanto outros não têm. Entre os últimos, alguns tinham um agente no passado, mas que havia transmitido a outros nós.



**Figura 11:** Modelo do ambiente simulado representando um mapa quadrado dividido em quarteirões regulares.

O movimento dos nós no primeiro ambiente se baseia-se no modelo de mobilidade Manhattan (BAI et al., 2003). Os nós movem ao longo das ruas entre os blocos que compõem o cenário da cidade. Eles selecionam um determinado ponto de uma rua e se movem em direção a ela. Quando um nó atinge um cruzamento, ele escolhe uma direção a seguir: norte, sul, oeste, ou leste. Esta escolha da direção é aleatória, mas considera a direção que o nó está dirigindo, ou seja, se o nó está

se movendo para um ponto localizado a nordeste de sua posição atual, selecionar aleatoriamente para se mover para o norte ou leste.

Se o nó é um táxi com um destino definido, ou seja, um táxi que está transportando um passageiro ou vai levar um passageiro, ele se move de preferência em linha reta para o destino. Isto é feito através da atribuição de uma maior probabilidade de a escolha de se mover para a frente, quer dizer para continuar na mesma direção. Por exemplo, se o destino for em um local ao nordeste em relação à localização do nó atual e o nó está se movendo para o norte, move-se de preferência em linha reta na direção norte, até atingir o nível ao norte do destino. Em seguida, ele vira para leste e segue nesta direção até que chega ao destino. Quando um nó chega ao seu destino, ele começa a procurar por um passageiro ou vai para outro destino, ou seja, tem um outro passageiro.

O segundo ambiente representa as ruas da cidade de Lavras/MG com dados geográficos fornecidos pela API do Google Maps (GOOGLE, 2012) (PIORKOWSKI et al., 2009). A Figura 12 apresenta a região do mapa no qual os pontos representam os nós móveis (isto é os táxis) que compõem a rede.

### **5.1.1 Configuração da Simulação**

O primeiro ambiente simulado é de  $2,5 \text{ Km} \times 2,5 \text{ Km}$  divididos em área de  $20 \times 20$  blocos de 70 metros de cada lado. A TR considerada possui dimensões de  $4 \times 4$  blocos. Os tamanhos de todo o ambiente e as TR são adequados para a primeira avaliação da abordagem proposta, pois a região ao redor da TR representa o ambiente onde os agentes devem encontrar um nó para alcançar a TR. Por outro lado, o TR é consideravelmente menor do que a área total do mapa. TRs maiores em relação à área total podem tornar a situação mais simples e mais fácil para todas as variações de raciocínio. Este fato motiva o uso de uma TR pequena, o que



uma antena que propaga em todas as direções com um alcance de 90 metros, o que é bastante realista, mesmo considerando um ambiente como uma cidade em que os blocos com prédios dificultam as comunicações sem fio, como discutido em (GIORDANO et al., 2010).

O período de despertar para procurar vizinhos para se comunicarem com nós que hospeda o agente foi de 5 segundos. Este valor foi empiricamente estabelecido. Verificou-se por meio de simulações que valores menores do que este, não oferece qualquer melhoria significativa nos resultados. Por outro lado, valores superiores a este causam um impacto negativo nos resultados, como os nós podem ter um deslocamento significativo em função da sua velocidade atual, então perdem a oportunidade de comunicar com os nós vizinhos. Um resumo dos parâmetros de simulação para o ambiente em primeiro lugar é apresentada na Tabela 1.

