

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E  
INFORMÁTICA INDUSTRIAL

BRUNO VIANA COUTINHO

**G-NET: UM PROTOCOLO DE ROTEAMENTO BASEADO EM  
ALGORITMOS GENÉTICOS PARA REDES AD HOC VEICULARES**

DISSERTAÇÃO

CURITIBA

2015

**BRUNO VIANA COUTINHO**

**G-NET: UM PROTOCOLO DE ROTEAMENTO BASEADO EM  
ALGORITMOS GENÉTICOS PARA REDES AD HOC VEICULARES**

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Mestre em Ciências” – Área de Concentração: Telemática.

Orientador: Emilio Carlos Gomes Wille

Co-orientador: Hermes Irineu Del Monego

**CURITIBA**

**2015**

C871g  
2014 Coutinho, Bruno Viana  
G-NET : um protocolo de roteamento baseado em algoritmos  
genéticos para redes ad hoc veiculares / Bruno Viana  
Coutinho.-- 2014.  
101 f. : il. ; 30 cm

Texto em português com resumo em inglês  
Dissertação (Mestrado) - Universidade Tecnológica  
Federal do Paraná. Programa de Pós-Graduação em Engenharia  
Elétrica e Informática Industrial. Área de concentração:  
Telmática, Curitiba, 2014  
Bibliografia: f. 95-101

1. Redes veiculares ad hoc (Redes de computadores).  
2. Redes ad hoc (Redes de computadores). 3. Algoritmos  
genéticos. 4. Engenharia elétrica - Dissertações.  
I.Wille, Emilio Carlos Gomes. II.Del Monego, Hermes  
Irineu. III.Universidade Tecnológica Federal do Paraná

Título da Dissertação Nº. 671

## **G-NET: Um Protocolo de Roteamento Baseado em Algoritmos Genéticos para Redes AD HOC Veiculares.**

por

**Bruno Viana Coutinho.**

**Orientador:** Prof. Dr. Emilio Carlos Gomes Wille.

**Coorientador:** Prof. Dr. Hermes Irineu Del Monego.

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM CIÊNCIAS – Área de Concentração: Telecomunicações e Redes do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI – da Universidade Tecnológica Federal do Paraná – UTFPR, às 14h00 do dia 18 de novembro de 2014. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores doutores:

---

Prof. Dr. Emilio Carlos Gomes Wille  
(Presidente – UTFPR)

---

Prof. Dr. Marcelo Eduardo Pellenz  
(PUC-PR)

---

Prof. Dr. Augusto Foronda  
(UTFPR)

Visto da coordenação:

---

**Prof. Emilio Carlos Gomes Wille, Dr.**  
(Coordenador do CPGEI)

A meus pais Rui e Joselia. Que sempre me sustentaram até o início da carreira profissional, me doutrinaram para o ambiente pessoal e profissional e deram suporte nos momentos mais difíceis da minha vida.

## **AGRADECIMENTOS**

Primeiramente a Deus que proporcionou minha família e educação para chegar até aqui. A minha esposa por acreditar sempre nas minhas conquistas e pela paciência em me acompanhar nos dias de trabalho que se prolongavam por noites e madrugadas.

Ao professor Emilio por acreditar em meu potencial. Pela experiência que demonstrou no esclarecimento de dúvidas durante as discussões sobre as tarefas executadas e principalmente por respeitar os momentos de dificuldades pessoais que ocorreram durante o decorrer do curso.

Ao professor Hermes por apresentar as ferramentas de simulação, pessoas e grupos de estudo que foram fundamentais para a execução dos experimentos práticos.

Ao IFTO, ao CNPq e a CPGEI/UTFPR pelos recursos financeiros e materiais disponibilizados que foram de grande importância durante o mestrado.

A todos os companheiros da coordenação de informática do IFTO, campus Palmas-TO, que acreditaram e incentivaram desde o início, indicando a área de conhecimento e auxiliando com apoio nas cargas horárias durante todo o curso. Especialmente aos professores Helder Cleber, Cláudio Monteiro, Manoel Campos, Mauro de Boni, Vinícius Costa, Carlos Tolentino, Francisco das Chagas, ao ex-coordenador Fagno Fonseca e o atual, Francisco Willians.

Aos professores Walter Godoy Júnior (Que Deus o tenha em paz), Hugo Vieira, Leandro Magatão e Sérgio Pichorim pela forma como conduziram suas atividades acadêmicas e por todos os ensinamentos de vida profissional e pessoal.

A todos da UTFPR que me acompanharam e me ajudaram nas tarefas mais difíceis. Pelo apoio moral nos momentos de maior pressão psicológica, quando deram forças para continuar e sem os quais não conseguiria chegar até o final deste trabalho. Principalmente Euclides Farias, Fabian Malheiros, Rodrigo Silva, Eden Dosciatti, Joilson Alves, Danillo Belmonte, Omero Bertol, Charles Gentilin, Acácio Zimbico, Charles Fung, Antonio Gortan, Kleber Nabas, Marcos Talau, Vilmar Candido, Carlos Micaloski e Aratã Saraiva.

Não posso deixar de fazer um agradecimento especial as pessoas e amigos que foram responsáveis diretos na indicação e finalização deste trabalho. A Giovanna Garcia e Adriano Ronszcka pela grande contribuição nas últimas tarefas e Augusto Foronda, que me apresentou este curso.

“Comece fazendo o que é necessário, depois o que é possível, e de repente você estará fazendo o impossível.”  
(S. Francisco de Assis)

## RESUMO

COUTINHO, Bruno. G-NET: UM PROTOCOLO DE ROTEAMENTO BASEADO EM ALGORITMOS GENÉTICOS PARA REDES AD HOC VEICULARES. 101 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2015.

As redes *ad hoc* são aquelas em que os próprios nós funcionam como roteadores, clientes e servidores ao mesmo tempo. As redes *ad hoc* móveis são conhecidas como MANETs (*Mobile Ad hoc Network*) e as VANETs (*Vehicle Ad hoc Network*) são um tipo de MANET específica para redes veiculares, onde os nós das redes são os próprios veículos equipados com antenas *wireless* para que possam trocar informações entre si. Na comunicação entre veículos acontecem problemas que as MANETs não estão habilitadas a solucionar. Neste contexto, enquanto nas redes móveis comuns as pessoas se deslocam em pequenos espaços com seus computadores e normalmente existem pontos fixos que facilitam a comunicação constante, nas VANETs a alta mobilidade dos veículos causa diversos problemas como mudanças rápidas e frequentes na topologia da rede, veículos heterogêneos, nós e caminhos disjuntos, obstáculos e exigência de entrega rápida para os quais as MANETs não foram preparadas. O *Dynamic Source Route* (DSR) é um protocolo de roteamento MANET reativo, ou seja, só descobre rotas entre um nó de origem e um destino quando necessita enviar dados. Por ser “*source route*”, o protocolo DSR armazena e utiliza vetores de rota (rotas completas) da origem até o destino, diferente de protocolos *hop-by-hop* que conhecem apenas o próximo salto. Este trabalho apresenta uma proposta de protocolo de roteamento, baseado no DSR, com o uso de algoritmo genético (AG) para atender as exigências das redes VANET. O DSR foi adotado pelo fato de ser um protocolo “*source route*” que possibilita utilizar técnicas de algoritmo genético, onde os nós podem ser os genes e as rotas são os cromossomos. O novo protocolo criado é chamado de G-Net e possui uma nova forma de funcionamento, modificando totalmente a etapa de manutenção de rotas do DSR. Ele tem o objetivo de atualizar e otimizar as rotas periodicamente de tal forma que as rotas com menor latência sejam escolhidas para transmissão de dados. O trabalho investiga o comportamento do G-Net com relação a taxa média de entrega de dados e sobrecarga de roteamento em relação ao DSR e o AODV (*Ad hoc On-demand Distance Vector*). A análise destes protocolos de roteamento foi realizada com o gerador de mobilidade VanetMobiSim, para simulação mais realista de movimentos veiculares em ambiente urbano, além de software de simulação de redes ns-3. Experimentos foram realizados com diferentes números de veículos e os resultados mostram que apesar de aumentar a sobrecarga de roteamento em relação ao DSR, o G-Net continua com a sobrecarga inferior ao AODV e apresenta taxa média de entrega de dados superior aos outros dois protocolos nos cenários analisados.

**Palavras-chave:** VANET, ns-3, VanetMobiSim, Protocolo DSR, Algoritmos Genéticos.



## ABSTRACT

COUTINHO, Bruno. G-NET: A ROUTING PROTOCOL BASED ON GENETIC ALGORITHMS FOR VEHICULAR AD HOC NETWORKS. 101 f. Dissertação – Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2015.

Ad hoc networks are those the nodes themselves work like routers, clients and servers simultaneously. Mobile ad hoc networks are known as MANETs and VANETs are a specific MANET to vehicle networks that the nodes are vehicles with wireless radios to exchange information. On communication between vehicles happen problems that MANETs are not qualified to solve. Thus, on simple mobile networks the people move in small spaces with their computers and usually have fixed points that facilitate the constant communication. But in VANETs, the high mobility of vehicles causes many problems as rapid and frequent changes in network topology, heterogeneous vehicles, disjoint nodes and links, obstacles and requiring fast delivery that MANETs were not prepared to solve. The DSR is a reactive routing protocol for MANET that only discovers routes when it needs to send data. The DSR is source route, for this it stores and uses route vectors (complete routes) from source to destination, unlike hop-by-hop protocols that know only the next hop. This document proposes a new routing protocol based on DSR, using genetic algorithm to meet the requirements of VANET networks. The DSR was adopted because it is a source route protocol that enables using genetic algorithm techniques, which nodes can be genes and chromosomes are routes. The new protocol is called G-Net and has a new mode of operation, modifying completely the maintaining routes of DSR. The G-Net aims to update and optimize routes periodically so that the routes with lower latency are chosen for data transmission. The dissertation investigates the behavior about average delivery ratio and routing overhead of the G-Net compared to DSR and AODV. The analysis of these routing protocols was performed with the generator mobility VanetMobiSim for more realistic simulation of vehicle movements in an urban environment, as well as network simulation ns-3 software. Experiments were conducted with different number of vehicle and the results show that despite increasing the routing overhead with respect to DSR, G-net continues with lower overhead in relation to AODV and has an average delivery ratio higher than the two other protocols in analyzed scenarios.

**Keywords:** VANET, ns-3, VanetMobiSim, DSR Protocol, Genetic Algorithm.

## LISTA DE FIGURAS

FIGURA 1	–	Categorias VANET quanto à arquitetura ou estrutura tecnológica. ....	24
FIGURA 2	–	Classificação dos protocolos de roteamento baseados em topologia ....	34
FIGURA 3	–	Classificação dos protocolos de roteamento geográficos .....	39
FIGURA 4	–	Classificação dos protocolos de roteamento pelo tipo de propagação dos pacotes de roteamento .....	42
FIGURA 5	–	Comparação entre os protocolos de roteamento VANET pela forma de propagação .....	42
FIGURA 6	–	Mecanismo de proteção de loop .....	58
FIGURA 7	–	Exemplos de cruzamento .....	63
FIGURA 8	–	Exemplo de mutação e geração da população final .....	64
FIGURA 9	–	Máquina de estados de funcionamento do G-Net .....	64
FIGURA 10	–	Diagrama de funcionamento do G-Net .....	66
FIGURA 11	–	Exemplo de funcionamento do G-Net - pacotes .....	67
FIGURA 12	–	Exemplo de funcionamento do G-Net - <i>cache</i> de rotas .....	68
FIGURA 13	–	Cálculo da latência .....	71
FIGURA 14	–	Exemplo de conversão de rotas .....	72
FIGURA 15	–	Imagem de movimento de 100 veículos simulados no VanetMobisim. ....	73
FIGURA 16	–	Exemplo de “ <i>trace</i> de mobilidade” do VanetMobiSim mostrando dados sobre o movimento de veículos. ....	73
FIGURA 17	–	Representação gráfica do tráfego de dados da simulação .....	77
FIGURA 18	–	Cenário para análise de caminhos percorridos de acordo com a latência das rotas .....	78
FIGURA 19	–	Exemplo de caminho percorrido no instante 20 segundos .....	79
FIGURA 20	–	Exemplo de caminhos percorridos no instante 40 segundos .....	80
FIGURA 21	–	Exemplo de caminhos percorridos no instante 60 segundos .....	80
FIGURA 22	–	Exemplo de caminhos percorridos no instante 80 segundos .....	81
FIGURA 23	–	Gráfico de resultados ADR do cenário 1 com tráfego CBR ininterrupto ...	84
FIGURA 24	–	Gráfico de resultados RO do cenário 1 com tráfego CBR ininterrupto ....	85
FIGURA 25	–	Gráfico de resultados ADR do cenário 2 com tráfego CBR ininterrupto ...	86
FIGURA 26	–	Gráfico de resultados RO do cenário 2 com tráfego CBR ininterrupto ....	87
FIGURA 27	–	Gráfico de resultados ADR do cenário 1 com pausa no tráfego de dados CBR .....	88
FIGURA 28	–	Gráfico de resultados RO do cenário 1 com pausa no tráfego de dados CBR	89
FIGURA 29	–	Gráfico de resultados ADR do cenário 2 com pausa no tráfego de dados CBR .....	90
FIGURA 30	–	Gráfico de resultados RO do cenário 2 com pausa no tráfego de dados CBR	90

## LISTA DE TABELAS

TABELA 1	– Comparação entre WAVE e Wi-Fi .....	31
TABELA 2	– Comparação entre os geradores de mobilidade veicular .....	54
TABELA 3	– Resumo dos parâmetros utilizados nas simulações do cenário 1 .....	74
TABELA 4	– Resumo dos parâmetros utilizados nas simulações do cenário 2 .....	75
TABELA 5	– Exemplo do resultado de caminhos percorridos .....	79

## LISTA DE SIGLAS

3G	<i>Cellular third generation</i>
ABR	<i>Associativity-Based Routing</i>
ACTP	<i>Application-Controlled Transport Protocol</i>
ADR	<i>Average Delivery Ratio</i>
AE	<i>Algoritmos Evolucionários</i>
AG	<i>Algoritmo Genético</i>
AODV	<i>Ad hoc On-demand Distance Vector</i>
API	<i>Application Programming Interface</i>
ATCP	<i>Ad Hoc TCP</i>
ATP	<i>Ad Hoc Transport Protocol</i>
BER	<i>Bit Error Rate</i>
CANU	<i>Communication in Ad Hoc Networks for Ubiquitous Computing</i>
CAR	<i>Connectivity-Aware Routing</i>
CBF	<i>Contention-Based Forwarding</i>
CBR	<i>Taxa de bits constante (do original Constant Bit Rate)</i>
CGSR	<i>Clusterhead Gateway Switch Routing</i>
CNPq	<i>Conselho Nacional de Desenvolvimento Científico e Tecnológico</i>
CPGEI	<i>Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial</i>
CRN	<i>Congested Road Notification</i>
DCF	<i>Distributed Coordination Function</i>
D-Greedy	<i>Delay-bounded Greedy Forwarding</i>
DIR	<i>Diagonal-Intersection-Based Routing Protocol</i>
DM	<i>Downtown Model</i>
D-MinCost	<i>Delay-bounded Min-Cost Forwarding</i>
DSDV	<i>Destination-Sequenced Distance-Vector</i>
DSR	<i>Dynamic Source Route</i>
DSRC	<i>Dedicated Short Range Communications</i>
DTN	<i>Delay Tolerant Network</i>
DYMO	<i>DYnamic MANET On Demand</i>
EDCA	<i>Enhanced Distributed Channel Access</i>
FCC	<i>Federal Communication Commission</i>
FreeSim	<i>Freeway Simulator</i>
FSLs	<i>Fuzzy Sighted Link State</i>
FSR	<i>Fisheye State Routing</i>
GDF	<i>Geographical Data Files</i>
GeOpps	<i>Geographical Opportunistic routing</i>
G-Net	<i>Genetic Network Protocol</i>
GPCR	<i>Greedy Perimeter Coordinator Routing</i>
GPL	<i>General Public License</i>
GPS	<i>Global Positioning System</i>
GPSR	<i>Greedy Perimeter Stateless Routing</i>

GSR	<i>Geographic Source Routing</i>
GTNetS	<i>Georgia TechNetwork Simulator</i>
GUI	<i>Graphical User Interface</i>
HSR	<i>Hierarchical State Routing</i>
HWMP	<i>Hybrid Wireless Mesh Protocol</i>
IDM_IM	<i>Intelligent Driving Model with Intersection Management</i>
IDM_LC	<i>Intelligent Driving Model with Lane Changing</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IFTO	<i>Instituto Federal de Educação, Ciência e Tecnologia do Tocantins</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Standards Organization</i>
ITS	<i>Intelligent Transport System</i>
IVC	<i>Inter-Vehicle Communication</i>
JIST	<i>Java in Simulation Time</i>
LAN	<i>Local Area Network</i>
LANMAR	<i>Landmark Ad Hoc Routing Protocol</i>
LMR	<i>Lightweight Mobile Routing</i>
MAC	<i>Media Access Control</i>
MANET	<i>Mobile Ad hoc Network</i>
MM	<i>Manhattan Model</i>
MOBIL	<i>Minimizing overall braking induced by lane change</i>
MOPR	<i>MOvement Prediction-based Routing</i>
MOVE	<i>Motion Vector</i>
MOVE	<i>MObility model generator for VEhicular networks</i>
NCTUns	<i>National Chiao Tung University Network Simulator</i>
ns-2	<i>Network Simulator - version 2</i>
ns-3	<i>Network Simulator - version 3</i>
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>
OLSR	<i>Optimized Link State Routing Protocol</i>
OORP	<i>OrderOne Routing Protocol</i>
OpenGL	<i>Open Graphics Library</i>
OSI	<i>Open Systems Interconnection</i>
OTcl	<i>Object oriented extension of Tcl</i>
PAN	<i>Parking Availability Notification</i>
PGB	<i>Prefered Group Broadcasting</i>
PHY	<i>Physic layer</i>
PMP	<i>Point Multipoint</i>
POD	<i>Problemas de Otimização Dinâmica</i>
PRA	<i>Problema de Roteamento Automático</i>
RB-UDP	<i>Reliable Blast-UDP</i>
RO	<i>Sobrecarga de roteamento (do original <i>Routing Overhead</i>)</i>
ROMSGP	<i>Receive on Most Stable Group-Path</i>
RTT	<i>Round Trip Times</i>
RVC	<i>Road-vehicle Communications</i>
SADV	<i>Static-Node Assisted Adaptive Routing Protocol</i>

SM	<i>Simple Model</i>
Split-TCP	<i>Split - Transmission Control Protocol</i>
SSR	<i>Signal Stability-Based Adaptive Routing</i>
STAR	<i>Spatial and Traffic Aware Routing</i>
STRAW	<i>STreet RAndom Waypoint</i>
SUMO	<i>Simulation of Urban MObility</i>
SVA	<i>Slow/Stop Advisor Vehicle</i>
SWANS	<i>Scalable Wireless Ad hoc Network Simulator</i>
TBRPF	<i>Topology Broadcast Based on Reverse Path Forwarding</i>
Tcl	<i>Tool Command Language</i>
TCP	<i>Transmission Control Protocol</i>
TCP-Bus	<i>Transmission Control Protocol - Bus</i>
TCP-ELFN	<i>TCP-Explicit Link Failure Notification</i>
TCP-F	<i>TCP-Failure</i>
TIGER	<i>Topologically Integrated Geographic Encoding and Referencing Geodatabases</i>
TO-GO	<i>TOpology-assist Geo-Opportunistic Routing</i>
TORA	<i>Temporally Ordered Routing Algorithm</i>
UDP	<i>User Datagram Protocol</i>
USB	<i>Universal Serial Bus</i>
UTFPR	<i>Universidade Tecnológica Federal do Paraná</i>
V2I	<i>Vehicle-to-Infrastrucuture</i>
V2IC	<i>Vehicle-to-Infrastrucuture Communication</i>
V2V	<i>Vehicle-to-Vehicle</i>
V2VC	<i>Vehicle-to-Vehicle Communication</i>
VADD	<i>Vehicle-Assisted Data Delivery</i>
VANET	<i>Vehicle Ad hoc Network</i>
VINT	<i>Virtual InterNetwork Testbed</i>
VTP	<i>Vehicular Transport Protocol</i>
WAVE	<i>Wireless Access in Vehicular Environments</i>
Wi-Fi	<i>Wireless Fidelity</i>
WLAN	<i>Wireless Local Area Network</i>
WRP	<i>Wireless Routing Protocol</i>
XML	<i>eXtensible Markup Language</i>
ZRP	<i>Zone Routing Protocol</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>16</b>
1.1 OBJETIVOS .....	17
1.2 DESCRIÇÃO DOS CAPÍTULOS .....	18
<b>2 REDES VEICULARES</b> .....	<b>19</b>
2.1 FUNCIONAMENTO GERAL .....	19
2.2 CLASSIFICAÇÃO DAS REDES VEICULARES .....	20
2.2.1 Quanto às aplicações de transporte inteligente .....	20
2.2.1.1 Aplicações de entretenimento .....	21
2.2.1.2 Aplicações de segurança rodoviária .....	21
2.2.1.3 Aplicações de assistência .....	22
2.2.2 Quanto à arquitetura ou estrutura tecnológica .....	23
2.2.2.1 Arquitetura pura WLAN/Celular ou infraestruturada (V2I) .....	24
2.2.2.2 Arquitetura pura ad hoc ou V2V .....	25
2.2.2.3 Arquitetura híbrida .....	25
2.2.3 Quanto ao ambiente (ou cenário) de atuação .....	26
2.2.3.1 Ambiente de rodovia .....	26
2.2.3.2 Ambiente urbano .....	26
2.3 PRINCIPAIS PROBLEMAS EXISTENTES NAS VANETS .....	27
2.3.1 Propostas de soluções para os problemas em VANETS .....	27
<b>3 NORMAS E PADRÕES PARA REDES SEM FIO E VANETS</b> .....	<b>30</b>
3.1 ASPECTOS TÉCNICOS DE TRANSMISSÃO (CAMADA FÍSICA E ACESSO AO MEIO) .....	30
3.2 PROTOCOLOS DE ROTEAMENTO (CAMADA DE REDE) .....	33
3.2.1 Classificação dos protocolos de roteamento pela área de abrangência (ou forma de controle) .....	33
3.2.1.1 Protocolo de roteamento baseado em topologia .....	34
3.2.1.1.1 Roteamento plano ( <i>Flat routing</i> ) .....	34
3.2.1.1.2 Roteamento hierárquico .....	37
3.2.1.2 Roteamento geográfico .....	38
3.2.1.2.1 DTN .....	39
3.2.1.2.2 Não-DTN .....	40
3.2.1.2.3 Híbrido (DTN/Não-DTN) .....	41
3.2.2 Classificação dos protocolos de roteamento pelo tipo de propagação dos pacotes de roteamento .....	41
3.2.2.1 Protocolo de Roteamento <i>Unicast</i> .....	42
3.2.2.1.1 Protocolo de Roteamento de Mínimo Atraso ( <i>Min-Delay Routing Protocol</i> ) ....	43
3.2.2.1.2 Protocolo de Roteamento de Atraso Limitado ( <i>Delay-Bounded Routing Protocol</i> )	43
3.2.2.2 Protocolo de Roteamento Multicast e Geocast .....	44
3.2.2.3 Protocolos de Roteamento com Base em Difusão ( <i>Broadcast Based Routing Protocols</i> ) .....	44

3.3	CAMADA DE TRANSPORTE .....	44
<b>4</b>	<b>FERRAMENTAS DE SIMULAÇÃO PARA VANETS .....</b>	<b>46</b>
4.1	SIMULADORES DE REDE .....	46
4.2	GERADORES DE MOBILIDADE .....	48
4.3	SIMULADORES VANET .....	51
4.4	FERRAMENTAS DE SIMULAÇÃO UTILIZADAS .....	53
<b>5</b>	<b>G-NET: ROTEAMENTO BASEADO EM ALGORITMOS GENÉTICOS .....</b>	<b>56</b>
5.1	FUNCIONAMENTO DO PROTOCOLO DSR .....	57
5.2	ALGORITMO GENÉTICO PARA OTIMIZAÇÃO DE ROTAS .....	58
5.2.1	Codificação do cromossomo .....	60
5.2.2	Função de <i>Fitness</i> .....	61
5.2.3	Seleção de indivíduos .....	61
5.2.4	Cruzamento ( <i>Crossover</i> ) .....	62
5.2.5	Mutação .....	63
5.2.6	Reparação de rotas .....	63
5.3	FUNCIONAMENTO DO PROTOCOLO G-NET .....	64
<b>6</b>	<b>IMPLEMENTAÇÃO E RESULTADOS DE SIMULAÇÃO .....</b>	<b>70</b>
6.1	IMPLEMENTAÇÃO DO G-NET NO NS-3 .....	70
6.2	TRACES DE MOBILIDADE .....	72
6.3	CONFIGURAÇÃO DE CENÁRIOS E PROTOCOLOS .....	74
6.4	PROTOCOLOS DE ROTEAMENTO ANALISADOS .....	75
6.4.1	AODV .....	75
6.4.2	DSR .....	76
6.5	TRÁFEGO DE DADOS .....	77
6.6	ANÁLISE DE CAMINHOS PERCORRIDOS .....	78
6.6.1	Análise de caminhos percorridos de acordo com a latência .....	78
6.6.2	Resultados da análise de caminhos percorridos .....	79
6.7	MÉTRICAS ANALISADAS .....	81
6.7.1	Taxa média de entrega de dados .....	82
6.7.2	Sobrecarga de roteamento .....	82
6.7.3	Resultados das métricas analisadas .....	83
6.7.3.1	Resultados com tráfego CBR ininterrupto .....	83
6.7.3.1.1	Resultados Cenário 1 com tráfego CBR ininterrupto .....	83
6.7.3.1.2	Resultados Cenário 2 com tráfego CBR ininterrupto .....	85
6.7.3.2	Resultados com pausa no tráfego CBR .....	87
6.7.3.2.1	Resultados Cenário 1 com pausa no tráfego de dados CBR .....	87
6.7.3.2.2	Resultados Cenário 2 com pausa no tráfego de dados CBR .....	89
<b>7</b>	<b>CONCLUSÃO .....</b>	<b>92</b>
7.1	TRABALHOS FUTUROS .....	93
	<b>REFERÊNCIAS .....</b>	<b>95</b>



## 1 INTRODUÇÃO

Com o aumento da frota de veículos nas ruas, em especial nas grandes cidades, a população já está acostumada a conviver com enormes engarrafamentos. Tentando amenizar os impactos desta realidade, a comunidade acadêmica e industrial está concentrando seus esforços para elaborar soluções tecnológicas que facilitem a vida do condutor com tantos veículos a sua volta.

Dentre as soluções tecnológicas, o uso de *Bluetooth* ou USB (*Universal Serial Bus*), para que o motorista não tire as mãos do volante enquanto fala no celular, é uma realidade nos veículos atuais.

Outra tecnologia bastante utilizada nos veículos é o aparelho de GPS (*Global Positioning System*), com mapas que facilitam a vida do condutor, permitindo-lhe localizar postos de combustíveis, áreas de serviço, caminhos mais curtos, posição de semáforos e até mesmo de radares. Porém, estes equipamentos não atualizam informações em tempo real (BITAGHSIR, 2009).

Além das tecnologias citadas, o acesso a Internet também se torna uma necessidade crescente para a sociedade atual. Isso é confirmado por Li e Wang (2007) quando afirmam que além das aplicações específicas de trânsito, vislumbra-se o acesso à Internet em qualquer lugar e a qualquer instante. O autor Costa (2010) reforça ainda que: "Existem três grandes classes de aplicações dentro da área dos Sistemas de Transportes Inteligentes (ITS): entretenimento, segurança rodoviária e assistência".

Por esses motivos a comunidade acadêmica está concentrando seus esforços em busca de tecnologias móveis que permitam a troca de informações entre os veículos automotores de forma eficaz. Desta forma, seria possível receber e passar informações para outros automóveis alertando-os sobre acidentes, congestionamentos e vias tranquilas por onde passaram, permitindo aos demais condutores, escolher caminhos alternativos através das mensagens captadas.

As MANETs são uma das principais tecnologias adotadas para realizar esses estudos

por apresentarem bons resultados no envio e recebimento de mensagens com antenas em movimento, como é o caso dos veículos. Da união desta tecnologia com meios automotivos surgiu o termo VANET, ou seja, rede *ad hoc* veicular, que consiste de uma MANET com veículos automotores. A tecnologia *ad hoc* permite que o equipamento envie e receba informações em ambiente não estruturado, ou seja, sem necessidade de antenas fixas para a comunicação entre os veículos.

No decorrer deste trabalho serão apresentadas as principais características sobre as VANETs, detalhando desde a sua importância e funcionamento geral até a apresentação de problemas e soluções de outros autores. A principal contribuição é a descrição da proposta de protocolo de roteamento G-Net (*Genetic Network Protocol*), baseado no DSR (*Dynamic Source Route*) com inclusão de algoritmo genético, criando um novo processo de atualização e manutenção de rotas. Além disso, será explicado como realizar experimentos VANET com o simulador de rede ns-3 (NS-3 RASTREAMENTO, 2013) e o VanetMobiSim (HARRI; FIORE, 2006) em ambiente urbano, que representa uma novidade e um desafio para o mundo acadêmico e científico, incentivando o uso destas ferramentas em simulações VANET.

## 1.1 OBJETIVOS

Esta dissertação propõe e analisa o funcionamento de um protocolo de roteamento VANET chamado de G-Net, que consiste na completa substituição da etapa de manutenção de rotas do protocolo DSR por um procedimento baseado em algoritmo genético. Os objetivos específicos podem ser detalhados conforme a seguir:

- Propor e validar um protocolo de roteamento baseado no DSR com algoritmo genético, de tal forma que o mesmo se apresente mais eficiente, tanto na escolha de rotas com melhores latência quanto na taxa média de entrega de dados.

- Avaliar o comportamento do novo protocolo em um ambiente tipicamente urbano de acordo com métricas pré-estabelecidas, comparando seu desempenho em relação ao DSR para melhor entendimento da contribuição científica e com o protocolo AODV para facilitar a análise com outros trabalhos relacionados.

- Implementar o novo protocolo G-Net em ferramentas de simulação mais atuais como o ns-3 e o VanetMobiSim, encorajando estudantes e pesquisadores da área de Redes de Computadores a utilizarem estes tipos de ferramentas. Em particular o VanetMobiSim é capaz de realizar experimentos com situações mais realistas de um ambiente urbano.

## 1.2 DESCRIÇÃO DOS CAPÍTULOS

Esta dissertação está dividida em 7 capítulos. O Capítulo 2 faz uma descrição detalhada das redes veiculares, cujo funcionamento geral é explicado no item 2.1, a classificação das redes veiculares é detalhada no item 2.2, que está subdividido quanto às aplicações de transporte inteligente (item 2.2.1), quanto à arquitetura ou estrutura tecnológica (item 2.2.2) e quanto ao ambiente (ou cenário) de atuação (item 2.2.3). No Capítulo 3 são apresentadas as normas e os padrões VANET existentes, explicando os aspectos técnicos de transmissão referentes à camada física e enlace do modelo OSI (item 3.1), sobre os protocolos de roteamento (camada de rede) para redes VANET (item 3.2) e camada de transporte em redes VANET (item 3.3). Os principais problemas são discutidos no item 3.4 e as propostas de soluções no item 3.4.1. No Capítulo 4 são apresentados os principais simuladores de rede (item 4.1), os geradores de mobilidade (item 4.2) e simuladores VANET (item 4.3). O Capítulo 5 explica a proposta da dissertação, com detalhes do protocolo DSR (item 5.1), algoritmo genético (item 5.2) e a descrição detalhada do novo protocolo G-Net (item 5.3). O Capítulo 6 explica a configuração, os experimentos e resultados obtidos em simulações VANET, realizadas no ns3 com o VanetMobiSim em ambiente urbano. E por fim, o Capítulo 7 apresenta uma conclusão a respeito dos estudos realizados neste trabalho, bem como, apresenta as dificuldades encontradas e indica direções futuras de investigação.

## 2 REDES VEICULARES

O autor Cavalcanti (2008) explica que VANETs são compostas de veículos automotores como carros, caminhões, ônibus ou qualquer outro tipo que contenha em si mesmo os meios de propulsão com um dispositivo de comunicação sem fio embarcado que possuem a capacidade de se comunicarem entre si ou com equipamentos fixos posicionados à margem das ruas e/ou estradas.

O estudo sobre as VANETs é uma tendência atual que está motivando os ambientes industriais e acadêmicos na procura por soluções que possibilite instalar equipamentos de comunicação sem fio (*wireless*) nos veículos e/ou nas margens das vias automotivas (ruas, avenidas, rodovias, etc.), buscando apresentar uma confiabilidade aceitável na troca de informações, para que possam finalmente ser comercializados.

As VANETs possuem algumas classificações específicas que influenciam diretamente na escolha de protocolos e diferentes estratégias de soluções. Porém, é importante entender o funcionamento geral das VANETs para melhor compreensão de suas classificações.

### 2.1 FUNCIONAMENTO GERAL

Uma VANET consiste em uma rede de computadores *ad hoc* móvel (MANET) para veículos automotivos. Ao invés de acessar somente serviços de Internet e GPS, a VANET permite que os veículos obtenham conhecimentos de tráfego, assistência e segurança atualizados em tempo real, pois os carros podem trocar informações diretamente entre si (*ad hoc*), sem que haja uma conexão com a Internet ou qualquer outra rede externa.

Para que seja possível a existência deste tipo de rede é preciso disponibilizar equipamentos com antenas nos veículos com configurações e protocolos que permitam a comunicação entre os mesmos.

As configurações exigidas respeitam as normas e padrões de acordo com as camadas do modelo OSI da ISO (*International Standards Organization*). Sendo assim, é necessário

utilizar um protocolo da camada física e enlace comuns entre os veículos para que haja uma conexão física entre eles, um protocolo da camada de rede para fazer o roteamento dos pacotes e um protocolo da camada de transporte para segmentar, organizar e identificar os segmentos dos dados enviados pela camada de aplicação.

## 2.2 CLASSIFICAÇÃO DAS REDES VEICULARES

As redes *ad hoc* veiculares podem ser classificadas quanto às aplicações de transporte inteligente, quanto à arquitetura ou estrutura tecnológica e quanto ao ambiente (ou cenário) de atuação.

### 2.2.1 QUANTO ÀS APLICAÇÕES DE TRANSPORTE INTELIGENTE

As aplicações de transporte inteligente incluem uma variedade de aplicações, tais como a navegação *on-board*, monitoramento de tráfego cooperativo, controle dos fluxos de tráfego, a análise do congestionamento do tráfego em tempo real e rotas de desvio calculadas com base nas condições de tráfego e destino (LI; WANG, 2007).

Com relação às redes veiculares, além das aplicações específicas de trânsito, vislumbra-se o acesso à Internet em qualquer lugar e a qualquer instante. Além disso, quanto às aplicações de transporte inteligente, as VANETs podem ser divididas em dois grupos: aplicações de transporte inteligente e aplicações de conforto (LI; WANG, 2007).

Ao analisar a definição de autores diferentes, percebe-se que as aplicações de transporte inteligente apresentadas por Li e Wang (2007) correspondem às aplicações de segurança rodoviária e assistência, explicadas por Costa (2010). Sendo assim, para uma definição mais detalhada, Costa (2010) explica que existem três grandes classes de aplicações dentro da área dos sistemas de transportes inteligente: entretenimento, segurança rodoviária e assistência. Os autores Kamini e Kumar (2010) também dividem a VANET em três classes de aplicações, explicando que são orientadas a segurança, conveniência e comercial. Entretanto, os softwares comerciais correspondem a aplicações de entretenimento e as aplicações de conveniência referem-se à classe de assistência de Costa (2010).

Sendo assim, de acordo com a análise de diversos autores, as aplicações de transporte inteligente podem ser classificadas em três tipos diferentes: aplicações de entretenimento, aplicações de segurança rodoviária e aplicações de assistência.

### 2.2.1.1 APLICAÇÕES DE ENTRETENIMENTO

As aplicações de entretenimento também são conhecidas como aplicações de conforto, pois possibilitam a comunicação entre veículos ou com *hosts* da Internet, melhorando o conforto dos passageiros. Permitem as VANETs fornecerem conectividade de Internet aos usuários, como passageiros do banco traseiro, de tal forma que possam baixar música ou executar jogos enquanto o veículo está em movimento (LI; WANG, 2007). Além disso, segundo Costa (2010), dentre as aplicações de entretenimento destacam-se os serviços de mensagens instantâneas e compartilhamento de áudio e vídeo.

O autor Costa (2010) explica também que a maioria das aplicações de entretenimento das redes veiculares está associada à Internet. Por isso, é importante informar que normalmente os veículos não acessam a Internet diretamente, necessitando de recursos extras para esse tipo de aplicação. Desta forma, alguns *gateways* fixos ou dinâmicos são adicionados às redes, de modo que permitam a troca de mensagens entre a VANET e a Internet (LI; WANG, 2007).

Portanto, as aplicações comerciais irão fornecer ao motorista o entretenimento e serviços como acesso à *Web*, transmissão de áudio e vídeo, bem como a comunicação entre os passageiros em carros na mesma vizinhança, o que lhes permite participar de jogos iterativos (KAMINI; KUMAR, 2010).

Verifica-se que os veículos trocam diversas informações entre si, de tal forma que um dos maiores desafios é o fato das diversas aplicações de entretenimento possuir requisitos muito diferentes (COSTA, 2010), pois os recursos exigidos pela troca de áudio e vídeo são bastante diferentes das mensagens de e-mail ou sites de relacionamento. E também quando acessam a Internet, deve-se observar que normalmente é necessário um ponto fixo que funcione como *gateway* que permita qualquer tipo de comunicação externa.

### 2.2.1.2 APLICAÇÕES DE SEGURANÇA RODOVIÁRIA

Para evitar acidentes e congestionamentos é de grande importância que os condutores recebam alertas preventivos e emergenciais. De acordo com Costa (2010), devido ao elevado índice de mortalidade nas estradas, a necessidade de maior segurança rodoviária faz com que informar os condutores sobre situações críticas seja uma prioridade.

As aplicações de segurança têm um caráter preventivo, com o objetivo de divulgar rapidamente a informação, de tal forma que o condutor tenha um maior tempo de reação (COSTA, 2010).

Uma das formas de utilizar as aplicações de segurança é através de pontos fixos nas margens das rodovias. Desta forma, em caso de um acidente, os nós móveis podem transmitir essa informação a sensores do lado da estrada, que, em seguida, alertam sobre o congestionamento do tráfego ou podem contatar equipes de emergência (LI; WANG, 2007).

Com alertas sobre acidentes, obras ou sobre congestionamentos, os veículos tem condição de se preparar para possíveis desvios, reduções de velocidade ou até mesmo retornos. Sendo assim, dentre às aplicações de segurança rodoviária, destaca-se a divulgação de informações sobre acidentes, ocorrências no trânsito e condições adversas de ruas e estradas, atendendo diretamente ao condutor. Além disso, para essas aplicações, o principal desafio é reduzir o tempo desses alertas, de forma que o condutor possa reagir adequadamente de acordo com o potencial obstáculo, reduzindo a velocidade e estando preparado para o problema (CAVALCANTI, 2008).

Acompanhando este raciocínio, Kamini e Kumar (2010) explicam que basicamente as aplicações de segurança irão monitorar a estrada, a aproximação de veículos, a superfície e curvas da estrada.

Ainda com relação à segurança VANET, os autores Kamini e Kumar (2010) acrescentam que o sinal de tráfego pode ser disseminado através de aplicações de segurança como o SVA (*Slow/Stop Advisor Vehicle*), em que um veículo lento ou parado transmite mensagem de alerta para os seus vizinhos e como o CRN (*Congested Road Notification*) que detecta e avisa sobre congestionamentos rodoviários e podem ser usados para planejamento de rotas e viagem.

Outra forma de segurança informada por Kamini e Kumar (2010) é a visão aprimorada, onde os motoristas recebem uma visão clara de veículos e obstáculos em condições de névoa pesada e também aprendem sobre a existência de veículos escondidos por obstáculos, edifícios e por outros veículos.