<b>Parâmetro</b>	<b>Valor</b>
Dimensão da Área	2,5 Km × 2,5 Km
Dimensão do Quarteirão	70m × 70m
Dimensão da TR	4 × 4 Quarteirões
Velocidade dos nós	30Km/h - 60Km/h
Numero de nós (destinos)	60(9,6 nós/Km <sup>2</sup> )
Número de agentes	5
Alcance da Comunicação dos Nós	80m
Período de Transmissão	5 Segundos

**Tabela 1:** Parametros de Simulação.

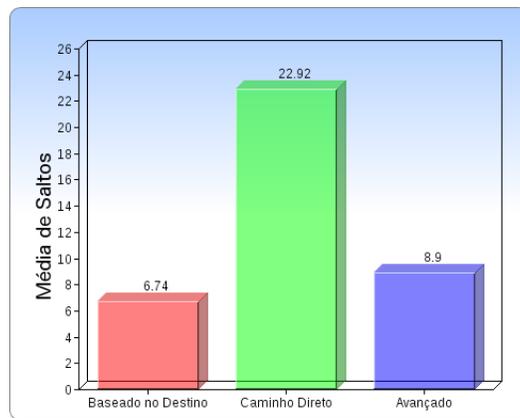
Para o segundo ambiente, a área total é de 10 Km × 10 Km da cidade de Lavras/MG e uma região correspondente a 4% desta área é selecionada como a TR, de forma semelhante à TR no primeiro ambiente. Os números de táxis que compõem a rede são os mesmos que no conjunto de simulação anterior (60 nós), mas para manter as condições equivalentes, tal como a densidade de nós apresentado no primeiro cenário, o alcance de comunicação foi ajustada até cerca

de 500 metros, a fim de que o número médio de vizinhos por nó é mantida. A velocidade dos nós varia no mesmo intervalo que o cenário do primeiro ambiente, assim como o resto dos parâmetros são os mesmos que os utilizados no primeiro ambiente. Assim mantém a equivalência entre ambos os cenários de simulação, permitindo a comparação entre os resultados obtidos em cada simulação.

## 5.2 Resultados das simulações utilizando o Modelo de Movimento em Quadras

Figura 13 demonstra as médias de resultados obtidos relacionados com a primeira métrica, o erro representa intervalo de confiança 95%. Os resultados são apresentados com uma média de migrações dos agentes que de acordo com seus diferentes níveis de inteligentes (*baseado no destino, caminho direto ou avançado*). A média foi calculada somando a quantidade de migrações que os agentes fizeram nas simulações e dividindo pela quantidade de simulações.

Os resultados revelam que dois níveis (o baseado no destino e o avançado) mantem os números baixos de migrações enquanto o Caminho Direto apresenta maior número de migrações em relação a estes dois primeiros. Estas observações podem ser compreendidas pela maneira que cada nível executa o algoritmo. Tanto o baseado no destino e o avançado apenas decidem migrar quando tem certeza que o outro nó é melhor, analisando a informação de que eles são capazes de analisar. Por outro lado, o caminho direto arrisca mais. Por exemplo, se o agente usando o raciocínio caminho direto está em um nó que tem 50% de chance de passar pelo TR e encontra outro nó com 51%, o agente migra para este nó. Analisando as médias ao longo que a simulação é executada, o nível de inteligência baseado no destino apresenta os números mais baixos de migrações, seguido pelo avançado e caminho direto.



**Figura 13:** Média de migrações por agentes no modelo de movimento quadras.

Apesar da alta variabilidade dos resultados (Tabela 3), é possível observar que o algoritmo baseado no destino apresenta claramente os valores mais baixos, na maioria dos ensaios. Os outros dois tipos de raciocínio proporcionam melhores resultados em média, mantendo os agentes na TR.

Numero de nós	Nível de Inteligência	Média	Desvio Padrão
60	Baseado no Destino	7	1,37
60	Caminho Direto	19	2,117
60	Avançado	27	1,925

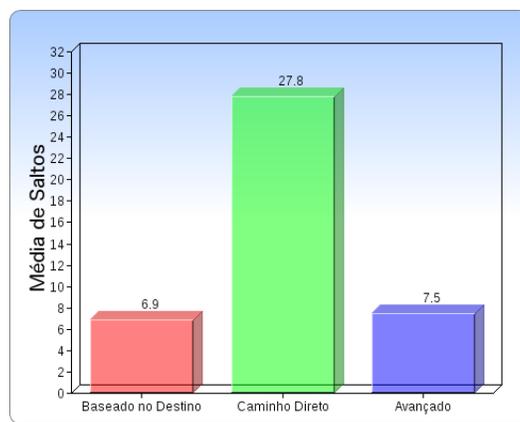
**Tabela 2:** Os valores de média e desvio padrão para a métrica que avalia o tempo dentro da TR no modelo Quadras.