### 2.2.1.3 APLICAÇÕES DE ASSISTÊNCIA

Ao contrário do que acontece nas informações de segurança, nas aplicações de assistência a divulgação rápida de informação não é um fator crítico, mas deve ser o mais abrangente possível (COSTA, 2010).

As aplicações de assistência, por exemplo, podem usar sensores laterais existentes nas estradas para monitorar a densidade de tráfego e velocidades veiculares e enviá-los a uma autoridade central para calcular os controles de fluxo de tráfego e melhorar a administração de

tempo dos semáforos (LI; WANG, 2007).

As aplicações de assistência ao condutor focam principalmente na automatização de tarefas como localização de postos de abastecimento, áreas de serviço e informações a nível turístico (COSTA, 2010). Isso é confirmado quando Li e Wang (2007) acrescentam que as aplicações de assistência podem fornecer prestação de serviços de consulta de informações de pontos de interesse próximos de uma determinada rota, como os próximos postos de gasolina ou hotéis.

Desta forma, as aplicações de conveniência serão principalmente do tipo de gestão de tráfego, segundo Kamini e Kumar (2010) ou para serviços de consulta de informações sobre pontos de interesse localizados as margens das rodovias como, postos de abastecimento, lojas de conveniência, farmácias ou informações turísticas.

Outra forma de assistência (conveniência) VANET útil para os motoristas seria utilizar um software capaz de alertar os motoristas quanto à condição do tempo. Os autores Kamini e Kumar (2010) informam que informações sobre o tempo podem ser atualizados/solicitados por um aplicativo via DSRC (*Dedicated Short Range Communications*). Além disso, a Notificação de Disponibilidade de Estacionamento (PAN - *Parking Availability Notification*) ajuda a encontrar disponibilidade de espaço para estacionamento em uma determinada área geográfica de acordo com a condição meteorológica.

## 2.2.2 QUANTO À ARQUITETURA OU ESTRUTURA TECNOLÓGICA

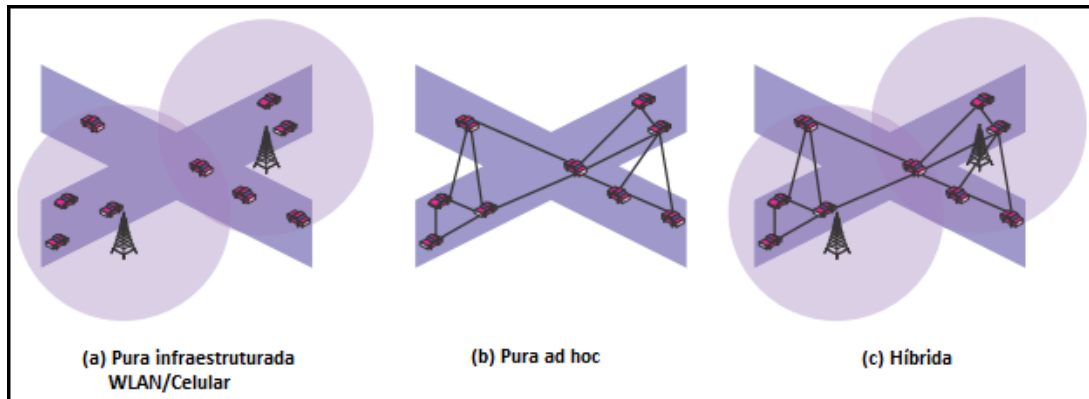
Existem duas formas de configuração das VANETs, uma baseada em comunicação entre veículos, identificada como V2V (*Vehicle-to-Vehicle*) (BHAKTHAVATHSALAM; NAYAK, 2011; MARTINEZ et al., 2011; LI; WANG, 2007; CAVALCANTI, 2008; KAMINI; KUMAR, 2010), também definida como V2VC (*Vehicle-to-Vehicle Communication*) (MENOVAR et al., 2006; THAKUR et al., 2013) ou IVC (*Inter-Vehicle Communication*) (LI; WANG, 2007; CAVALCANTI, 2008) e; a outra forma é baseada na comunicação conjunta entre veículos e infraestrutura das rodovias, conhecida como V2I (*Vehicle-to-Infrastructure*) (BHAKTHAVATHSALAM; NAYAK, 2011; MARTINEZ et al., 2011; CAVALCANTI, 2008; KAMINI; KUMAR, 2010), V2IC (*Vehicle-to-Infrastructure Communication*) (THAKUR et al., 2013) ou RVC (*Road-vehicle Communications*) (CAVALCANTI, 2008).

Normalmente as VANETs necessitam apenas de recursos básicos de redes móveis, como antenas *wireless* e protocolos configurados nos veículos. Entretanto, como existem situações em que outras aplicações e equipamentos são necessários, as VANETs podem



trabalhar de forma pura ou híbrida.

Segundo os autores Li e Wang (2007), a arquitetura das VANETs divide-se em três categorias: puro celular/sem fio, puro *ad hoc*, e híbridos conforme visualizado na Figura 1.



**Figura 1: Categorias VANET quanto à arquitetura ou estrutura tecnológica.**

**Fonte: Adaptado de Li e Wang (2007)**

A seguir será explicada cada uma destas categorias VANET quanto à arquitetura ou estrutura tecnológica.

#### 2.2.2.1 ARQUITETURA PURA WLAN/CELULAR OU INFRAESTRUTURADA (V2I)

As VANETs podem usar *gateways* celulares fixos e pontos de acesso WLAN (*Wireless Local Area Network*) nos cruzamentos para obterem acesso à Internet (Figura 1 (a)), captarem informações sobre o trânsito, ou para realizar roteamento (LI; WANG, 2007).

Conforme já informado, esta forma de comunicação também é conhecida como V2I, V2IC ou RVC e os veículos dependem de infraestruturas com antenas posicionadas nas margens das vias para trocarem informações entre si.

A arquitetura V2I realiza a comunicação entre veículos e pontos de acesso posicionados nas vias, possivelmente em sinais de trânsito, postes de iluminação, postos de gasolina e acostamentos, dentre outros (CAVALCANTI, 2008).

Sabendo que pontos fixos nas margens das vias podem intermediar a comunicação entre os veículos, *gateways* celulares fixos e pontos de acesso WLAN seriam suficientes para facilitar toda a troca de informações entre os nós. Entretanto, como a compra de equipamentos fixos aumentaria muito o custo da estrutura da rede, então criar uma rede totalmente *ad hoc* entre os veículos poderia atingir determinados objetivos como detecção de obstáculos no ponto cego do veículo (LI; WANG, 2007).

Além disso, apesar da arquitetura baseada em infraestrutura aproveitar as redes celulares existentes, ela apresenta três inconvenientes: alto custo de operação, a largura de banda limitada e alocação do mesmo canal para *uplink* e *downlink* (YANG et al., 2008).

#### 2.2.2.2 ARQUITETURA PURA AD HOC OU V2V

As redes *ad hoc* não precisam de infraestrutura, de modo que o custo de construção de tal rede é muito baixo e pode funcionar em eventos de desastres (YANG et al., 2008).

Devido a falta de infraestrutura nas margens das rodovias, os veículos possuem a capacidade de gerenciar toda troca de informações. Por isso, a rede pura *ad hoc* (Figura 1 (b)) é conhecida como V2V, ou seja, quando ocorre somente a comunicação entre veículos.

A arquitetura pura *ad hoc* também é conhecida como V2VC ou IVC. Os autores Blum et al. (2004) explicam que as redes IVC possuem rápidas mudanças de topologia que são difíceis de gerenciar, também estão sujeitas à frequentes fragmentações e a redundância nestas redes é muito limitada no tempo e na função.

Conforme já informado, o conceito de rede *ad hoc* não exige pontos fixos porque os nós móveis são capazes de se comunicar diretamente entre si. No caso das VANETs puras, portanto, os veículos trocam informações sem a necessidade de dispositivos fixos nas margens das rodovias, formando uma rede totalmente *ad hoc*. Entretanto, em algumas situações pode ser necessário acessar uma rede externa e o uso de pontos fixos pode ser muito útil nestas situações.

#### 2.2.2.3 ARQUITETURA HÍBRIDA

A arquitetura híbrida (Figura 1 (c)) interliga a tecnologia celular, WLAN e redes *ad hoc* como possível solução para VANETs (LI; WANG, 2007).

Para que a VANET seja funcional, a LAN (*Local Area Network*) sem fio é usada quando um ponto de acesso está disponível e caso contrário uma conexão 3G é usada. Desta forma, assume-se que cada usuário teria um hardware preparado para os dois tipos de conexão, bem como os *gateways* deveriam ter tanto WLAN quanto interfaces de celulares (NAMBOODIRI et al., 2004).

A estrutura híbrida trabalha com as duas arquiteturas VANET de tal forma que os veículos se comunicam normalmente pela rede *ad hoc*, mas também aproveitam os pontos fixos das margens. A arquitetura híbrida também requer a função de comunicação multissalto entre os veículos, o que é a parte essencial da rede *ad hoc* (YANG et al., 2008).

### 2.2.3 QUANTO AO AMBIENTE (OU CENÁRIO) DE ATUAÇÃO

Quanto ao ambiente de atuação, as VANETs podem funcionar em dois tipos de cenários: Ambiente urbano e ambiente de rodovias (ou estradas). Apesar de ambas apresentarem dificuldades em relação às MANETs, cada um desses cenários possui suas características específicas que influenciam diretamente nas formas de configuração e no desempenho da rede.

#### 2.2.3.1 AMBIENTE DE RODOVIA

No ambiente de rodovia, apesar da grande velocidade dos nós (veículos) na pista, normalmente a posição destes é bem previsível, pois existem poucos obstáculos, poucos semáforos e poucas curvas bruscas ou desvios. Os autores Li e Wang (2007) afirmam que no tráfego de cenários de estrada, o ambiente é relativamente simples e direto (por exemplo, restrito a movimento unidimensional).

#### 2.2.3.2 AMBIENTE URBANO

No ambiente urbano a rede veicular tende a ser muito mais complexa do que em rodovias. No sentido de comunicação de dados, nem sempre há uma linha direta de comunicação porque as ruas da cidade são muitas vezes separadas por edifícios, árvores e outros obstáculos (LI; WANG, 2007).

Além disso, o tráfego de veículos costuma ser mais intenso devido à quantidade de veículos em vias mais estreitas. Em um cenário urbano, por exemplo, o tráfego lento devido ao congestionamento em uma rua é comum (BHAKTHAVATHSALAM; NAYAK, 2011). Também existem semáforos e cruzamentos que tornam o movimento bem mais instável e imprevisível do que em rodovias.

Por esses motivos, num cenário urbano a densidade de uma VANET pode atingir dezenas de milhares de nós e nenhum protocolo de roteamento MANET convencional está preparado para suportar redes tão grandes (LEE et al., 2006).

Por outro lado, com a grande densidade de veículos, diminui a necessidade de nós fixos nas margens das ruas, reduzindo o custo e permitindo muitas vezes a utilização de uma rede totalmente *ad hoc* para a troca de informações.

## 2.3 PRINCIPAIS PROBLEMAS EXISTENTES NAS VANETS

As MANETs são redes preparadas para trabalhar com antenas móveis que podem enviar e receber informações de controle de rede sem fio. Porém, na comunicação entre veículos acontecem problemas que as MANETs não estão habilitadas a solucionar. Neste contexto, enquanto nas redes móveis comuns as pessoas se deslocam em pequenos espaços com seus computadores e normalmente existem pontos fixos que facilitam a comunicação constante, nas redes veiculares a velocidade relativamente elevada dos veículos causa mudanças rápidas e frequentes na topologia da rede (MENOUAR et al., 2006).

Além disso, existem vários outros problemas como estimativa da largura de banda disponível, o controle de acesso ao meio, problema de nós escondidos e expostos, alta mobilidade, veículos heterogêneos, o movimento do nó, velocidade rápida, obstáculos e entrega rápida para os quais as MANETs não conseguem dar suporte (MENOUAR et al., 2006; BLUM et al., 2004; LOCHERT et al., 2003).

Nas VANETs existem ainda problemas de nós disjuntos e caminhos disjuntos que se formam devido aos problemas mencionados anteriormente e que fazem parte das mudanças da topologia (MAOWAD; SHAABAN, 2012). Além disso, dependendo da tecnologia adotada, o custo por bit pode ser mais alto com o aumento do número de assinantes (BHAKTHAVATHSALAM; NAYAK, 2011). Este último não chega a ser um problema grave, mas o custo pode ser minimizado de acordo com a tecnologia adotada.

Basicamente, os problemas das VANETs são decorrentes da complexidade dos movimentos dos veículos que os protocolos MANETs não conseguem resolver. Para entender como esses problemas podem ser resolvidos, é melhor analisar propostas de alterações dos protocolos de acordo com as camadas do modelo OSI.

### 2.3.1 PROPOSTAS DE SOLUÇÕES PARA OS PROBLEMAS EM VANETS

Devido aos problemas discutidos no início deste capítulo, diversas soluções têm sido propostas para os problemas detectados em redes VANET. Com relação às soluções apresentadas pela comunidade científica, normalmente os pesquisadores realizam modificações em protocolos da camada física, enlace, rede e transporte. Mas algumas soluções de protocolos totalmente novos para as VANETs também estão sendo criados.

Na camada física e de enlace, geralmente as modificações são feitas em protocolos IEEE 802.11 (Wi-Fi) e IEEE 802.16 (WiMAX). No Wi-Fi, foi elaborado o padrão IEEE

802.11p, específico para VANET e uma família de protocolos IEEE 1609 (WAVE) foi criada unicamente para lidar com as redes veiculares, cujo IEEE 1609.4 corresponde a uma atualização do IEEE 802.11p com algumas funções extras e as demais derivações WAVE apresentam muito mais funções específicas para VANETs. No WiMAX, surgiu o padrão IEEE 802.16e que evoluiu para o IEEE 802.16m conforme pode ser verificado no item 3.1 deste trabalho. Além disso, os autores Bhakthavathsalam e Nayak (2011) mostram que o custo por bit é menor com IEEE 802.16e quando comparado a outras tecnologias de rede *wireless*.

Como ocorrem problemas com a largura de banda disponível, nós escondidos e expostos, alta mobilidade, velocidade rápida e obstáculos (MENOVAR et al., 2006; BLUM et al., 2004; LOCHERT et al., 2003), além do problema de disjunção frequente (MAOWAD; SHAABAN, 2012), na tentativa de solucionar estes problemas, na camada de rede são propostas várias modificações dos protocolos de roteamento MANET existentes como as variações do protocolo AODV (AOMDV (*Ad hoc On-Demand Multi-path Distance Vector*), AODV+PGB, PRAODV (*Prediction Based AODV routing*), PRAODV-M (PRAODV - *Maximum predicted*), AODV-bis, etc.)), GPSR (GPSR+AGF, GpsrJ+) e DYMO (Ant-DYMO (*Ant Colony Optimization - DYMO*), MAR-DYMO (*Mobility-aware Ant Colony Optimization Routing*)).

Apesar das alterações dos protocolos MANET, em algumas situações, protocolos de roteamento são criados especificamente para VANET como é o caso dos protocolos CAR e MORA.

Na camada de transporte, os autores Khorashadi et al. (2007) explicam que o protocolo UDP não costuma apresentar problemas, mas como existem dificuldades com o protocolo TCP, algumas modificações para criar uma variação do UDP confiável são apresentadas como o UDP com rajada confiável (RB-UDP) (HE et al., 2002).

Além disso, devido aos problemas enfrentados pelo TCP em redes VANET, diversas variações deste protocolo são apresentados como TCP-F, TCP-ELFN, TCP-*Bus*, ATCP e *Split-TCP*, conforme demonstra (TOOR et al., 2008).

Apesar das evoluções dos protocolos MANETs, na camada de transporte também foram criados protocolos totalmente do zero, especificamente para as redes veiculares, como é o caso do ACTP, do ATP e do VTP.

Quanto as pesquisas, universidades costumam realizar experimentos com 3 ou 4 veículos para análise dos protocolos em ambientes reais. Entretanto, verificar o comportamento dos protocolos com maiores quantidades de nós em ambientes reais ainda é um desafio para a

comunidade científica.

Neste trabalho é realizada uma forma de simulação variando de 20 a 100 veículos em um ambiente tipicamente urbano. Também foram realizadas configurações de cenários o mais próximo possível da realidade com aceleração, frenagem, semáforos, vias com 2 ou mais faixas, ultrapassagens e pequenas pausas na transmissão de dados.

Devido a problema de nós escondidos e expostos, alta mobilidade e velocidade dos nós e a constante mudança de topologia que refletem na redução da taxa de entrega, nesta dissertação também é apresentada uma proposta de um protocolo VANET chamado de G-Net. Este protocolo foi criado com o intuito de melhorar a taxa média de entrega de dados nas VANETs unindo características do DSR, atualização de rotas constantes e aplicação de algoritmo genético para otimização das rotas.

### 3 NORMAS E PADRÕES PARA REDES SEM FIO E VANETS

Apesar de VANET ser um termo relativamente recente, seu princípio de funcionamento utiliza conceitos e tecnologias de redes móveis *ad hoc* (MANET) que já foram bastante discutidos pela comunidade científica e cujas normas e padrões podem ser aproveitados. Entretanto, devido a grande velocidade dos nós *ad hoc*, do número aleatório de veículos e outras características já discutidas, aos poucos estão surgindo algumas normas e padrões específicos para VANETs que podem ser melhor entendidos conforme as camadas do modelo OSI em que são utilizados.

#### 3.1 ASPECTOS TÉCNICOS DE TRANSMISSÃO (CAMADA FÍSICA E ACESSO AO MEIO)

Alguns padrões específicos estão sendo criados para VANET de acordo com as camadas do modelo OSI. O IEEE (*Institute of Electrical and Electronics Engineers*), por exemplo, está desenvolvendo padrões como o WAVE (*Wireless Access in Vehicular Environments*), onde é proposta uma nova família de quatro protocolos dedicados à comunicação entre veículos, denominados IEEE 1609 (CAVALCANTI, 2008). Desta forma, o 1609.1 descreve o gerenciamento de recursos (*Resource Manager*) o IEEE 1609.2 é responsável pela segurança da comunicação (LI, 2012; IEEE, 2006), o padrão IEEE 1609.3 refere-se a configuração e gerenciamento de conexão (IEEE, 2010) e o IEEE 1609.4 especifica a operação multicanal (IEEE, 2011) e as extensões para o 802.11p (CAVALCANTI, 2008).

Os autores Bhakthavathsalam e Nayak (2011) informam que o IEEE 802.11p é um projeto de alteração no padrão IEEE 802.11 que incorpora aplicações para redes de veículos de mudanças rápidas, e também afirmam que a taxa de dados máxima suportada por este padrão é de 27 Mbps, contra 54 Mbps de outros padrões 802.11. Além disso, o WAVE utiliza o DSRC para comunicação por radiofrequência (CAVALCANTI, 2008).

Os autores Jiang et al. (2006) informam que em 1999 o FCC (*Federal Communication Commission*) alocou 75 MHz de DSRC na frequência de 5,9 GHz exclusivamente para uso de

V2V e V2I. Além disso, o DSRC é dividido em sete canais, de 10 MHz cada, os quais são utilizados para fins específicos como canais de controle, de serviço e canais para avanços futuros como aplicações para evitar acidentes. Também é útil lembrar que a tecnologia de rádio do DSRC é uma extensão do padrão IEEE 802.11a ajustado para operações com baixo *overhead* no espectro e a camada MAC (*Media Access Control*) é equivalente ao IEEE 802.11e EDCA (*Enhanced Distributed Channel Access*) com qualidade de serviço (CAVALCANTI, 2008).

Os sistemas de comunicações V2V e V2I permitem, geralmente, que os veículos se movam a velocidades de até 140 km/h numa área de visibilidade de até 1000 m (IEEE, 2006). Além disso, os autores Li et al. (2010) acrescentam que o padrão 802.11p foi projetado para trabalhar com os veículos a uma velocidade de até 250 km/h e um alcance para transmissão de 1000 metros.

Como o WAVE se caracteriza como uma tecnologia específica para VANET, para entender um pouco mais sobre as modificações ocorridas no padrão IEEE 1609 em relação ao IEEE 802.11, o artigo (LI, 2012) faz uma comparação entre WAVE e Wi-Fi (*Wireless Fidelity*), conforme pode ser observado na Tabela 1.