### 5.3 Resultados das simulações utilizando o modelo de movimento baseado em dados geográficos do google maps.

O segundo conjunto simulado avaliou o desempenho dos diferentes níveis de inteligência em um cenário mais realista, construído através dos dados geográficos do Google Maps (GOOGLE, 2012) (PELUSI et al., 2006). Depois das experiências realizadas no mapa quadrado usando o modelo de movimento Manhattan, este se-

gundo conjunto de simulações proporcionar meios para avaliar o desempenho da abordagem proposta em um ambiente menos favorável e previsível.

A figura 14 apresenta as médias de resultados obtidos relacionados com a primeira métrica, com barras representam intervalo de confiança de 95%. Estes resultados seguem a mesma apresentação que é utilizado na subseção anterior, comparando o número médio de migrações dos agentes de acordo com os diferentes níveis de inteligentes (*baseado no destino, caminho direto ou avançado*).



**Figura 14:** Média de migrações por agentes no modelo de movimento Google Maps.

É possível observar a coerência dos resultados alcançados nesta segunda simulação em conjunto com a comparação da primeira simulação apresentada anteriormente. Note-se que em relação aos outros níveis de inteligência preservam a mesma relação, em que o baseado no destino e o avançado mantem um menor o número de migrações realizadas pelo agente enquanto o caminho direto possui um desempenho pior.

Os resultados alcançados neste segundo cenário, em geral, apresenta um pior desempenho (apresentando mais saltos e menor porcentagem do tempo dentro da TR) do que no primeiro cenário. Isto pode ser explicado pela geografia do

Numero de nós	Nível de Inteligência	Média	Desvio Padrão
60	Baseado no Destino	8	0,96
60	Caminho Direto	21	2,117
60	Avançado	29	3,528

**Tabela 3:** Os valores de média e desvio padrão para a métrica que avalia o tempo dentro da TR no modelo do Google Maps.

mapa. O primeiro cenário é uma área avaliada regularmente quadrada dividida em blocos quadrados do mesmo tamanho, enquanto Lavras é muito irregular cheia de ruas com mão únicas e com blocos de diferentes formas e dimensões. Além disso, como os nós no segundo modelo de movimento simula movimentos de táxis em uma cidade como Lavras, muitos eventos que não são considerados no cenário anterior podem ocorrer, tais como carros usando ruas secundárias, vias com uma grande concentração de carros, entre outros.

A análise cruzada dos resultados de ambos os cenários e das duas métricas torna possível considerar o uso do nível de inteligência avançada para fornecer suporte a aplicações que rodam em cima de nós móveis, tais como o proposto neste trabalho. Mesmo o nível de raciocínio baseado em destino fornece resultados consideráveis que são melhorados pelo aumento da capacidade de analisar mais informações, ou seja, uma melhora para o caminho direto, e, finalmente, estes resultados apresentam melhora ainda maior com a utilização do nível utilizando a inteligência avançada.

Apesar da desvantagem apresentada pela maior sobrecarga do caminho direto em relação ao nível baseado no destino, os melhores resultados obtidos pelo agente com inteligência avançada em todas as métricas mostra o valor em utilizar informações mais relevantes.

Observando os resultados apresentados nas Figuras 13 e 14, os algoritmos com um nível de inteligência baseado no destino apresentam um melhor desem-

penho no cenário da primeira simulação utilizando o modelo de quadras com as quadras regularmente divididas. Na segunda simulação utilizando dados reais do Google Maps os resultados não foram muito satisfatórios.

Isso apresenta uma evidência de que os níveis mais elevados inteligentes são mais robustos mantendo os seus comportamentos em diferentes cenários, mas também que é possível tirar proveito no nível baseado no destino em cenários mais regulares. Este fato suporta a ideia proposta de permitir que os agentes selecionem os nível de inteligente para usar de acordo com o contexto de tempo de execução real.

## 6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou uma abordagem em que os agentes de software móveis são usados para divulgar dados em VANETs. A utilização de informação geográfica foi planejada para apoiar as decisões dos agentes sobre suas migrações entre os nós na rede. Foram propostos diferentes níveis de inteligência para serem desenvolvidos e testados.

Os resultados experimentais indicam que o aumento da quantidade de informação fornece melhores resultados em manter os agentes em suas respectivas TRs. No entanto, o custo associado devido à quantidade de migração não segue a mesma tendência, como pode ser observado através da comparação dos resultados obtidos pela abordagem de caminho direto para os alcançar o destino. Apesar de manter os agentes um tempo razoável dentro (ou perto) da TR, a abordagem caminho direto requer uma quantidade de migração do agente aproximadamente três vezes maior do que o baseado no destino. A abordagem avançada, que utiliza informações mais completas, permitem obter os melhores resultados mantendo o agente dentro TR por mais tempo. A quantidade de migrações realizada pelo agente na abordagem avançada é menor, resultado que é muito semelhante ao obtido pela abordagem baseada no destino.

Diferenças geográficas nos cenários simulados também foram exploradas. O modelo de movimento baseado em quadras possui uma distribuição dos nós melhor distribuída. O modelo de movimento utilizando o Google Maps apresenta uma distribuição dos nós mais concentrada. A conclusão obtida com essa variação geográfica causa impacto a densidade dos nós. A melhor distribuição dos nós no modelo baseado em quadras gerou resultados melhores dos algoritmos apresentados do que o modelo utilizando o Google Maps. A análise dos resultados obtidos nos dois cenários fornecem evidências de que os níveis mais elevados de inteli-

gência são mais robustos, mantendo o seu comportamento em ambos os cenários, enquanto o baseado no destino possui um comportamento mais inconstante.

Como propostas para continuidade deste trabalho, sugere-se combinar as três diferentes abordagens apresentadas neste trabalho, de modo que um agente possa adaptar o seu comportamento de acordo com a densidade dos nós, as condições ambientais e informação de contexto disponível. O uso de informações de contexto está também prevista para melhorar a comunicação entre os nós oportunistas, tais como utilização de algoritmos de inteligência artificial nos dados transmitidos e análise nas condições de tráfego, além do uso de informação geográfica. O tempo de contato entre os nós, tal como apresentado em (MARFIA et al., 2011), podem também ser adicionados para melhorar a proposta. O uso das técnicas propostas nas diferentes aplicações também é um campo que apresenta muitas oportunidades a serem exploradas, que enriquecerão a proposta acima referida do uso de informação semântica adicional.

## REFERÊNCIAS

ANDM, K. L.; LARSSON, T. A spatial qos requirements specification for v2v applications. In: *Intelligent Vehicles Symposium (IV), 2010 IEEE*. [S.l.: s.n.], 2010. p. 548 –553. ISSN 1931-0587.

BAI, F.; SADAGOPAN, N.; HELMY, A. Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. [S.l.: s.n.], 2003. v. 2, p. 825 – 835 vol.2. ISSN 0743-166X.

BURGESS, J.; GALLAGHER, B.; JENSEN, D.; LEVINE, B. N. Maxprop: Routing for vehicle-based disruption-tolerant networks. In: *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*. [S.l.: s.n.], 2006. p. 1 –11. ISSN 0743-166X.

BYCHKOVSKY, V.; CHEN, K.; GORACZKO, M.; HU, H.; HULL, B.; MIU, A.; SHIH, E.; ZHANG, Y.; BALAKRISHNAN, H.; MADDEN, S. The cartel mobile sensor computing system. In: *Proceedings of the 4th international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2006. (SenSys '06), p. 383–384. ISBN 1-59593-343-3. Disponível em: <<http://doi.acm.org/10.1145/1182807.1182866>>.