**Tabela 1: Comparação entre WAVE e Wi-Fi com parâmetros OFDM.**

Parâmetros	WAVE	Wi-Fi
Freq. operação (GHz)	5,9	2,4/5
Largura de banda do canal (MHz)	10	20
Taxa de dados (Mbps)	3/4.5/6/9/12/18/24/27	6/9/12/18/24/36/48/54
Modulação	Idem Wi-Fi	BPSK, QPSK, 16 QAM e 64 QAM
Codif. do canal	Idem Wi-Fi	Código convol. com taxa: 1/2; 2/3 e 3/4
Nº de subportadoras de dados	Idem Wi-Fi	48
Nº de subportadoras piloto	Idem Wi-Fi	4
Nº de subportadoras de virtuais	Idem Wi-Fi	12
Tamanho FFT/IFFT	Idem Wi-Fi	64
Intervalo FFT/IFFT ( $\mu$ S)	6,4	3,2
Espaçamento de subportadora (MHz)	0,15625	0,3125
Intervalo CP ( $\mu$ S)	1,6	0,8
Intervalo de símbolo OFDM ( $\mu$ S)	8	4

**Fonte: Adaptado de Li (2012).**

Os autores Bhakthavathsalam e Nayak (2011) mostraram que WAVE (IEEE 802.11p) e WiMAX móvel (IEEE 802.16e) eram as tecnologias de rede com capacidade adequada para VANETs e realizaram um estudo comparativo entre estas duas tecnologias apresentando uma avaliação de desempenho por meio de simulações realistas.

O artigo (BHAKTHAVATHSALAM; NAYAK, 2011) explica ainda que, o WiMAX móvel é uma extensão móvel do WiMAX que visa a oferta de banda larga sem fio para redes metropolitanas e esta tecnologia é otimizada por canais de rádio móveis dinâmicos e suporta



transferências e *roaming* de até 100 quilômetros por hora.

O padrão 802.16m representa uma atualização conjunta de alguns padrões 802.16, inclusive do 802.16e (MIN et al., 2009). Os resultados obtidos através de pesquisas com simulações demonstram que o canal com a norma 802.16m tem um bom desempenho de taxa de erro de bit (BER - *Bit Error Rate*), em comparação com o método convencional, e à velocidade de 350 km/h tem o mesmo desempenho da velocidade de 3 km/h como é demonstrado por (MIN et al., 2009).

O WiMAX móvel (IEEE 802.16e) pode suportar o serviço de banda larga de qualidade em uma configuração celular, tornando-se um dos mais fortes concorrentes para VANETs, pois as modificações em sua camada física (PHY - *Physic layer*) e camada de controle de acesso ao meio (MAC) tornam o WiMAX móvel adequado para tais aplicações (BHAKTHAVATHSALAM; NAYAK, 2011).

Com relação à camada física, o IEEE 802.16e suporta OFDM (*Orthogonal Frequency Division Multiplexing*), que divide a largura de banda em canais estreitos, sendo que cada banda estreita ocupa uma frequência diferente que pode levar diferentes partes de uma mensagem separadamente (VAUGHAN-NICHOLS, 2008; BHAKTHAVATHSALAM; NAYAK, 2011).

Uma outra vantagem do WiMAX móvel é que na camada MAC ele suporta tanto o modo ponto-a-multiponto (PMP - *Point Multipoint*) quanto o modo de comunicação em malha (LU et al., 2007; BHAKTHAVATHSALAM; NAYAK, 2011).

Apesar das vantagens, há alguns desafios enfrentados pelo WiMAX móvel como a alocação de espectro, pois não há espectro mundial licenciado para o WiMAX (BHAKTHAVATHSALAM; NAYAK, 2011). Sendo assim, dependendo da região ele opera em frequência diferente, adotando por exemplo, 2,3 GHz na região do Pacífico da Ásia, 2,5 GHz nos Estados Unidos e 5,5 GHz na União Europeia (VAUGHAN-NICHOLS, 2008; BHAKTHAVATHSALAM; NAYAK, 2011).

Outro problema do WiMAX é com relação ao número de assinantes de acesso à Internet. Um dos desafios dos provedores de serviços é oferecer um *backbone* de Internet com largura de banda suficiente para fornecer um apoio de transporte necessário que suporte um número potencialmente grande de clientes de WiMAX móvel (VAUGHAN-NICHOLS, 2008; BHAKTHAVATHSALAM; NAYAK, 2011).

### 3.2 PROTOCOLOS DE ROTEAMENTO (CAMADA DE REDE)

De acordo com Lee et al. (2010), um protocolo de roteamento controla a maneira com que duas entidades de comunicação trocam informações entre si, incluindo o procedimento para o estabelecimento de uma rota, a decisão de encaminhamento, e ação na manutenção da rota ou recuperação de falha de roteamento.

Assim como outras camadas do modelo ISO/OSI, os protocolos de roteamento utilizados em VANET são normalmente derivados da MANET. Desta forma, existem duas formas de classificar os protocolos de roteamento VANET: quanto à área de abrangência (ou forma de controle) e quanto ao tipo de propagação de pacotes de roteamento. A primeira, quanto à área de abrangência (ou forma de controle), cujos protocolos de roteamento *ad hoc* podem ser classificados em duas grandes áreas, de acordo com (WANG et al., 2010) o protocolo de roteamento pode ser baseado em topologia ou roteamento geográfico, cujas subclassificações serão discutidas nas próximas seções. E quanto ao tipo de propagação de pacotes de roteamento, os protocolos podem ser *unicast*, *broadcast* ou *multicast/geocast*, como será mostrado no item 4.2.2.

É importante observar que a forma de propagação dos pacotes de roteamento nem sempre têm relação direta com a classificação dos protocolos quanto à área de abrangência. Desta forma, é possível observar protocolos unicast baseados em topologia como AODV e DSR, bem como baseados em roteamento geográfico como os protocolos GPSR (*Greedy Perimeter Stateless Routing*), GPCR (*Greedy Perimeter Coordinator Routing*), STAR (*Spatial and Traffic Aware Routing*) e GSR (*Geographic Source Routing*) conforme apresentado por (BERNSEN; MANIVANNAN, 2009).

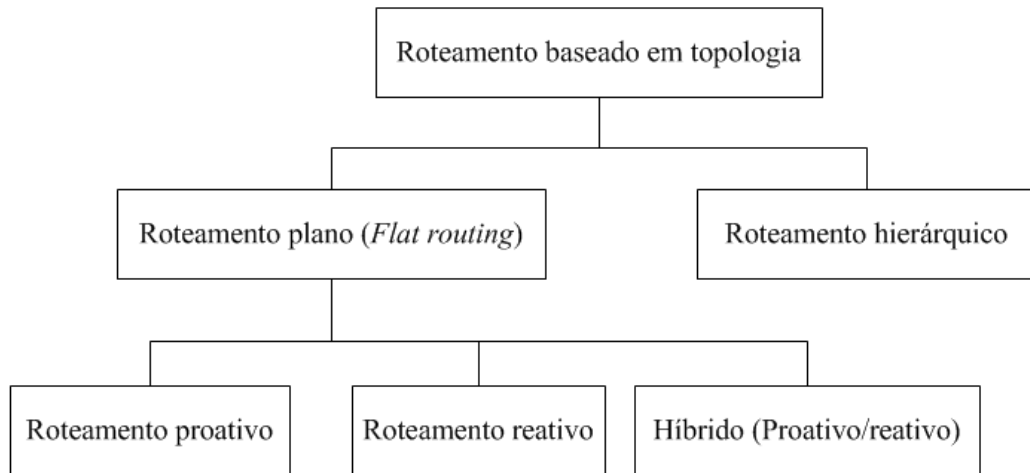
#### 3.2.1 CLASSIFICAÇÃO DOS PROTOCOLOS DE ROTEAMENTO PELA ÁREA DE ABRANGÊNCIA (OU FORMA DE CONTROLE)

Para conhecer os nós vizinhos e manter as rotas entre eles, os protocolos precisam trocar informações de rotas para atualizar suas tabelas de roteamento.

Conforme já informado, pela área de abrangência ou forma de controle, os protocolos de roteamento VANET podem atualizar suas rotas baseados em topologia ou através do roteamento geográfico.

### 3.2.1.1 PROTOCOLO DE ROTEAMENTO BASEADO EM TOPOLOGIA

O roteamento baseado em topologia, ainda pode ser analisado pela forma de atualização de rotas, sendo classificado em roteamento plano (*Flat routing*), cujos protocolos podem ser definidos como proativos (*Table-Driven*), reativos (*On-Demand*) ou híbridos (proativo/reactivo); ou classificado em roteamento hierárquico, conforme observado na Figura 2.



**Figura 2: Classificação dos protocolos de roteamento baseados em topologia**

**Fonte: Autoria própria.**

Para entender melhor os tipos de protocolos apresentados na Figura 2, a seguir são explicadas as diferenças a respeito das subclassificações dos protocolos de roteamento baseados em topologia.

#### 3.2.1.1.1 ROTEAMENTO PLANO (*FLAT ROUTING*)

De acordo com Silva (2008) os protocolos de roteamento plano consideram que todos os nós são iguais do ponto de vista funcional, de tal forma que todos os nós executam o mesmo algoritmo de roteamento, sem diferenciação hierárquica ou física entre eles.

Os autores Hong et al. (2002) explicam que o roteamento plano é classificado em duas classes: proativa e reativa. Entretanto, os autores Wang et al. (2010) afirmam que os protocolos de roteamento baseados em topologia podem ser divididos em três categorias: proativo, reativo e híbrido.

### A. Protocolos de roteamento proativos (*Table-driven*)

Em (WANG et al., 2010), os autores explicam que nos protocolos de roteamento proativos, cada nó móvel mantém sua tabela de roteamento e usa outros nós para transmitir suas próprias informações de roteamento periodicamente para troca de dados e atualizar o caminho para o destino.

Como o protocolo de roteamento proativo tem que manter a tabela de roteamento, Wang et al. (2010) informam que também é chamado de protocolo de roteamento de tabela dirigida (*table-driven*), termo que também pode ser confirmado em (LEE et al., 2010; PAUL et al., 2011a).

De acordo com Xu et al. (2010), os protocolos de roteamento proativos têm uma resposta rápida às mudanças de topologia, monitorando continuamente as mudanças e disseminando através da rede as informações relacionadas conforme necessário. Apesar dos protocolos proativos apresentarem uma resposta rápida às mudanças de topologia, essa rapidez ocasiona um aumento de sinalização e conseqüentemente pode resultar em menores taxas de entrega de pacotes e atrasos mais longos quando ocorre muita mudança de topologia. E no pior dos casos, tempestades de *broadcast* podem resultar em congestionamento de toda a rede.

Os protocolos de roteamento proativos tem a vantagem de não necessitar descobrir rotas e também possuem baixa latência para aplicações em tempo real. Por outro lado, um ponto negativo dos protocolos proativos é o fato de incluírem caminhos não utilizados nas mensagens de atualização de rotas, e isso acaba ocupando parte significativa da largura de banda disponível (PAUL et al., 2011a).

Um exemplo de protocolo proativo apresentado por (LEE et al., 2010; PAUL et al., 2011a) é o FSR (*Fisheye State Routing*) (IWATA et al., 1999; PEI et al., 2000). Outros exemplos são o protocolo FSLs (*Fuzzy Sighted Link State*), o OLSR (*Optimized Link State Routing Protocol*) e TBRPF (*Topology Broadcast Based on Reverse Path Forwarding*) exibidos por (WANG et al., 2010; SINGH et al., 2011), e também DSDV (*Destination-Sequenced Distance-Vector*), WRP (*Wireless Routing Protocol*) e CGSR (*Clusterhead Gateway Switch Routing*) apresentados por (ROYER; TOH, 1999).

### B. Protocolos de roteamento reativos (*On-Demand*)

Conforme os autores Wang et al. (2010) explicam, os protocolos reativos não transmitem informações de roteamento periódica e automaticamente para outros nós na rede. Sendo assim as rotas só são estabelecidas quando um nó móvel precisa entregar um pacote.

Como as rotas só são estabelecidas quando necessário, os autores Wang et al. (2010) informam que os protocolos de roteamento reativos também são chamados de protocolos de roteamento sob demanda (*On-Demand*).

De acordo com Lee et al. (2010) os protocolos reativos abrem um percurso apenas quando um nó necessita se comunicar com outro. Além disso, eles mantêm apenas as rotas que estão atualmente em uso, reduzindo assim a carga na rede.

Normalmente o roteamento reativo tem uma fase de descoberta de rotas onde os pacotes de consulta são disseminados na rede em busca de um caminho e esta fase termina quando uma rota é encontrada, explica (LEE et al., 2010).

Os autores Paul et al. (2011a) informam que os protocolos reativos têm a vantagem de não requerer inundações periódicas da rede e poupam largura de banda porque não trabalham com *beacon*<sup>1</sup>. Por outro lado, a latência é alta para encontrar rotas e a inundação excessiva da rede provoca falhas na comunicação entre nós.

Com relação aos protocolos de roteamento reativos, de acordo com Xu et al. (2010), quando ocorre uma nova sessão de tráfego, os nós têm que configurar os caminhos entre origem e destino antes de começar a entregar os pacotes de dados. Esse processo de configuração do caminho é conhecido como descoberta de rota. Além disso, esses protocolos também necessitam de outro processo conhecido como manutenção de rota para encontrar um caminho alternativo se um caminho anterior falhar.

Como exemplo de protocolo reativo os artigos (LEE et al., 2010; PAUL et al., 2011a) apresentam o AODV, PGB (AODV + *Preferred Group Broadcasting*), TORA (*Temporally Ordered Routing Algorithm*) e o DSR, em (BISOYI; SAHU, 2010) é possível observar o DYMO (*DYNAMIC MANET On Demand*) e Royer e Toh (1999) ainda acrescentam os protocolos LMR (*Lightweight Mobile Routing*), ABR (*Associativity-Based Routing*) e SSR (*Signal Stability-Based Adaptive Routing*).

### C. Protocolos de roteamento híbrido (Proativo/Reativo)

Conforme informado anteriormente, o protocolo de roteamento plano pode ainda ser classificado como híbrido. Neste caso, o protocolo híbrido apresenta características tanto do protocolo proativo quanto do protocolo reativo.

Para Dongre et al. (2012) a necessidade dos protocolos híbridos surge devido às

---

<sup>1</sup>De acordo com Paul et al. (2011a) (grifo do autor), *beacon* significa transmissão periódica de uma mensagem “Olá” curta que informa a presença e posição do nó.

deficiências de roteamento proativo e reativo e quando há demanda de tal protocolo que pode resolver a descoberta de rotas sob demanda com um número limitado de pesquisas de rotas.

De acordo com Hong et al. (2002) o esquema híbrido proativo/reativo limita a sobrecarga proativa para apenas o tamanho da zona, e a sobrecarga de pesquisa reativa para nós selecionados.

Segundo Kaur e Singh (2013) o protocolo de roteamento híbrido é aquele em que o roteamento é inicialmente estabelecido com algumas rotas proativas e serve então a demanda de nós adicionalmente ativados através de inundações reativas.

Os autores Dongre et al. (2012) informam que a maioria dos protocolos de roteamento híbridos são baseados em zona, como o ZRP (*Zone Routing Protocol*), mas explica que a desvantagem do ZRP é que ele não é projetado para ambientes com mudanças altamente dinâmicas e rápidas na topologia, como VANET.

Os autores Kaur e Singh (2013) afirmam que alguns dos protocolos de roteamento híbridos são o HWMP (*Hybrid Wireless Mesh Protocol*), OORP (*OrderOne Routing Protocol*) e ZRP.

### 3.2.1.1.2 ROTEAMENTO HIERÁRQUICO

Para os autores Hong et al. (2002) quando ocorre o aumento de tamanho da rede sem fio acima de certos limites, o roteamento plano tornar-se inviável por causa da ligação e sobrecarga de processamento e uma maneira de resolver este problema é utilizando o roteamento hierárquico.

O roteamento hierárquico difere do roteamento plano por apresentar uma divisão de classes entre os nós. De acordo com Silva (2008) no roteamento hierárquico existe uma divisão dos nós da rede em duas classes: nós comuns e líderes de grupo (*cluster-heads*).

Os protocolos hierárquicos possuem uma área de monitoramento que concentra algumas regiões onde os nós de cada uma das regiões formam um grupo (*cluster*) e elegem um dos nós para ser o líder do grupo (*cluster-head*) conforme explica (SILVA, 2008).

Para Hong et al. (2002) a forma mais popular de configurar um roteamento hierárquico é reunindo os nós, geograficamente próximos, em grupos (*clusters*) explícitos. Cada grupo tem um nó principal (*cluster-head*) para se comunicar com outros nós em nome do grupo.

Além disso, o autor Silva (2008) explica que os líderes de grupo também podem

executar algumas funções como agregação de dados ou roteamento apenas entre os líderes de grupo antes de enviar os dados.

Os autores Hong et al. (2002) informam que no roteamento hierárquico tanto o tamanho da tabela de roteamento quanto do pacote de atualização são reduzidos por incluir apenas uma parte da rede, diminuindo a sobrecarga de pacotes de controle.

Como exemplo de protocolo hierárquico pode-se citar o HSR (*Hierarchical State Routing*), o CGSR e o LANMAR (*Landmark Ad Hoc Routing Protocol*).

### 3.2.1.2 ROTEAMENTO GEOGRÁFICO

De acordo com Lee et al. (2010), no roteamento geográfico (baseado em posição), a decisão de encaminhamento é feita principalmente com base na posição de destino de um pacote e a posição dos vizinhos de um salto do nó, sendo que a posição do destino é armazenada no cabeçalho do pacote.

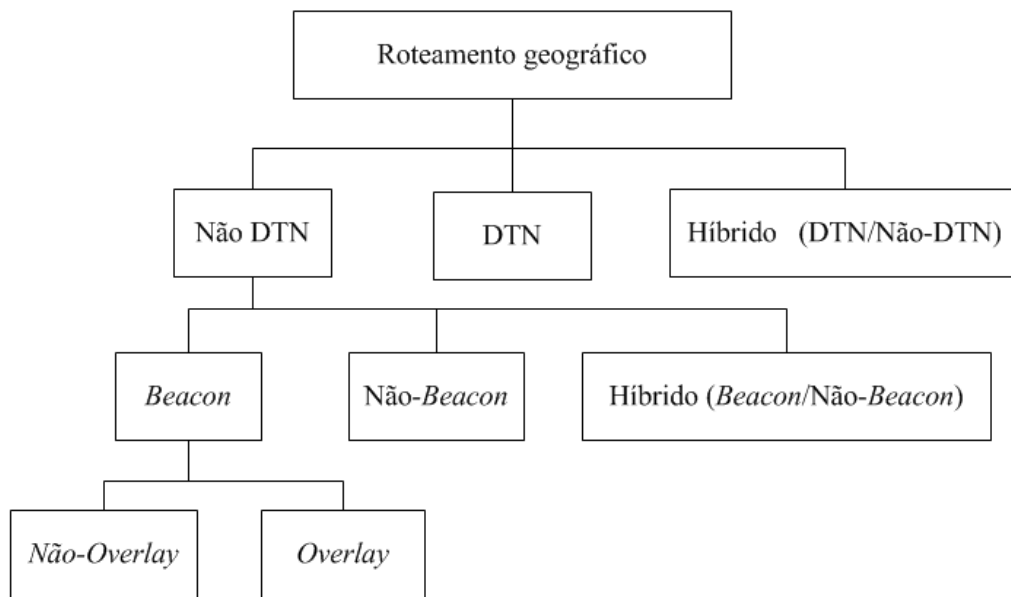
O artigo (PAUL et al., 2011a), acrescenta que o roteamento geográfico é um roteamento em que cada nó conhece a sua própria posição geográfica e do nó vizinho determinando serviços como GPS. Ele não mantém qualquer tabela de roteamento, nem realiza qualquer troca de informação de estado de ligação com os nós vizinhos.

É importante lembrar que as informações do dispositivo GPS são usadas para decisão de roteamento como alerta (PAUL et al., 2011a). E o autor Silva (2008) acrescenta que as informações de posicionamento dos nós usadas no roteamento podem ser obtidas através de um equipamento de localização (GPS), ou através de um sistema de posicionamento local, definido entre os nós vizinhos em uma determinada região.

De acordo com Paul et al. (2011a) o roteamento geográfico apresenta vantagens porque não necessita descobrir e administrar rotas, possui escalabilidade e é adequado para o padrão de alta mobilidade de nó. Por outro lado, são necessários serviços para determinar a posição, e um dispositivo GPS não funciona em túneis porque o sinal por satélite é ausente neste tipo de local.

Existem várias subclassificações do protocolo de roteamento geográfico, bem como diversos protocolos deste tipo como pode ser observado na Figura 3.

O roteamento geográfico pode ser subdividido em DTN (*Delay Tolerant Network*), Não-DTN ou híbrido (DTN/Não-DTN). Além disso, o protocolo Não-DTN ainda possui algumas subclassificações mais específicas para resolver determinados problemas de rota.



**Figura 3: Classificação dos protocolos de roteamento geográficos**

**Fonte: Autoria própria.**

### 3.2.1.2.1 DTN

A Rede Tolerante a Atraso (DTN) é uma subclassificação do roteamento geográfico que aceita um certo atraso antes de encaminhar os pacotes.

De acordo com Lee et al. (2010), como ocorrem desconexões frequentes nas VANETs, as DTNs permitem que os nós armazenem os pacotes enquanto não há contato com outros nós, para enviá-los a certa distância até que haja a união com outros nós. Este procedimento é conhecido como estratégia de armazenar-e-encaminhar (*carry-and-forward*).

Os nós DTN funcionam de modo intermitente, de tal modo que armazenam as mensagens e depois as encaminham quando os enlaces ficam disponíveis (TANENBAUM ANDREW S.; WETHERALL, 2011).

Alguns exemplos de DTN são os protocolos de roteamento veicular VADD (*Vehicle-Assisted Data Delivery*) e GeOpps (*Geographical Opportunistic routing*) confirmados por (LEE et al., 2010; BERNSEN; MANIVANNAN, 2009) e os protocolos MOVE (*Motion Vector*), SADV (*Static-Node Assisted Adaptive Routing Protocol*) e MaxProp apresentados por (BERNSEN; MANIVANNAN, 2009).



### 3.2.1.2.2 NÃO-DTN

Os tipos de protocolos de roteamento geográficos não-DTN não consideram conectividade intermitente e só são recomendados em VANETs densamente povoadas enquanto os tipos de protocolos de roteamento geográficos DTN não consideram desconectividade (LEE et al., 2010).

Os protocolos de roteamento não-DTN tem sua própria estratégia de recuperação de falhas de acordo com subdivisões específicas para cada caso.

Por isso as definições dos protocolos não-DTN podem ser melhor explicadas conforme suas subclassificações como Sinalizador (*Beacon*), Não-Sinalizador (*Non-Beacon*) ou híbrido (*Beacon/Non-Beacon*).

#### a) Sinalizador (*Beacon*)

Como o próprio nome indica, este tipo de protocolo precisa enviar um sinal (*beacon*) de tempos em tempos para verificar a existência e posição atualizada do nó. De acordo com (PAUL et al., 2011a), se um nó receptor não recebe um *beacon*, após um período de tempo, a rota correspondente a este *beacon* será removida da tabela vizinha, considerando que a mesma está inacessível por não enviar a mensagem no tempo previsto.

O protocolo de roteamento baseado em *beacon* ainda pode ser classificado como Sobreposto (*Overlay*) ou Não-Sobreposto (*Non-Overlay*).

#### b) Não-Sinalizador (*Non-Beacon*)

O protocolo não-sinalizador é um protocolo de roteamento geográfico que não exige a transmissão de *beacons*. Neste protocolo os pacotes de dados são transmitidos para todos os vizinhos diretos (vizinhos de um salto), os quais decidem se devem ou não encaminhar o pacote. O transmissor atual é selecionado por um processo de contenção baseado num contador de tempo (temporizador) distribuído que permite ao nó mais adequado encaminhar o pacote e descartar outros transmissores potenciais. Além disso, os receptores dos dados difundidos devem comparar a sua distância para o destino com a distância do último salto para o destino e, quanto maior a diferença mais curto é o temporizador (LEE et al., 2010). O principal exemplo deste protocolo é o CBF (*Contention-Based Forwarding*).

### c) Híbrido (*Beacon/Non-Beacon*)

O protocolo híbrido (*Beacon/Non-Beacon*) é um protocolo de roteamento geográfico que apresenta algumas características do protocolo Não-Sinalizador, mas difere-se deste em três aspectos principais. Primeiramente, ao invés de escolher o próximo nó de encaminhamento que faz o melhor percurso para o destino, ele seleciona o próximo nó de encaminhamento que tem o melhor progresso para um nó de destino. Depois, ao contrário do protocolo de Encaminhamento Baseado em Contenção (CBF), o protocolo híbrido ainda tem necessidade de transmissão de *beacons*, os quais são utilizados para escolher o nó de destino. Ele usa uma nova maneira de escolher o conjunto de nós que são candidatos para o próximo nó de encaminhamento. O conjunto é escolhido de forma que todos os nós podem ouvir uns aos outros (sem terminais ocultos) e progredir em direção ao nó de destino. Por fim, este protocolo utiliza a decisão de roteamento de ambos os modos, ganancioso e de recuperação (LEE et al., 2010).

O roteamento híbrido (*Beacon/Non-Beacon*) tem como principal exemplo o protocolo TO-GO (*TOpology-assist Geo-Opportunistic Routing*).

#### 3.2.1.2.3 HÍBRIDO (DTN/NÃO-DTN)

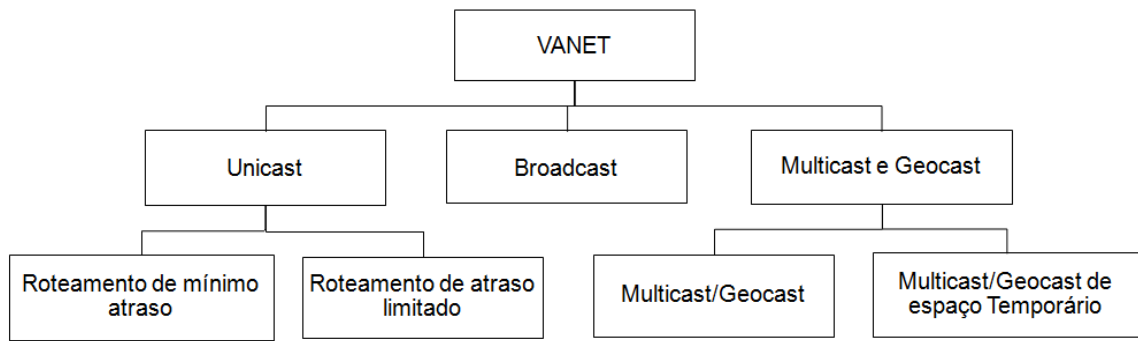
Tipos de protocolos de roteamento geográficos híbridos combinam protocolos de roteamento não-DTN e DTN para explorar a conectividade de rede parcial (LEE et al., 2010).

Um dos exemplos de protocolo híbrido (DTN/Não-DTN) é o GeoDTN+Nav que possui três modos diferentes: o modo ganancioso, modo de perímetro e modo DTN (LEE et al., 2010).

### 3.2.2 CLASSIFICAÇÃO DOS PROTOCOLOS DE ROTEAMENTO PELO TIPO DE PROPAGAÇÃO DOS PACOTES DE ROTEAMENTO

Cada protocolo de roteamento tem uma forma de propagar seus pacotes de roteamento. Essas formas de propagação podem ser classificadas em *unicast*, *broadcast* ou *multicast/geocast* como pode ser observado na Figura 4.

A seguir é explicado o funcionamento de cada um dos protocolos VANET classificados pelo tipo de propagação dos pacotes de roteamento.

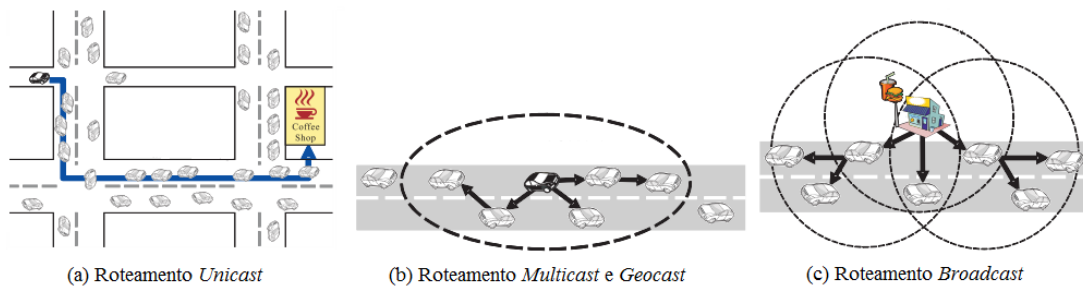


**Figura 4: Classificação dos protocolos de roteamento pelo tipo de propagação dos pacotes de roteamento**

**Fonte: Adaptado de Lin et al. (2010).**

### 3.2.2.1 PROTOCOLO DE ROTEAMENTO *UNICAST*

De acordo com Lin et al. (2010) o principal objetivo do roteamento *unicast* em VANET é a transmissão de dados a partir de uma única origem para um único destino através de transmissão multissalto sem fio (*multi-hop wireless*) ou da técnica *carry-and-forward*. A Figura 5 permite visualizar a diferença entre o roteamento *unicast*, *broadcast* e *multicast/geocast*.



**Figura 5: Comparação entre os protocolos de roteamento VANET pela forma de propagação**

**Fonte: Adaptado de Lin et al. (2010).**

Esta técnica de transmissão multissalto sem fio é conhecida como roteamento multissalto, e os veículos intermediários de uma rota devem transmitir os dados o mais rápido possível da origem para o destino (LIN et al., 2010).

Os autores Lin et al. (2010) explicam que duas categorias de protocolo de roteamento *unicast* são classificadas, o protocolo de roteamento de mínimo atraso (*Min-Delay Routing Protocol*) e o protocolo de roteamento de atraso limitado (*Delay-Bounded Routing Protocol*).

### **3.2.2.1.1 PROTOCOLO DE ROTEAMENTO DE MÍNIMO ATRASO (*MIN-DELAY ROUTING PROTOCOL*)**

O Protocolo de roteamento Min-Delay procura minimizar o valor do tempo de entrega da origem ao destino. O objetivo do Protocolo de Roteamento de Mínimo Atraso é transmitir pacotes de dados para o destino o mais rápido possível. Desta forma, o tempo de atraso de transmissão é a maior preocupação e geralmente o caminho mais curto é adotado. No entanto, um dos problemas deste protocolo é que o caminho mais curto de roteamento pode não ser mais rápido com relação ao tempo de atraso mínimo em VANETs (LIN et al., 2010).

Alguns exemplos de protocolos de Mínimo Atraso são: GPCR, VADD, CAR (*Connectivity-Aware Routing*), DIR (*Diagonal-Intersection-Based Routing Protocol*), ROMSGP (*Receive on Most Stable Group-Path*), Roteamento Confiável (*Reliable Routing*) e GVGrid, conforme apresentado por (LIN et al., 2010).

### **3.2.2.1.2 PROTOCOLO DE ROTEAMENTO DE ATRASO LIMITADO (*DELAY-BOUNDED ROUTING PROTOCOL*)**

O Protocolo de Roteamento de Atraso Limitado (SKORDYLIS; TRIGONI, 2008) tenta manter um baixo nível de utilização do canal através do roteamento com atraso de tempo limitado.

De acordo com Lin et al. (2010), esse protocolo fornece um sistema de roteamento que satisfaz requisitos de atraso definidos pelo usuário e, ao mesmo tempo mantém um baixo nível de utilização do canal. Além disso, ele centra-se no desenvolvimento de esquemas de *carry-and-forward* que tentam entregar os dados de veículos para o ponto de acesso de infraestrutura estática em um ambiente urbano.

Dois exemplos deste protocolo são o D-Greedy (*Delay-bounded Greedy Forwarding*) e o D-MinCost (*Delay-bounded Min-Cost Forwarding*) que avaliam as informações de trânsito e o atraso limitado de tempo para optar cuidadosamente entre as estratégias de dados ponderados e de encaminhamento multissalto para minimizar a sobrecarga de comunicação, desde que satisfaçam às restrições impostas pelo atraso da aplicação (LIN et al., 2010).

### 3.2.2.2 PROTOCOLO DE ROTEAMENTO MULTICAST E GEOCAST

Os protocolos *multicast* são definidos pela entrega de múltiplos pacotes de um único veículo de origem para todos os membros do roteamento *multicast* de comunicação multissalto (Figura 5.b). Enquanto o roteamento *geocast* consiste na entrega de um pacote a uma região geográfica específica, no qual os veículos localizados nesta região devem receber e encaminhar o pacote *geocast*, caso contrário, o pacote é descartado (LIN et al., 2010).

Alguns testes foram realizados com os protocolos *multicast/geocast* em uma VANET por (JOSHI et al., 2007; CHEN et al., 2010). De acordo com a propriedade da região geográfica, os resultados existentes podem ser classificados em protocolos *multicast/geocast* e protocolos de roteamento *multicast/geocast* de espaço temporário (*Spatiotemporary Multicast/Geocast Routing Protocol*) (LIN et al., 2010).

### 3.2.2.3 PROTOCOLOS DE ROTEAMENTO COM BASE EM DIFUSÃO (*BROADCAST BASED ROUTING PROTOCOLS*)

Os autores Lin et al. (2010) informam que o protocolo de *broadcast* é utilizado quando um veículo de origem transmite mensagem para todos os outros veículos na rede (Figura 5.c). O autor Nagaraj e Dhamal (2012) acrescenta que o protocolo de roteamento com base em difusão é frequentemente usado em VANET para compartilhamento, tráfego, meteorologia e emergência, condições de estrada entre veículos e entrega de propagandas e anúncios.

*Broadcast* é o termo usado em redes de computadores quando pacotes são enviados para todos os nós da rede, através de inundação de mensagens. Ao contrário do *multicast* que escolhe um grupo inferior a todos os nós de uma rede, o *broadcast* envia as mensagens para todos os nós participantes da rede. Isto pode ser útil quando realmente há a necessidade de enviar informações para todos os nós. Caso contrário é melhor usar mensagens de *multicast* ou *unicast*, pois o *broadcast* compromete a largura de banda enviando mensagens a endereços desnecessários. Por esse motivo, o autor Nagaraj e Dhamal (2012) explica que estes protocolos têm melhor desempenho para VANETs com pequeno número de nós.

## 3.3 CAMADA DE TRANSPORTE

Na camada de transporte os autores Khorashadi et al. (2007) informam que o ambiente VANET é muito dinâmico e interativo, o que resulta em desempenho imprevisível com o uso do protocolo TCP (*Transmission Control Protocol*). Ele complementa ainda que o desempenho

do TCP em redes sem fio multissalto pode ser muito pobre, devido aos protocolos de controle de congestionamento subjacentes.

Devido aos problemas enfrentados pelo TCP em redes VANET, diversas variações deste protocolo são apresentados como TCP-F (*TCP-Failure*) (CHANDRAN et al., 2001), TCP-ELFN (*TCP-Explicit Link Failure Notification*) (HOLLAND; VAIDYA, 2002), TCP-Bus (KIM et al., 2000), ATCP (*Ad Hoc TCP*) e *Split-TCP* (KOPPARTY et al., 2002), conforme demonstra (TOOR et al., 2008).

Além disso, novamente os autores Khorashadi et al. (2007) explicam que muitas aplicações de informação e entretenimento, como áudio e vídeo usam o protocolo de transporte UDP (*User Datagram Protocol*). Por isso, o protocolo UDP não costuma apresentar problemas, mas como existem dificuldades com o protocolo TCP, algumas modificações para criar uma variação do UDP confiável são apresentadas como o RB-UDP (*Reliable Blast-UDP*) (HE et al., 2002).

Apesar das pesquisas com variações do UDP e TCP, novos protocolos da camada de transporte foram criados exclusivamente para VANETs. De acordo com Toor et al. (2008) outras soluções foram também propostas para construir um novo protocolo da camada de transporte a partir do zero, como é o caso do ACTP (*Application-Controlled Transport Protocol*) e do ATP (*Ad Hoc Transport Protocol*).

Outro exemplo de protocolo da camada de transporte criado especialmente para VANETs é o VTP (*Vehicular Transport Protocol*) apresentado por (SCHMILZ et al., 2006; SCHMITZ, 2007). De acordo com Schmilz et al. (2006), o protocolo VTP tem como principais objetivos: maximizar a taxa de transferência de uma conexão, preservar a equidade entre fluxos rivais, adaptar-se a largura de banda disponível, transmitir dados de forma confiável, lidar com a taxa de perda de pacotes frequente, alta RTT (*Round Trip Times*) fim-a-fim e *jitter*<sup>2</sup> RTT, fornecer dados em ordem e lidar com a alta taxa de pacotes com *jitter* reajustado.

---

<sup>2</sup>*jitter* é um termo utilizado em redes de computadores que indica a variação no atraso da chegada de pacotes. Segundo Tanenbaum Andrew S.; Wetherall (2011) (grifo do autor), “mesmo que os pacotes sejam despachados exatamente com intervalos certos entre eles no transmissor, eles alcançarão o receptor com tempos relativamente diferente. Essa variação no atraso é chamada de *jitter*.”

## 4 FERRAMENTAS DE SIMULAÇÃO PARA VANETS

Para realizar experimentos com redes VANET, três tipos de ferramentas podem ser utilizados: Simuladores de rede, geradores de mobilidade (ou geradores de tráfego de veículos) e simuladores VANET. Na realidade, apenas duas formas de simulação são possíveis, uma integrando os geradores de mobilidade com os simuladores de rede e a outra utilizando diretamente os simuladores VANET.

### 4.1 SIMULADORES DE REDE

Há vários simuladores de rede usados em pesquisas de protocolos de comunicação que também podem ser aproveitados em sistemas de comunicação VANET. Neste trabalho, serão apresentados os simuladores de rede considerados mais promissores para VANET, dos quais quatro deles são citados por Martinez et al. (2011). E, de acordo com (KAMOLTHAM et al., 2012), o ns-3 também se encaixa entre os simuladores mais importantes, quando afirmam que este último tem mais capacidade do que o ns-2 e fornece tanto a capacidade de simulação quanto de emulação.

Para um melhor entendimento, em seguida é informada a ideia central destes simuladores de rede.

- ns-2 (*Network Simulator* versão 2)

De acordo com Martinez et al. (2011) o ns-2 é um simulador de eventos discretos desenvolvido pelo grupo VINT (*Virtual InterNetwork Testbed*) como projeto de pesquisa na Universidade da Califórnia, em Berkeley.

Os autores Martinez et al. (2011) informam também que o simulador foi estendido pelo grupo de pesquisa do Monarch (MONARCH, 2013) na *Carnegie Mellon University* para incluir mobilidade dos nós, uma camada física realista com um modelo de propagação de rádio, interfaces de rede de rádio, e o protocolo MAC IEEE 802.11 com DCF (*Distributed Coordination Function*).

O ns foi criado com o objetivo de fornecer um apoio substancial para a simulação de TCP, roteamento e protocolos de *multicast* para redes com e sem fio (local e satélite) (ns-2, 2013).

O ns-2 é um conjunto de mudanças importantes do ns versão 1, que foi bastante estudado em laboratório em meados da década de 90. Em 1996 surgiu a versão 2 do ns com grandes mudanças arquitetônicas e uma API (*Application Programming Interface*) Tcl (*Tool Command Language*) mais limpa, de tal modo que os desenvolvedores podiam agora mudá-lo mais facilmente, enquanto mantinham uma compatibilidade com versões anteriores. Além disso, com o OTcl (*Object oriented extension of Tcl*) tornou-se mais fácil estender uma classe com novos métodos, seja em C++ ou no Tcl (HISTORY, 2013).

O ns-2 foi um dos simuladores de redes mais usados ou o mais usado pela comunidade científica e talvez ainda seja. Pois houve uma grande contribuição para esta ferramenta por parte dos pesquisadores durante muitos anos. Mas consultando a história de sua evolução em (HISTORY, 2013) é possível perceber que a última atualização ocorreu em novembro de 2011 com a versão 2.35.