CHENG, P.-C.; LEE, K. C.; GERLA, M.; HÄRRI, J. Geodtn+nav: Geographic dtn routing with navigator prediction for urban vehicular environments. *Mob. Netw. Appl.*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 15, n. 1, p. 61–82, fev. 2010. ISSN 1383-469X. Disponível em: <<http://dx.doi.org/10.1007/s11036-009-0181-6>>.

CHESS, D.; GROSOFF, B.; PARRIS, C.; LEVINE, D.; PARRIS, C.; TSUDIK, G.; SWITZERLAND, R.; WATSON, T. J. *Itinerant Agents for Mobile Computing*. 1995.

CONTI, M.; CROWCROFT, J.; GIORDANO, S.; HUI, P.; NGUYEN, H. A.; PASSARELLA, A. Routing Issues in Opportunistic Networks Middleware for Network Eccentric and Mobile Applications. In: GARBINATO, B.; MIRANDA, H.; RODRIGUES, L. (Ed.). *Middleware for Network Eccentric and Mobile Applications*. Berlin, Germany: Springer, 2009. cap. 6, p. 121–147. ISBN 978-3-540-89706-4. Disponível em: <[http://dx.doi.org/10.1007/978-3-540-89707-1\\_6](http://dx.doi.org/10.1007/978-3-540-89707-1_6)>.

DRESSLER, F. A study of self-organization mechanisms in ad hoc and sensor networks. *Computer Communications*, v. 31, n. 13, p. 3018 – 3029, 2008. ISSN 0140-3664. <ce:title>Special Issue:Self-organization and self-management in communications as applied to autonomic networks</ce:title>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0140366408000959>>.

EICHLER, S.; SCHROTH, C.; EBERSPACHER, J. Car-to-car communication. VDE Verlag, 2006.

FALL, K. A delay-tolerant network architecture for challenged internets. In: *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2003. (SIGCOMM '03), p. 27–34. ISBN 1-58113-735-4.

FREITAS, E. P. de; HEIMFARTH, T.; COSTA, L. A. G. a.; FERREIRA, A. M.; PEREIRA, C. E.; WAGNER, F. R.; LARSSON, T. Analyzing different levels of geographic context awareness in agent ferrying over vanets. In: *Proceedings of the 2011 ACM Symposium on Applied Computing*. New York, NY, USA:

ACM, 2011. (SAC '11), p. 413–418. ISBN 978-1-4503-0113-8. Disponível em: <<http://doi.acm.org/10.1145/1982185.1982275>>.

FREITAS TALES HEIMFARTH, L. A. G. C. E. P. Exploring geographic context awareness for data dissemination on mobile ad hoc networks. *Submetido Adhoc Network Elsevier*, 2012.

GIORDANO, E.; FRANK, R.; PAU, G.; GERLA, M. Corner: a realistic urban propagation model for vanet. In: *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*. [S.l.: s.n.], 2010. p. 57–60.

GOOGLE. *Google Maps API*. 10 2012. Disponível em: <<https://developers.google.com/maps/>>.

GUISE, B.; PROCTOR, M. D. Expectations for presentation of engineering and scientific mobile platform information within a virtual globe geographic information systems. *J. Geographic Information System*, v. 3, n. 2, p. 120–127, 2011.

HUANG, C.-M.; LAN, K. chan; TSAI, C.-Z. A survey of opportunistic networks. In: *Advanced Information Networking and Applications - Workshops, 2008. AI-NAW 2008. 22nd International Conference on*. [S.l.: s.n.], 2008. p. 1672–1677.

HUHNS, M. N.; SINGH, M. P. (Ed.). *Readings in agents*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. ISBN 1-55860-495-2.

KUMAR, R.; DAVE, M. A comparative study of various routing protocols in vanet. *CoRR*, abs/1108.2094, 2011.

LANGE, D. B.; OSHIMA, M. Seven good reasons for mobile agents. *Commun. ACM*, ACM, New York, NY, USA, v. 42, n. 3, p. 88–89, mar. 1999. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/295685.298136>>.

LEE, U.; MAGISTRETTI, E.; GERLA, M.; BELLAVISTA, P.; CORRADI, A. Dissemination and harvesting of urban data using vehicular sensing platforms. *Vehicular Technology, IEEE Transactions on*, v. 58, n. 2, p. 882–901, feb. 2009. ISSN 0018-9545.

LEONTIADIS, I.; MASCOLO, C. Geopps: Geographical opportunistic routing for vehicular networks. In: *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*. [S.l.: s.n.], 2007. p. 1–6.

LESSMANN, J.; HEIMFARTH, T.; JANACIK, P. Shox: An easy to use simulation platform for wireless networks. In: *Computer Modeling and Simulation, 2008. UKSIM 2008. Tenth International Conference on*. [S.l.: s.n.], 2008. p. 410–415.

LEVIS, P.; CULLER, D. Mate: a tiny virtual machine for sensor networks. *SI-GOPS Oper. Syst. Rev.*, ACM, New York, NY, USA, v. 36, n. 5, p. 85–95, out. 2002. ISSN 0163-5980. Disponível em: <<http://doi.acm.org/10.1145/635508.605407>>.

LOULLOUDES, N.; PALLIS, G.; DIKAIAKOS, M. D. The dynamics of vehicular networks in urban environments. *CoRR*, abs/1007.4106, 2010.

MARFIA, G.; ROCCETTI, M.; PALAZZI, C. E.; AMOROSO, A. Efficient vehicle-to-pedestrian exchange of medical data: an empirical model with preliminary results. In: *Proceedings of the First ACM MobiHoc Workshop on Pervasive Wireless Healthcare*. New York, NY, USA: ACM, 2011. (MobileHealth '11), p. 3:1–3:4. ISBN 978-1-4503-0780-2. Disponível em: <<http://doi.acm.org/10.1145/2007036.2007040>>.

MEYER, H.; HUMMEL, K. A. A geo-location based opportunistic data dissemination approach for manets. In: *Proceedings of the 4th*

*ACM workshop on Challenged networks*. New York, NY, USA: ACM, 2009. (CHANTS '09), p. 1–8. ISBN 978-1-60558-741-7. Disponível em: <<http://doi.acm.org/10.1145/1614222.1614224>>.

MILLER, C. C. A beast in the field: The google maps mashup as gis/2. *Cartographica The International Journal for Geographic Information and Geovisualization*, UT Press, v. 41, n. 3, p. 187–199, 2006.

NIAZI, M.; HUSSAIN, A. Agent-based tools for modeling and simulation of self-organization in peer-to-peer, ad hoc, and other complex networks. *Communications Magazine, IEEE*, v. 47, n. 3, p. 166–173, march 2009. ISSN 0163-6804.

NICHOLS, R. A.; HAMMONS, A. R. Performance of dtn-based free-space optical networks with mobility. In: *Military Communications Conference, 2007. MILCOM 2007. IEEE*. [S.l.: s.n.], 2007. p. 1–6.

NICULESCU, D.; NATH, B. Localized positioning in ad hoc networks. *Ad Hoc Networks*, v. 1, n. 2–3, p. 247 – 259, 2003. ISSN 1570-8705. <ce:title>Sensor Network Protocols and Applications</ce:title>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1570870503000052>>.

PELUSI, L.; PASSARELLA, A.; CONTI, M. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE*, v. 44, n. 11, p. 134–141, november 2006. ISSN 0163-6804.

PIORKOWSKI, M.; SARAFIJANOVIC-DJUKIC, N.; GROSSGLAUSER, M. A parsimonious model of mobile partitioned networks with clustering. In: *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*. [S.l.: s.n.], 2009. p. 1–10.