Apesar disso, o ns-2 ainda é bastante utilizado por apresentar-se bem estável e com vários módulos auxiliares já desenvolvidos, de tal forma que facilita muito as pesquisas na área de redes e telecomunicações. Entretanto, como as contribuições estão migrando para o ns-3, a tendência é que este último substitua de vez o uso do ns-2.

- ns-3 (*Network Simulator* versão 3)

De acordo com (NS-3 RASTREAMENTO, 2013) o ns-3 é baseado em eventos discretos, desenvolvido especialmente para pesquisa e uso educacional, cujo projeto iniciou-se em 2006 e possui seu código aberto.

O ns-3 é uma evolução natural do ns-2, mas não é uma extensão do mesmo. Ele é um novo simulador que foi construído totalmente do zero. Sendo assim, apesar de ambos serem escritos em C++, o ns-3 é totalmente novo e não suporta as APIs da versão anterior (NS-3 RASTREAMENTO, 2013). Mesmo assim, algumas funcionalidades do ns-2 já foram exportadas para o ns-3, e seu projeto ainda será mantido enquanto o ns-3 estiver em fase de desenvolvimento. Além disso, formas de integração e transição estão em estudo para melhorar a importação de utilidades pelo ns-3.



## 4.2 GERADORES DE MOBILIDADE

Individualmente, os simuladores de rede não são recomendáveis para realizar experimentos científicos de VANET quando trabalham sem o auxílio de geradores de mobilidade. Acontece que os módulos de mobilidade destas ferramentas não conseguem refletir um tráfego de veículos de forma adequada, reduzindo as chances de extração de resultados mais eficazes.

Para realizar as simulações VANET com simuladores de rede, primeiramente são feitas simulações de tráfego dos veículos pelos geradores de mobilidade. Estes, por sua vez, permitem extrair um arquivo de mobilidade com informações sobre a identificação, o tempo da simulação, a posição e a velocidade dos veículos (nós) no decorrer do tempo. Com isso, é possível importar estas informações através dos simuladores de rede que, então podem realizar as simulações VANET com ambientes de mobilidade mais realísticos e conseqüentemente, resultados mais precisos.

De acordo com Martinez et al. (2011), atualmente existem vários ambientes de software de simulação e eles são capazes de gerar arquivos de rastreamento refletindo os movimentos dos veículos. Estes geradores de tráfego de veículos permitem configurar diferentes condições de trânsito e fornecem simulações de mobilidade bem próximas a um ambiente real, com obstáculos como cruzamentos e semáforos. Além de definir veículos diferenciados como caminhões, carros, motos e ônibus.

É importante observar que os arquivos de mobilidade que podem ser importados pelo ns-2, conseqüentemente também podem ser importados pelo ns-3, pois foi implementado um módulo no ns-3 capaz de importar arquivos de mobilidade gerados para o ns-2.

A seguir são informadas as características fundamentais de alguns dos geradores de mobilidade mais conhecidos conforme indicado por Martinez et al. (2011).

- *VanetMobiSim (Vehicular Ad Hoc Networks Mobility Simulator)*

O Simulator de Mobilidade de Rede *Ad Hoc* Veicular (VanetMobiSim) é um conjunto de extensões para CanuMobiSim, uma estrutura flexível para modelar a mobilidade do usuário utilizado pelo CANU (*Communication in Ad Hoc Networks for Ubiquitous Computing*) (CANUMOBISIM, 2013).

O CanuMobiSim é baseado em Java e pode gerar traços de movimento em diferentes formatos, apoiando diferentes ferramentas de simulação/emulação para redes móveis como NS2, GloMoSim e QualNet. O CanuMobiSim originalmente inclui analisadores

de mapas no padrão GDF (*Geographical Data Files*) e fornece implementações de vários modelos de mobilidade aleatórios, bem como modelos veiculares físicos e dinâmicos (VANETMOBISIM, 2013).

- SUMO (*Simulation of Urban MOBility*)

Segundo Behrisch et al. (2011), o SUMO é um simulador de tráfego puramente microscópico. Cada veículo é definido pelo menos por um identificador (nome), o instante de tempo de saída, e o percurso do veículo através da rede. Se for necessário, cada veículo pode ser descrito de forma mais detalhada, com informações como as propriedades de saída e chegada, a faixa usada na via, a velocidade ou a posição gerada. Além disso, pode ser atribuído um tipo que descreve as propriedades físicas do veículo e as variáveis do modelo de mobilidade usado.

De acordo com Martinez et al. (2011) as principais características do SUMO incluem: movimento do veículo livre de colisão, diferentes tipos de veículos, roteamento de veículo, ruas multifaixas com mudança de faixa, regras de direito de passagem em cruzamentos, hierarquia de tipos de cruzamento, uma GUI (*Graphical User Interface*) OpenGL (*Open Graphics Library*) e roteamento dinâmico.

O SUMO pode gerenciar ambientes de grande porte, com 10000 ruas, por exemplo, e pode importar muitos formatos de rede como Visum, Vissim, ArcView, ou descrições XML (*eXtensible Markup Language*). No entanto, uma vez que o SUMO é um gerador de tráfego puro, seus arquivos de mobilidade gerados não podem ser utilizados diretamente pelos simuladores de rede disponíveis, o que é um defeito grave, conforme explica o autor Martinez et al. (2011).

- MOVE (*MOBility model generator for VEhicular networks*)

O MOVE é uma ferramenta desenvolvida para facilitar a criação rápida de modelos de mobilidade para simulações VANET realistas. Ele é construído em cima de um simulador microtráfego SUMO de código fonte aberto e gera um arquivo de saída de rastreamento de mobilidade que contém informações dos movimentos de veículos que podem ser imediatamente utilizados por ferramentas de simulação populares, como o ns-2 ou Qualnet. Além disso, o MOVE também fornece um conjunto de GUIs que permitem ao usuário gerar cenários realistas de simulação rapidamente, sem o incômodo de escrever os *scripts* de simulação, bem como aprender sobre os detalhes internos do simulador (KARNADI et al., 2007).

- STRAW (*STreet RAndom Waypoint*)

O STRAW fornece resultados de simulação mais precisos usando um modelo de mobilidade com base na operação do tráfego de veículos reais em cidades norte-americanas. A implementação atual do STRAW é escrita para os simuladores de eventos discretos altamente eficientes conhecidos como JIST/SWANS (*Java in Simulation Time/Scalable Wireless Ad hoc Network Simulator*), tem o código fonte aberto e é disponível gratuitamente (STRAW, 2013). Para Martinez et al. (2011) um modelo de mobilidade mais realista com o nível de detalhe apropriado para redes veiculares é fundamental para uma simulação precisa. O gerador de mobilidade STRAW restringe o movimento do nó em ruas definidas por mapas de cidades dos Estados Unidos e limita sua mobilidade de acordo com o congestionamento de veículos e mecanismos de controle simplificados de trânsito.

- FreeSim (*Freeway Simulator*)

FreeSim é um simulador de tráfego macroscópico e microscópico, totalmente personalizável, que permite aos sistemas de autoestrada serem facilmente representados e carregados em um simulador. FreeSim está licenciado sob a GNU GPL (*General Public License*), e o código-fonte está disponível para download. Além disso, os veículos em FreeSim podem se comunicar com o sistema de monitoramento do tráfego nas autoestradas, o que torna o FreeSim ideal para a simulação em ITS (FREESIM, 2013).

- CityMob v.2

Os autores Martinez et al. (2008) explicam que CityMob é um gerador de mobilidade padrão, concebido especialmente para investigar os diferentes modelos de mobilidade em VANETs, e seu impacto no desempenho da comunicação entre veículos. O CityMob cria cenários de mobilidade urbana e simula carros danificados usando a rede para enviar informações para outros veículos, tentando evitar acidentes ou engarrafamentos e pode implementar modelos como o SM (*Simple Model*), MM (*Manhattan Model*), e DM (*Downtown Model*).

Conforme Martinez et al. (2008) informam, no modelo de mobilidade SM não há mudança de direção e nem semáforos. No modelo MM, as ruas estão dispostas numa rede estilo Manhattan, com um tamanho de bloco uniforme em toda a área de simulação. Todas as ruas são de mão dupla, com faixas em ambos os sentidos e os movimentos dos carros são limitados por essas pistas. O modelo DM também adiciona uma densidade de tráfego no modelo MM, de forma semelhante a uma cidade real, em que o tráfego não é uniformemente distribuído. Assim, como normalmente existem zonas de maior

concentração de veículos no centro das cidades, os veículos devem se mover mais lentamente nestas áreas do que na periferia.

### 4.3 SIMULADORES VANET

Conforme mencionado anteriormente, os geradores de mobilidade e os simuladores de rede são configurados em duas ferramentas de simulação de forma separada. Para Martinez et al. (2011), existe uma clara necessidade de uma mobilidade integrada ao simulador de rede, a fim de avaliar de forma eficaz o desempenho de sistemas de comunicação entre veículos. Sendo assim, alguns simuladores VANET, apresentados pelo autor são descritos a seguir.

- GrooveNet

GrooveNet é um simulador híbrido que permite a comunicação entre veículos simulados e veículos reais, facilitando o entendimento do protocolo e do veículo incluído ao modelar a comunicação interveicular dentro de uma topografia com base em mapa de ruas reais (MARTINEZ et al., 2011).

De acordo com os autores Mangharam et al. (2006), o GrooveNet é um simulador de rede veicular baseado em mapa de rua. Formado por eventos modulares com modelo de interfaces bem definido, suporta vários veículos, viagem e modelos de mobilidade, além de uma variedade de modelos de rede da camada de enlace e física. Sua interface gráfica facilita simulações com milhares de veículos em qualquer lugar dos EUA. Suporta três tipos de nós simulados: veículos capazes de multissaltos de dados por um ou mais canais de DSRC, nós de infraestrutura fixa e *gateways* móveis capazes de comunicação V2V e V2I.

O GrooveNet também suporta vários tipos de mensagens, como mensagens de GPS, que são inundadas na rede periodicamente para informar os vizinhos sobre a posição atual do veículo, mensagens de veículos de emergência e eventos de alerta com prioridades. Suporta múltiplas interfaces de rede para comunicação V2V e V2I real. Além disso, ele é capaz de aceitar simulações híbridas (ou seja, comunicação entre veículos simulados e veículos reais na estrada) onde a posição do veículo simulado, a direção e as mensagens, são difundidas na rede através da interface celular a partir de um ou mais nós de infraestrutura. E por fim, o GrooveNet ainda pode se conectar ao computador de bordo do veículo e ler códigos de diagnóstico como a desaceleração súbita, frenagem, *air bag* e sinais do sistema de trava antibloqueio que podem desencadear avisos ou mensagens de alerta como afirmam Mangharam et al. (2006).

- NCTUns (*National Chiao Tung University Network Simulator*)

O NCTUns é um simulador/emulador de rede de alta fidelidade e extensível capaz de simular vários protocolos usados tanto em redes IP cabeadas quanto em redes sem fio. Sendo assim, como se trata de um emulador também é útil para testar os dispositivos de rede da vida real. Além disso, a tecnologia do núcleo do NCTUns é baseada em uma nova metodologia de simulação que oferece muitas vantagens exclusivas que os simuladores de rede tradicionais não podem fornecer (NCTUNS, 2013).

De acordo com Martinez et al. (2011) o NCTUns usa pilha TCP/IP Linux para gerar resultados de simulação de alta fidelidade e pode executar qualquer programa UNIX da vida real em um nó simulado sem quaisquer modificações. Mas sua principal desvantagem é que o NCTUns requer distribuição Linux Fedora 9, o que representa um problema para a maioria dos pesquisadores VANET, limitando sua ampla utilização.

- MobiREAL

O MobiREAL é um novo simulador de rede com dispositivos móveis que pode simular a mobilidade realista dos seres humanos e automóveis, e permite a mudança de seu comportamento de acordo com determinado contexto de aplicação. Ele possibilita obter uma avaliação mais detalhada do desempenho de aplicações de rede, protocolos de roteamento e infraestrutura que os simuladores existentes não conseguem avaliar bem (MOBIREAL, 2013).

O simulador MobiREAL é composto de duas partes chamadas: simulador de comportamento MobiREAL, que simula o comportamento dos nós móveis e; o simulador de rede MobiREAL, que simula a troca de dados entre os nós móveis. Os simuladores de comportamento e de rede são dois programas independentes que trocam periodicamente os dados necessários através de um canal TCP (KONISHI et al., 2005).

O simulador de rede MobiREAL, desenvolvido em C++, pode facilmente descrever a mobilidade dos nós. Seus desenvolvedores adotaram um modelo baseado em regras probabilísticas para descrever o comportamento de nós móveis frequentemente usados em modelagem cognitiva do comportamento humano. Isto permite descrever como os nós móveis mudam seus destinos, rotas e direções/velocidades com base em suas posições, ambiente (obstáculos e nós vizinhos), informações obtidas a partir de aplicações e assim por diante (MOBIREAL, 2013).

Os autores Martinez et al. (2011) explicam que o MobiREAL simula MANETs usando o suporte de mobilidade encontrado no GTNetS (*Georgia TechNetwork Simulator*). Além disso, o animador MobiREAL visualiza dinamicamente o movimento dos nós, os estados

de conectividade e transmissão de pacotes, possibilitando uma melhor compreensão dos resultados de simulação de forma intuitiva.

Apesar do MobiREAL permitir o uso de outros simuladores de tráfego para fornecer mobilidade veicular, Martinez et al. (2011) informam que ele utiliza um simulador de tráfego chamado NETSTREAM, que foi desenvolvido pela Toyota. Por ser um software proprietário sua principal desvantagem é que os usuários não podem acessar e modificar esta parte do simulador, limitando sua ampla utilização.

#### 4.4 FERRAMENTAS DE SIMULAÇÃO UTILIZADAS

A extensão VanetMobiSim apresenta novos modelos de movimento realista de automóveis, tanto a nível macroscópico e microscópico. A nível macroscópico, o VanetMobiSim pode importar mapas do banco de dados geográfico TIGER (*Topologically Integrated Geographic Encoding and Referencing Geodatabases*) (TIGER, 2013), ou pode gerá-los aleatoriamente usando diagramas de Voronoi (Separação de uma região em sub-regiões). Além disso, ele adiciona suporte para estradas multifaixas, fluxos direcionais separados, restrições de velocidade diferenciadas e sinais de trânsito nos cruzamentos. A nível microscópico, o VanetMobiSim implementa novos modelos de mobilidade, proporcionando uma interação realística tanto para V2V quanto para V2I. De acordo com estes modelos, os veículos regulam a sua velocidade dependendo dos carros das proximidades, ultrapassam uns aos outros e agem conforme a presença dos sinais de trânsito nos cruzamentos (VANETMOBISIM, 2013).

Os autores Harri e Fiore (2006) afirmam que o principal componente do modelo orientado veicular é o suporte de um modelo de mobilidade de nível microscópico chamado IDM\_IM (*Intelligent Driving Model with Intersection Management*). Além disso, no IDM\_LC (*Intelligent Driving Model with Lane Changing*) também foi incluído um modelo de ultrapassagens (MOBIL - *Minimizing overall braking induced by lane change*) (KESTING et al., 2007), que interage com o IDM\_IM para gerenciar mudanças de faixa e acelerações e desacelerações do veículo.

Ao invés de usar modelos de mobilidade MANET como o *RandomWayPoint* usado por (PAUL et al., 2011b), ou o *Gipps* usado por (CORREIA et al., 2011), neste trabalho optou-se pelo VanetMobiSim. Assim, para desfrutar do movimento realista dos veículos no VanetMobiSim, foi escolhido o modelo de mobilidade inteligente IDM\_LC e MOBIL.

Desta forma, o VanetMobiSim possui uma forma de configuração menos complexa

e movimentos com aceleração, frenagem, ultrapassagem e semáforos que possibilitam analisar resultados mais próximos de um ambiente real. A Tabela 2 permite observar uma comparação entre os geradores de mobilidade veicular, onde verifica-se características úteis do VanetMobiSim.

**Tabela 2: Comparação entre os geradores de mobilidade veicular**

	VanetMobiSim	SUMO	MOVE	STRAW	FreeSim	CityMob
<b>SOFTWARE</b>						
Portabilidade	S	S	S	S	S	S
Gratuito	S	S	S	S	S	S
Código aberto	S	S	S	S	S	S
GUI	S	S	S	S	S	S
Exemplos	S	S	S	-	S	N
Facilidade de uso	moderado	difícil	moderado	moderado	fácil	fácil
Facilidade de configuração	moderado	moderado	fácil	moderado	fácil	fácil
<b>MAPAS</b>						
Mapas reais	S	S	S	S	S	N
Mapas definidos pelo usuário	S	S	S	-	N	N
Aleatório	S	S	S	N	N	S
Voronoi	S	N	N	N	N	N
<b>MOBILIDADE</b>						
<i>Random waypoint</i>	S	S	S	N	N	S
Manhattan	N	S	S	N	N	S
<b>MODELO DE TRÁFEGO</b>						
Macroscópio	S	N	N	N	S	N
Microscópio	S	S	S	S	S	S
Rodovia multifaixas	S	S	S	S	-	S
Mudança de faixas	S	S	S	S	-	S
Fluxo direcional separado	S	S	S	S	-	S
Restrição de velocidade	S	S	S	S	S	S
Semáforos	S	S	S	S	-	S
Gerenciamento de interseções	S	S	S	-	-	S
Critério de ultrapassagem	S	-	-	-	-	N
Diferentes tipos de veículos	S	S	S	-	N	S
Cálculo de rota	S	S	S	S	S	N
<b>SUPORTE A ARQUIVOS DE TRACES</b>						
<i>traces</i> de suporte a ns2	S	N	S	N	N	S
Suporte a <i>traces</i> baseados em XML	S	N	N	N	N	N
Importa outros formatos	S	S	S	N	N	N

**Fonte: Adaptado de Martinez et al. (2011)**

Através da Tabela 2 é possível inferir que apesar do SUMO, MOVE e CityMob trabalharem com o Modelo Manhattan, deve-se destacar que segundo Härrri et al. (2009), o modelo Manhattan mantém a velocidade máxima e quando o veículo encontra um obstáculo, pára abruptamente fazendo com que sua velocidade seja igual a zero, e retorna novamente a velocidade máxima quando o obstáculo é retirado. Por outro lado, com o modelo IDM a velocidade reduz e aumenta gradativamente diante de obstáculos, tornando o movimento mais realista.

Outra observação importante com relação a Tabela 2 é que apesar do MOVE ser compatível com o ns-2, ele é pré-configurado com parâmetros do ns-2, tornando-o compatível com este simulador (KARNADI et al., 2007). Porém, o mesmo não é compatível com o ns-3,

necessitando de adaptações ou outros softwares para realizar esta integração.

Para realizar as simulações deste trabalho, os protocolos foram implementados no ns3 versão 3.17 e simulações foram realizadas no ambiente urbano através da importação dos traces gerados pelo VanetMobiSim.

A escolha do ns-3 se justifica porque apesar de não ter todos os modelos do ns-2, possui novas funcionalidades como: capacidade de trabalhar corretamente com nós de rede com múltiplas interfaces de rede, suporte a endereçamento IP, maior consistência com arquiteturas e protocolos da Internet e melhor detalhamento do modelo 802.11, além de outras funcionalidades (NS-3 RASTREAMENTO, 2013).

Em todo o caso, de acordo com (NS-3 RASTREAMENTO, 2013) os modelos do ns-2 podem ser portados para o ns-3 e um guia está em desenvolvimento para facilitar este processo. Além disso, atualmente existem várias frentes de trabalho para o desenvolvimento do simulador e, os desenvolvedores bem como os primeiros usuários acreditam que o ns-3 está pronto para o uso e é uma ótima alternativa para usuários que querem iniciar novos projetos de simulação.



## 5 G-NET: ROTEAMENTO BASEADO EM ALGORITMOS GENÉTICOS

Como as VANETs representam um tipo de rede *ad hoc* móvel e os problemas da rede veicular, conforme já citados, costumam impedir o bom funcionamento de protocolos de roteamento, modificações com algoritmos bio inspirados estão sendo realizadas para adequar estes protocolos a ambiente de rede veicular como pode ser verificado em (GUNASEKAR; HINDUJA, 2014) e (CORREIA et al., 2011).

Seguindo esta premissa, surgiu a motivação em criar um protocolo baseado em algoritmo genético para tentar amenizar as falhas conhecidas das redes veiculares. O protocolo de roteamento proposto neste trabalho foi batizado de *Genetic Network Protocol* (G-Net). Trata-se de um protocolo “*source routing*”, pois esse tipo de protocolo armazena a rota completa da origem até o destino. Sendo assim, os nós intermediários da rota devem ser conhecidos de tal forma que possibilitem a aplicação do algoritmo genético, diferente dos protocolos de roteamento “*hop-by-hop*” que conhecem somente o próximo salto da rota, impossibilitando o cruzamento e mutação entre as rotas descobertas. Desta forma, por ser um protocolo “*source route*”, o protocolo DSR foi escolhido como ponto de partida para o desenvolvimento do G-Net.

O Algoritmo Genético é um tipo de algoritmo bio inspirado que permite resolver problemas em ambientes com mobilidade. Tanomaru (1995) explica que ambientes com mudanças de cenário consistem num tipo de problema avançado de AG. Apesar disso, verifica-se que o AG consegue encontrar soluções suficientemente viáveis nestes tipos de ambiente como observado em (GREFENSTETTE et al., 1992).

Como as VANETs representam um ambiente de alta mobilidade e velocidade, espera-se que a implementação de algoritmo genético em protocolos de roteamento *ad hoc* consiga gerar rotas melhores possibilitando assim, obter maiores taxas médias de entrega de dados na transmissão de pacotes entre os veículos.

Antes de conhecer a proposta é importante, primeiramente, conhecer o funcionamento do protocolo DSR e os fundamentos básicos de Algoritmos Genéticos para aplicação no G-Net.

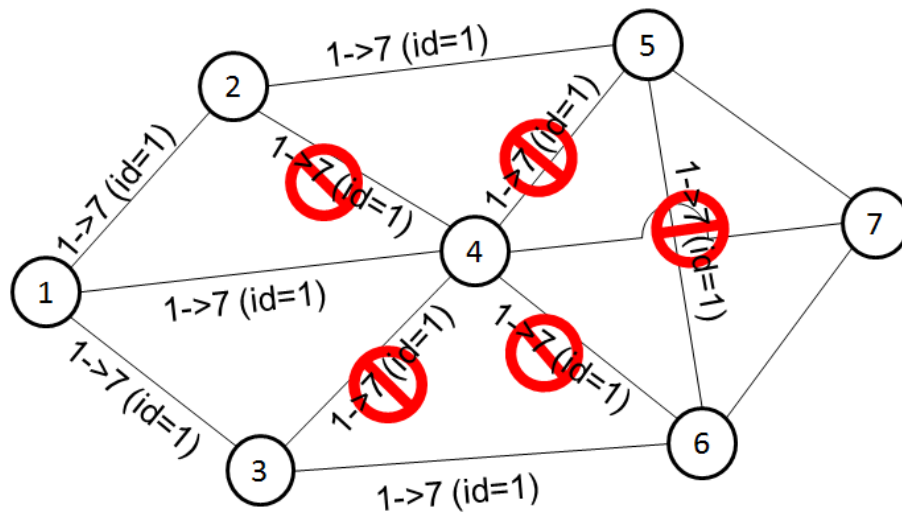
## 5.1 FUNCIONAMENTO DO PROTOCOLO DSR

O DSR é um protocolo de roteamento MANET reativo (JOHNSON et al., 2001). Isto significa que ele só descobre rotas quando precisa enviar dados. Sendo assim, quando um nó A deseja enviar dados para um nó B, o nó A configura o IP de origem (seu próprio endereço IP) e o IP de destino (Endereço IP de B) junto com os dados no cabeçalho IP e armazena o pacote de dados no “*send buffer*”. Enquanto o pacote de dados aguarda no “*send buffer*”, o nó A procura uma rota para B em seu “*route cache*”. Ou seja, ele pega o endereço IP de destino do “*IP HEADER*” e procura uma rota para este destino em seu “*route cache*”. Caso haja uma rota para B, o nó A transfere o pacote de dados para o “*maintenance buffer*”, descarta o pacote do “*send buffer*” e envia o pacote para B através da primeira rota encontrada no “*route cache*”. Caso não exista rota no “*route cache*”, o nó A inicia o processo de descoberta de rota para B, enviando um pacote de controle “*dsr-request*” para *broadcast* até chegar em um nó que conheça a rota para B ou até chegar no próprio destino. Quando o nó B, ou um nó intermediário que conheça sua rota, recebe o pacote “*dsr-request*”, o nó obtém o “*source route*” por onde o “*dsr-request*” passou e devolve um pacote de controle “*dsr-reply*” através da rota reversa.

Como o DSR é um protocolo *multipath*, ele pode descobrir várias rotas de uma origem A para o destino B. Quando o nó A recebe o primeiro “*dsr-reply*” com a rota (“*source route*”) para B, ele armazena esta rota no “*route cache*”, transfere o pacote de dados para o “*maintenance buffer*”, descarta o pacote do “*send buffer*” e envia o pacote para B através desta primeira rota encontrada no “*route cache*”.

Durante o envio de pacotes de dados, eles ficam armazenados no “*maintenance buffer*” e caso haja algum problema em um nó ou *link*, o nó procura a próxima rota do “*route cache*” para continuar enviando estes dados (JOHNSON et al., 2007). Caso a rota seja útil, os dados continuam sendo enviados até o final. Caso nenhuma das outras rotas do “*route cache*” consiga enviar dados para B, o processo de descoberta é acionado novamente até descobrir uma nova rota de A para B. Se o nó A descobre novas rotas para B, os dados são enviados pela primeira rota descoberta e os pacotes continuam no “*maintenance buffer*” para o caso de haver novos problemas na rede. Caso nenhuma rota de A para B seja descoberta até o tempo do “*maintenance buffer time out*” expirar, os dados são descartados.

O DSR possui um mecanismo de proteção de *loops* para evitar que o “*dsr-request*” seja enviado indefinidamente pela rede. Para isso, cada pacote recebe um ID e, caso um nó intermediário já tenha recebido um pacote “*dsr-request*”, ele descarta outros pacotes com o mesmo ID conforme pode ser observado na Figura 6.



**Figura 6: Mecanismo de proteção de *loop* na descoberta de rotas do DSR**

**Fonte: Autoria própria.**

Outra característica do DSR é que ele busca rotas para um destino percorrendo sub-rotas das rotas salvas em seu próprio *route cache*. Se uma rota para um destino esteja contida em uma rota existente, o DSR armazena esta sub-rota como uma nova rota para o destino desejado, utiliza a mesma para enviar os dados e evita a sobrecarga de roteamento com o envio de *broadcast* pela rede para a descoberta de outras rotas.

## 5.2 ALGORITMO GENÉTICO PARA OTIMIZAÇÃO DE ROTAS

Algoritmos Genéticos (AGs) são métodos computacionais de busca baseados nos mecanismos de evolução natural e na genética. São algoritmos que usam o conceito de probabilidade, mas não são “simples” buscas aleatórias. Pelo contrário, eles tentam dirigir a busca para regiões do espaço onde é “provável” que os pontos ótimos estejam presentes. Em AGs, uma população de possíveis soluções para um problema específico evolui de acordo com operadores probabilísticos concebidos a partir de metáforas biológicas, de modo que há uma tendência que os indivíduos representem soluções cada vez melhores a medida que o processo evolutivo continua. Neste contexto, cada solução é representada por uma sequência de símbolos. No caso mais simples, usa-se o alfabeto binário, mas no caso geral tanto o método de representação quanto o alfabeto genético dependem de cada problema. Sendo assim, cada sequência de símbolos corresponde a um cromossomo, e cada símbolo equivale a um gene. Além disso, na maior parte dos AGs assume-se que cada indivíduo seja constituído de um único cromossomo, razão pela qual é comum usarem-se os termos indivíduo e cromossomo

indistintamente em trabalhos científicos e livros-textos (TANOMARU, 1995).

Buriol et al. (2005) explicam que o algoritmo genético (AG) é uma metaheurística de base populacional para otimização combinatória. No AG, uma população é simplesmente um conjunto de soluções possíveis. Soluções em uma população são combinadas (através de cruzamento) e perturbado (por mutação) para produzir uma nova geração de soluções. Quando as soluções são combinadas, os atributos das soluções de maior qualidade têm uma maior probabilidade de ser passada para a próxima geração. Este processo é repetido ao longo de gerações, melhorando a qualidade das soluções da nova população ao longo do tempo.

Desta forma, assim como na relação humana, cada par de indivíduos selecionados para gerarem possíveis soluções são chamados de pais, e conseqüentemente as soluções geradas são chamadas de indivíduos ou cromossomos filhos.

Mitchell (1998) detalha um pouco mais e informa que, em algoritmos genéticos, o termo cromossomo normalmente se refere a uma solução candidata a um problema, muitas vezes codificada como uma cadeia de bits. Os “genes” são pedaços individuais ou blocos curtos de bits adjacentes que codificam um elemento particular do candidato a solução. O cruzamento normalmente consiste em troca de material genético entre dois pais e a mutação consiste de inverter o bit num local escolhido aleatoriamente (ou, para alfabetos maiores, a substituição de um símbolo num local escolhido aleatoriamente com o novo símbolo escolhido aleatoriamente).

Um tipo de estudo avançado de AG é o problema de roteamento automático (PRA). O PRA consiste em encontrar trilhas não-conflitantes unindo terminais distribuídos numa dada região. Um exemplo de PRA é a distribuição de trilhas de placas de circuito. No caso destas placas, o objetivo da evolução é melhorar a qualidade de roteamento, no sentido de eliminar os curtos circuitos e ao mesmo tempo produzir trilhas simples, curtas, e com poucas curvas, de modo a aumentar a confiabilidade do circuito final (TANOMARU, 1995).

Numa analogia as placas de circuito, na proposta do protocolo G-Net, o objetivo da evolução é melhorar a qualidade de roteamento, no sentido de reduzir a latência das rotas (*source routes*) e ao mesmo tempo produzir rotas diferentes, com possíveis melhores latências, que o DSR não é capaz de produzir devido a proteção de *loop* que ele possui (vide item 5.1). Sendo assim, pode-se dizer que a busca por melhores rotas no G-Net trata-se de um tipo de PRA. Além disso, o PRA pertence a classe de problemas de otimização com espaço de busca altamente multimodais, neste caso é mais importante buscar uma solução viável em tempo prático do que a solução ótima absoluta (TANOMARU, 1995).

Outro estudo avançado de AG é a otimização de ambientes dinâmicos ou não-

estacionários. A otimização de funções não-estacionárias é aquela em que ambientes variam com o tempo. Por exemplo, variações climáticas podem fazer com que indivíduos bem adequados a um determinado ambiente tenham menor chance de sobrevivência do que outros indivíduos até então não tão adequados (TANOMARU, 1995). Existem muitos problemas de otimização dinâmica (POD) em aplicações do mundo real. Apesar de Algoritmos Evolucionários (AE) serem bons para resolverem PODs, uma vez que o ambiente muda, é difícil para os indivíduos saltar para novas regiões promissoras do espaço de busca. Entre as abordagens para PODs, esquemas de imigrantes têm se mostrado eficazes para manutenção da diversidade da população durante a execução. Eles funcionam por substituição de uma proporção predefinida da população como migrantes recém-gerados do fim de cada geração (YU et al., 2011).

Sendo assim, indivíduos gerados em uma população migram para outra população afim de manter a diversidade e possíveis “bons” genes para a geração de novos indivíduos. Na literatura pode ser encontrado tanto exemplo de migração de genes para mutação (GREFENSTETTE et al., 1992), como migração de indivíduos em busca de melhores soluções (YU et al., 2011).

Como nas VANETs os veículos representam nós de alta mobilidade e alta velocidade, as variações dos nós (genes) podem fazer com que indivíduos (rotas) considerados melhores num determinado momento, não sejam mais adequados poucos segundos depois. Desta forma, mesmo que hajam muitas gerações, a latência de uma rota pode ser ótima, mas segundos mais tarde a mesma rota (indivíduo) pode ter latência muito ruim em função do deslocamento dos nós móveis. Neste caso, pode-se fazer uma migração de indivíduos de um ambiente de um tempo anterior para um ambiente de um tempo imediatamente posterior, a fim de aproveitar as melhores rotas anteriores como parte da próxima população no decorrer do tempo.

### 5.2.1 CODIFICAÇÃO DO CROMOSSOMO

A codificação do cromossomo é a questão-chave quando se aplica um algoritmo evolucionário. Ela afeta não só os métodos de decodificação e avaliação da aptidão, mas também a realização de seleção, procedimentos de cruzamento e mutação (HUANG; LIU, 2010). O primeiro desafio em usar o AG é definir como os cromossomos irão representar possíveis soluções de problemas. Normalmente os problemas são codificados em bits que representam genes e cadeias de bits representam cromossomos. No caso do G-Net, cada rota completa de uma origem para um destino representa uma possível solução (cromossomo) e sua latência determina se esta rota é boa ou ruim. Sendo assim, para converter as rotas em

cromossomos, cada nó da rede é considerado um gene e o conjunto de nós de uma rota completa representa o cromossomo.

### 5.2.2 FUNÇÃO DE *FITNESS*

Algoritmos Genéticos necessitam da informação do valor de uma função objetivo para cada membro da população, que deve ser um valor não-negativo. A função objetivo dá, para cada indivíduo, uma medida de quão bem adaptado ao ambiente ele está, ou seja, num problema de maximização quanto maior o valor da função objetivo, maiores são as chances do indivíduo sobreviver no ambiente e reproduzir-se, passando parte de seu material genético a gerações posteriores. Em geral, gera-se uma população temporária de indivíduos com probabilidade proporcional à adequabilidade relativa de cada indivíduo na população. Neste processo, indivíduos com alta adequabilidade terão alta probabilidade de serem selecionados e consequentemente, grandes chances de sobreviverem (TANOMARU, 1995).

No G-Net, uma rota é considerada boa se apresenta baixa latência. Como os indivíduos representam as rotas, a baixa latência identifica indivíduos melhor adaptados ao meio. Porém, para uniformidade com a operação do algoritmo genético, foi utilizado o inverso das latências e a função de *Fitness* foi definida conforme apresentado na Equação 1 a seguir:

$$F_{id} = \frac{1}{\sum_j \frac{1}{L_{jd}}}, \quad (1)$$

Onde:

$L_{id}$  = Latência da rota  $i$  para o destino  $d$ .

Trata-se, portanto, de um problema de maximização. Sendo assim, quanto maior o *Fitness*, mais o indivíduo é adequado ao meio naquele determinado momento.

### 5.2.3 SELEÇÃO DE INDIVÍDUOS

Nesta dissertação cada cromossomo é um indivíduo completo e os indivíduos da população inicial são selecionados como pais para que possam gerar indivíduos filhos como possíveis melhores soluções. De acordo com Tanomaru (1995), “não existe um consenso sobre o melhor e mais justo método de seleção...”. Além disso, o autor também explica que o modelo de seleção por torneio estocástico, consiste em extrair aleatoriamente um par de indivíduos por

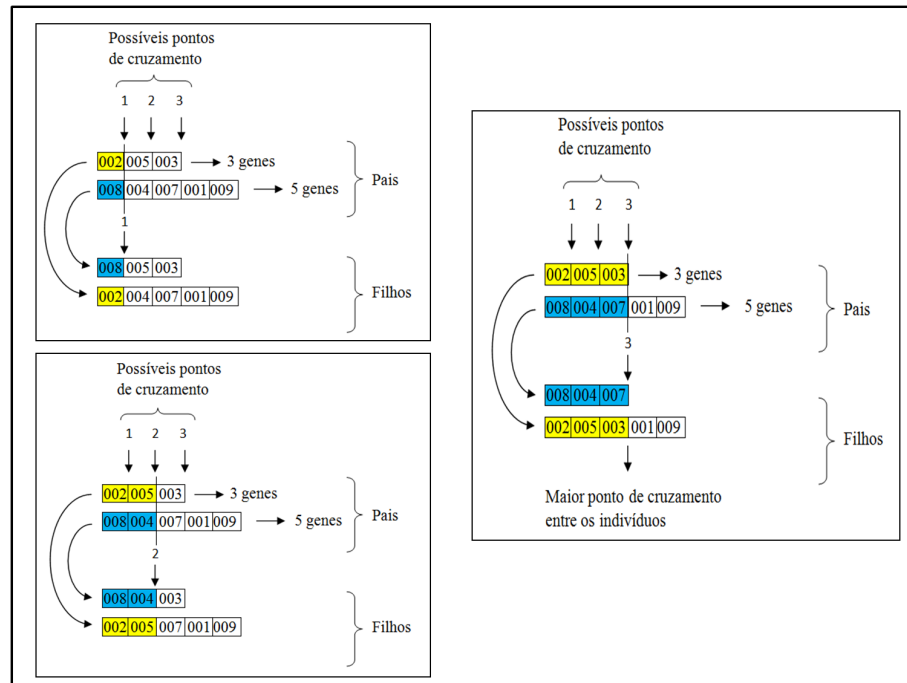
vez, mas apenas o indivíduo com o melhor *Fitness* de cada par é realmente selecionado. Sendo assim, dependendo do cenário e da área de estudo, um tipo de seleção pode ser adequado ou não. Por isso, neste trabalho foi adotado o método de seleção por torneio para escolher os pais que realizarão o cruzamento e a mutação.

#### 5.2.4 CRUZAMENTO (*CROSSOVER*)

No G-Net é identificado um ponto de cruzamento entre os indivíduos e inverte-se os genes dos pais à partir deste ponto de cruzamento. Na solução do G-Net existem alguns problemas que diferem de cruzamentos comuns:

- O primeiro, consiste no tamanho variável das rotas. Neste caso não tem como saber quantos genes tem um indivíduo porque podem haver rotas com vários nós intermediários ou nenhum nó intermediário quando a origem e o destino são vizinhos diretos. Desta forma, as rotas só são enviadas para o AG, se houver pelo menos duas rotas com no mínimo um nó intermediário entre a origem e o destino. Por isso, nem sempre o algoritmo genético é utilizado, mas quando é necessário ele atua em busca do melhor *Fitness*.
- Outro problema é que os nós de origem e destino nunca podem mudar suas posições. Sendo assim, caso uma rota tenha apenas um nó intermediário, somente este nó é convertido em gene e assim haverá um cromossomo com apenas um gene, pois a origem e o destino não podem ter suas posições alteradas. Entretanto, também podem haver cromossomos com vários genes dependendo da quantidade de nós e do ambiente configurado.
- Devido ao tamanho diferente das rotas, o ponto de cruzamento deve ser comum para as duas rotas. Neste contexto, deve-se identificar o menor indivíduo (com menos genes) e o cruzamento é feito em um ponto aleatório que varia do primeiro gene até o tamanho do menor indivíduo. Por exemplo, se o menor indivíduo possui três genes o ponto de cruzamento varia de um até três como pode ser observado na Figura 7 (cada gene corresponde a 3 algarismos).

De acordo com a Figura 7, pode-se observar que, se o cruzamento fosse escolhido através do maior cromossomo, caso fosse escolhido o ponto de corte à partir do quarto gene, haveria uma falha no cruzamento devido a incompatibilidade de tamanho.



**Figura 7: Exemplos de cruzamento**

**Fonte: Autoria própria.**

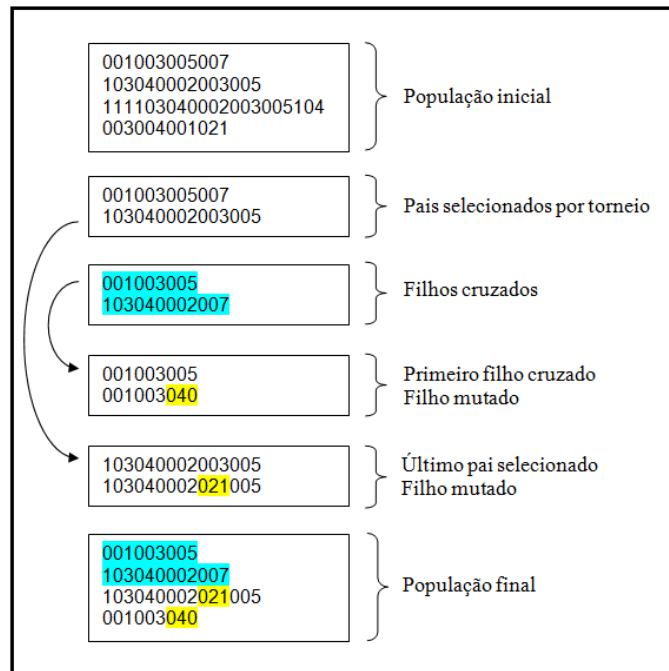
### 5.2.5 MUTAÇÃO

No G-Net é realizada a mutação em alguns pais e filhos. Para isso, são identificados todos os genes da população que podem ser usados na mutação. Em seguida, é selecionado o último pai de cada par e o primeiro filho de cada cruzamento para realizar a mutação. À partir deste momento é escolhido um gene aleatório destes indivíduos e este gene é substituído por outro gene da população. Assim, a população final é gerada conforme observado na Figura 8.