PIRAN, M. J.; MURTHY, G. R.; BABU, G. P. Vehicular ad hoc and sensor networks; principles and challenges. *CoRR*, abs/1108.2776, 2011.

REZENDE, C. G.; ROCHA, B. P. S.; LOUREIRO, A. A. F. Publish/subscribe architecture for mobile ad hoc networks. In: *Proceedings of the 2008 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2008. (SAC '08), p. 1913–1917. ISBN 978-1-59593-753-7. Disponível em: <<http://doi.acm.org/10.1145/1363686.1364149>>.

SARKAR, S. K.; BASAVARAJU, T. G.; PUTTAMADAPPA, C. *Ad Hoc Mobile Wireless Networks: Principles, Protocols and Applications*. 1st. ed. Boston, MA, USA: Auerbach Publications, 2007. ISBN 1420062212, 9781420062212.

SHEHORY, O.; SYCARA, K.; CHALASANI, P.; JHA, S. Agent cloning: An approach to agent mobility and resource allocation. *IEEE Communications*, 1998.

SHU, L.; ZHANG, Y.; YANG, L.; WANG, Y.; HAUSWIRTH, M.; XIONG, N. Tpgf: geographic routing in wireless multimedia sensor networks. *Telecommunication Systems*, Springer Netherlands, v. 44, p. 79–95, 2010. ISSN 1018-4864. 10.1007/s11235-009-9227-0. Disponível em: <<http://dx.doi.org/10.1007/s11235-009-9227-0>>.

SPYROPOULOS, T.; RAIS, R. N.; TURLETTI, T.; OBRACZKA, K.; VASILAKOS, A. Routing for disruption tolerant networks: taxonomy and design. *Wirel. Netw.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 16, n. 8, p. 2349–2370, nov. 2010. ISSN 1022-0038. Disponível em: <<http://dx.doi.org/10.1007/s11276-010-0276-9>>.

SUN, J.-Z.; SAUVOLA, J. Mobility and mobility management: a conceptual framework. In: *Networks, 2002. ICON 2002. 10th IEEE International Conference on*. [S.l.: s.n.], 2002. p. 205 – 210.

TAYLOR, C. An overview of geographic routing. In: *Seminar Technische Informatik*. [S.l.: s.n.], 2006.

TEI, K.; FUKAZAWA, Y.; HONIDEN, S.; YOSHIOKA, N. Geographically bound mobile agent in manet. *Mobile and Ubiquitous Systems, Annual International Conference on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 516–518, 2005.

WOLF, W. *Computers as Components, Second Edition: Principles of Embedded Computing System Design*. 2nd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008. ISBN 0123743974, 9780123743978.

WOOLRIDGE, M. *Introduction to Multiagent Systems*. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN 047149691X.

WU, H.; FUJIMOTO, R.; RILEY, G.; HUNTER, M. Spatial propagation of information in vehicular networks. *Vehicular Technology, IEEE Transactions on*, v. 58, n. 1, p. 420 –431, jan. 2009. ISSN 0018-9545.

WU, H.; PALEKAR, M.; FUJIMOTO, R.; LEE, J.; KO, J.; GUENSLER, R.; HUNTER, M. Vehicular networks in urban transportation systems. In: *Proceedings of the 2005 national conference on Digital government research*. Digital Government Society of North America, 2005. (dg.o '05), p. 9–10. Disponível em: <<http://dl.acm.org/citation.cfm?id=1065226.1065231>>.

ZAFOUNE, Y.; KANAWATI, R.; MOKHTARI, A. Mobile agents localization in ad hoc networks: a comparative study of centralized and distributed approaches.

In: *Information and Communications Technology, 2007. ICICT 2007. ITI 5th International Conference on*. [S.l.: s.n.], 2007. p. 269–275.

ZHAO, W.; AMMAR, M.; ZEGURA, E. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2004. (MobiHoc '04), p. 187–198. ISBN 1-58113-849-0. Disponível em: <<http://doi.acm.org/10.1145/989459.989483>>.