### 5.2.6 REPARAÇÃO DE ROTAS

Durante o cruzamento e mutação podem surgir genes iguais em um indivíduo gerado. Por isso, tanto no cruzamento quanto na mutação é realizada uma verificação para confirmar que não existem genes iguais no mesmo indivíduo. Isto é necessário porque se houver gene repetido, significa que há nó repetido na rota e por isso haveria *loop* na solução encontrada. Por isso, caso haja genes repetidos, somente o primeiro gene é mantido no cromossomo gerado.



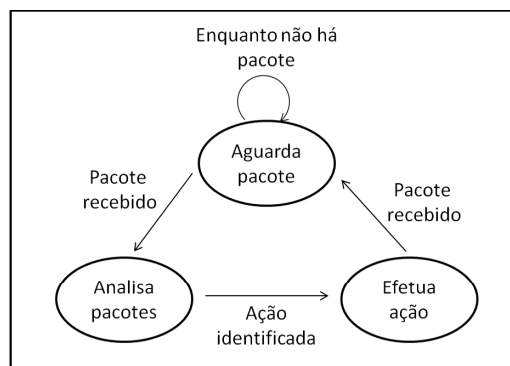


**Figura 8: Exemplo de mutação e geração da população final**

**Fonte: Autoria própria.**

### 5.3 FUNCIONAMENTO DO PROTOCOLO G-NET

Em um nível elevado de abstração o funcionamento do protocolo G-Net pode ser descrito pela máquina de estados da Figura 9. O G-Net aguarda por pacotes e, no seu recebimento, efetua ações específicas de acordo com o tipo de pacote. Então, retorna ao estado de espera.



**Figura 9: Máquina de estados de funcionamento do G-Net**

**Fonte: Autoria própria.**

As ações específicas da máquina de estado são detalhadas no diagrama da Figura 10, o qual apresenta o diagrama completo do protocolo G-Net e permite um melhor entendimento de todo processo de funcionamento do mesmo.

De acordo com a Figura 10, o nó configurado com o G-Net fica aguardando um pacote para tomar uma decisão. Quando um pacote é detectado, primeiramente o G-Net verifica se foi originado no próprio nó. Caso positivo, o G-Net verifica se existe rota para o destino que este pacote deseja atingir. Caso não haja rota para o destino, o G-Net inicia o processo de descoberta de rota. No processo de descoberta de rota, o G-Net retira o pacote da fila de transmissão, descobre rotas com o mesmo processo do DSR, atualiza rotas pai e repõe o pacote na fila de transmissão. Caso existam rotas na tabela de roteamento (“*route cache*”) ele escolhe a melhor rota, envia o pacote para o próximo salto (“*next hop*”) e incrementa o  $Nt(dest)$ .

O G-Net usa os mesmos processos de descoberta de rota do DSR. Porém, o G-Net possui um monitor de pacotes  $Nt(dest)$  que identifica a quantidade de pacotes transmitidos. Quando este monitor percebe que o nó de origem enviou um número pré-determinado  $Np$  de pacotes de dados para o destino escolhido, ele ativa um processo de manutenção e otimização de rotas.

No funcionamento do G-Net, caso o pacote recebido não tenha sido originado no próprio nó (Figura 10), o G-Net verifica se o nó é o destino do pacote ou se é um nó intermediário. Assim, se for um nó intermediário, ele apenas envia o pacote para o próximo salto da rota para atingir o destino. Porém, se o nó é o destino do pacote, o G-Net verifica qual é o tipo do pacote para tomar sua decisão. O G-Net pode identificar três tipos de pacotes e toma suas decisões conforme a seguir:

1. Pacote de dados. Se o nó de destino recebe um pacote de dados, ele apenas extrai a mensagem recebida.
2. Pacote *request*. Caso o destino receba um pacote *request*, ele identifica a origem, inverte a rota e envia um pacote *response* de volta para a origem.
3. Pacote *response*. Quando o nó de destino recebe um pacote *response*, na verdade, significa que este nó de destino atual é o nó de origem inicial que enviou um *request* para manutenção de rotas. Neste caso, quando um nó configurado com o G-Net recebe um pacote *response*, ele atualiza o “*route cache*”, calcula latência e atualiza o “*Fitness*”. Se for rota filho, substitui a pior rota pai. Se não for encontrada uma rota filho, aplica-se o algoritmo genético (cruzamento e mutação) e envia *request* para cada rota filho.

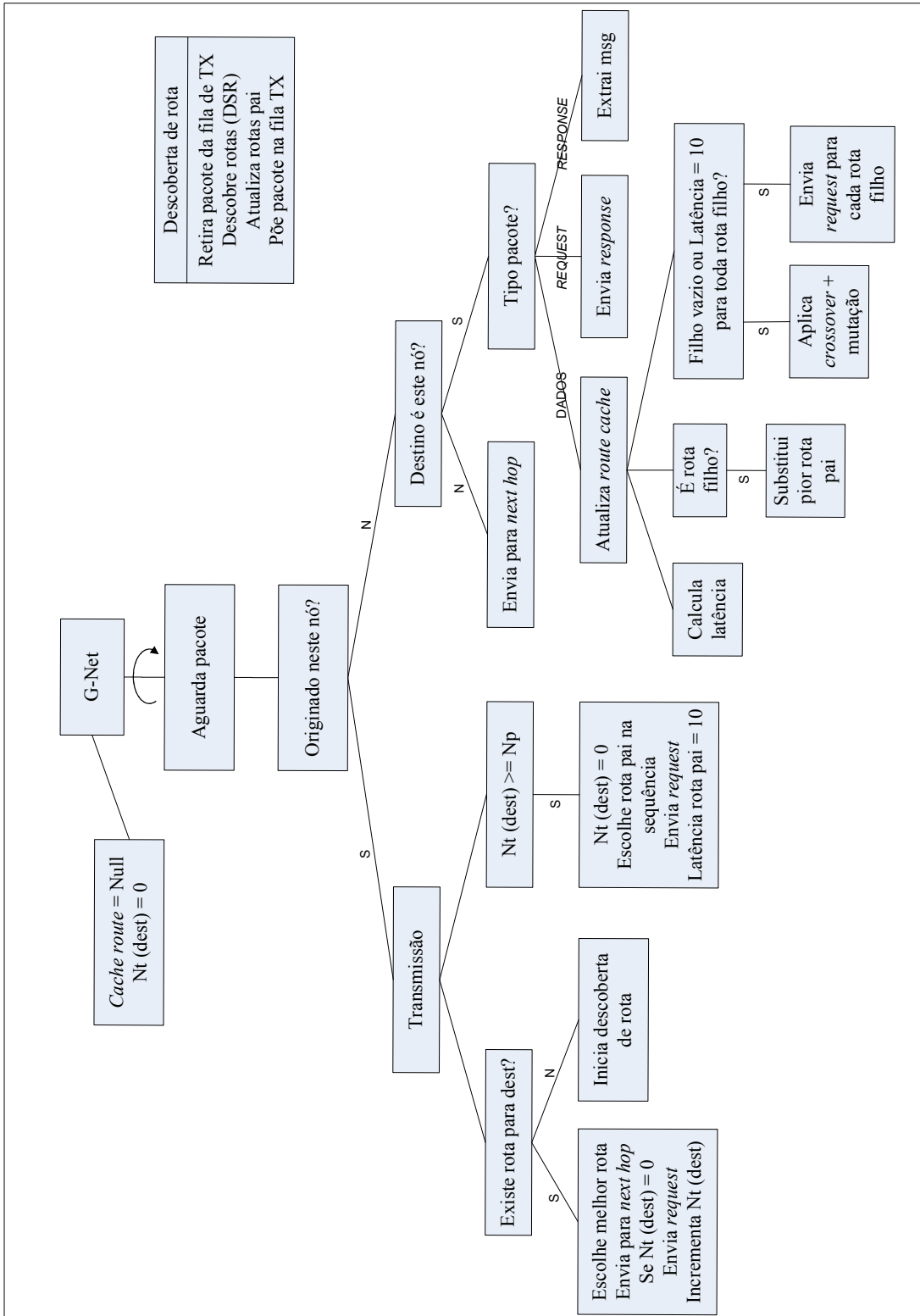


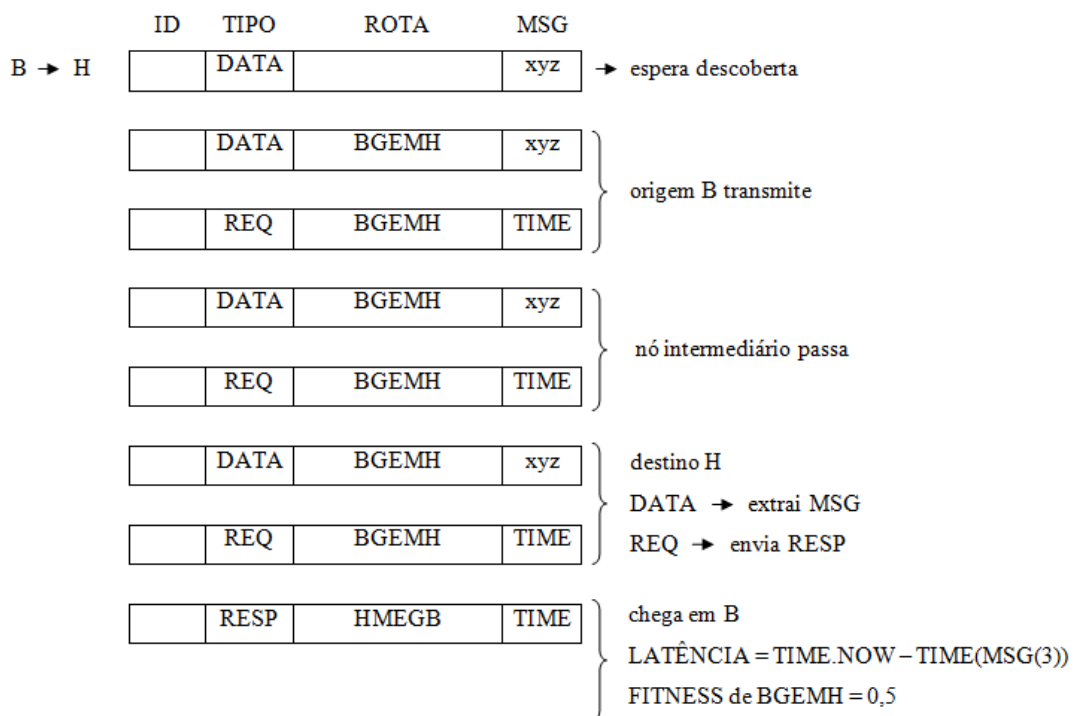
Figura 10: Diagrama de funcionamento do G-Net

Fonte: Autoria própria.

Através da Figura 10 é possível perceber que todo o processo de manutenção do G-Net se diferencia completamente do DSR, tanto na parte de ativação periódica de descoberta de rotas, quanto na parte de otimização com o algoritmo genético.

A Figura 11 apresenta um exemplo de transmissão de pacotes G-Net de um nó de origem B para um nó de destino H, onde G, E e M são possíveis nós intermediários. Verifica-se que o pacote possui os seguintes campos:

- ID - O campo ID é usado para identificar única e exclusivamente cada pacote.
- TIPO - Campo que identifica se o pacote é de dados, *request* ou *response*.
- ROTA - O campo ROTA é preenchido com a rota descoberta pelo G-Net
- MSG - Este campo é preenchido com os dados do pacote de dados ou com o tempo dos pacotes de controle.

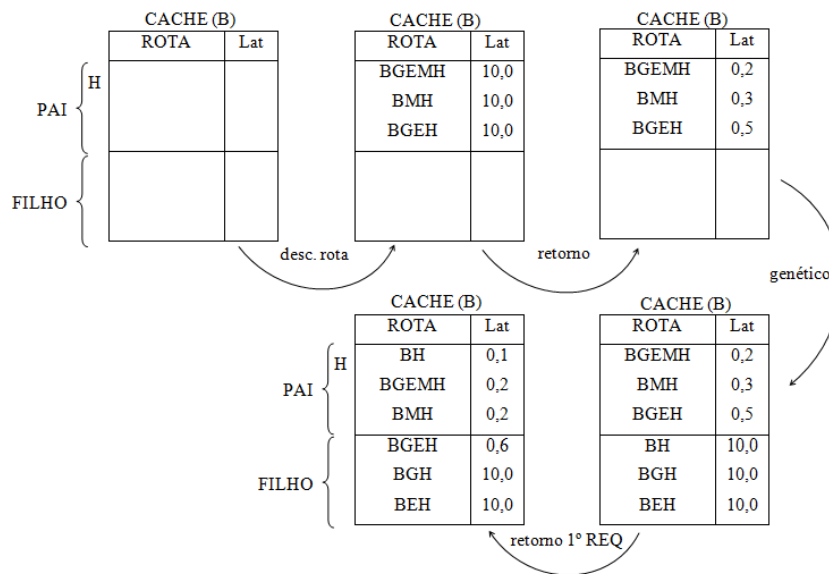


**Figura 11: Exemplo de funcionamento do G-Net - pacotes**

**Fonte: Autoria própria.**

Conforme observado na Figura 11, o pacote de dados preenche o campo MSG com dados e aguarda a descoberta de uma rota de B para H. Se uma rota é descoberta, o campo ROTA é preenchido e o pacote é transmitido. Para calcular a latência da rota utilizada, um pacote *request* com REQ no campo TIPO e com o tempo exato de transmissão é enviado. É possível verificar que os nós intermediários repassam os pacotes até alcançarem o destino. Quando os pacotes chegam no destino H, este extrai os dados e, caso seja um pacote *request*, ele envia um pacote *response* (preenche o campo TIPO com RESP) para B. Quando o pacote RESP chega em B ele calcula a latência subtraindo o tempo atual de recebimento pelo tempo de transmissão inicial, extraído do campo MSG.

A evolução temporal do “*route cache*” do G-Net para o nó B, considerando o exemplo presente, pode ser observada na Figura 12.



**Figura 12: Exemplo de funcionamento do G-Net - *cache* de rotas**

**Fonte: Autoria própria.**

A Figura 12 mostra um exemplo do *cache* de rotas do nó B para o destino H. Observa-se que quando o nó B deseja enviar dados para H e não possui rota para este destino, B descobre as rotas para H com o mesmo mecanismo de descoberta do DSR. As rotas descobertas são incluídas no *cache* de rotas com latências altas (latência=10) enquanto são calculadas suas verdadeiras latências. Em seguida, essas rotas descobertas com suas respectivas latências são utilizadas pelo AG como população inicial e conseqüentemente possíveis pais para a geração das rotas filhos. Depois de gerar as rotas filhos, estas são cadastradas no *cache* de rotas com

latências altas enquanto são verificadas se estas rotas são existentes. Para verificar se as rotas filhas são rotas existentes, são realizados os cálculos de suas latências. Após o cálculo das latências, verificam-se duas situações:

- Caso seja possível calcular a latência de uma rota filho, esta rota é confirmada no *cache* de rotas com sua respectiva latência. Além disso, se sua latência for melhor que as latências das rotas pai, esta rota substitui a melhor rota pai, sendo a nova rota escolhida para enviar os dados, como acontece com a rota “BH” da Figura 12.
- Caso não seja possível calcular a latência de uma rota filho, esta rota é considerada inexistente ou inalcançável. Neste caso, ela permanece no *cache* de rotas até ser descartada pelo tempo de expiração de rotas. Na Figura 12, pode-se visualizar que não foi possível calcular as latências das rotas “BGH” e “BEH” e por isso permanecem no “*route cache*” até serem descartadas.

Desta forma, é possível perceber que as rotas filhas com boas latências são aproveitadas ao passo que as rotas inalcançáveis são simplesmente descartadas. Observa-se também que o G-Net funciona ora como reativo e ora como proativo, pois atualiza seu “*route cache*” periodicamente a cada  $Np$  pacotes transmitidos. Com isso, podem ser verificados dois ganhos significativos:

- a) Primeiramente, o simples fato de atualizar as rotas periodicamente, já caracteriza uma vantagem em relação ao DSR, pois quanto mais novas são as rotas num ambiente de alta mobilidade como é o caso das VANETs, mais confiáveis elas são. Pois conforme o tempo passa, os veículos se deslocam e aumenta o risco de ocorrerem problemas de nós e/ou link disjuntos.
- b) Outra vantagem do G-Net é que ele utiliza o algoritmo genético para otimizar as rotas. Sendo assim, o G-Net realiza o cruzamento e mutação destas rotas permitindo que novas rotas sejam geradas, com possíveis melhores latências. Diferente do DSR que não utiliza alguns nós intermediários em mais de uma rota, devido a proteção contra *loop* que ele possui (vide item 5.1).

## 6 IMPLEMENTAÇÃO E RESULTADOS DE SIMULAÇÃO

### 6.1 IMPLEMENTAÇÃO DO G-NET NO NS-3

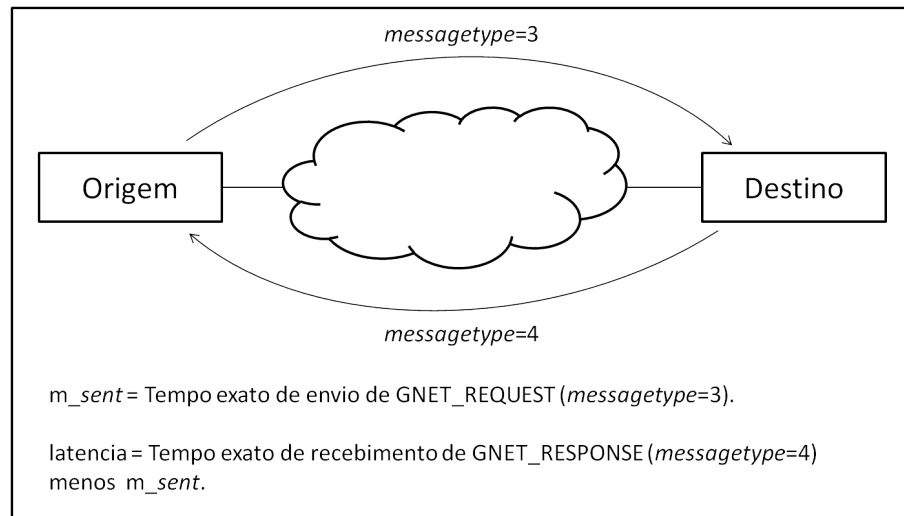
Para analisar o comportamento do G-Net, o protocolo foi implementado no ns3 versão 3.17. Para a implementação do G-Net no ns-3 foi necessário criar uma estrutura chamada *route cache* que consiste num mapa de vetor de rotas do G-Net, que armazena as rotas com suas respectivas latências, bem como o contador de pacotes de dados enviados ao destino,  $Nt(dest)$ .

Para o cálculo de latência das rotas foram criados dois novos tipos de pacote de controle chamados de GNET\_REQUEST e GNET\_RESPONSE. Assim como o DSR, o G-Net possui dois tipos de mensagem nativos, o CONTROL\_PACKET com código 1, e o DATA\_PACKET com código 2. Porém, no G-Net foram criados dois novos tipos de mensagem: o GNET\_REQUEST e GNET\_RESPONSE, que foram identificados com os códigos 3 e 4, respectivamente.

Desta forma, quando se deseja calcular a latência da rota, uma mensagem de código “3” é enviada para o destino desta rota. Quando o nó de destino recebe o pacote de dados com o código “3”, captura-se o endereço IP de origem e usa este endereço como destino para enviar um pacote GNET\_RESPONSE com o código “4” de volta para o nó de origem. Uma variável *m\_sent* armazena o tempo exato que o GNET\_REQUEST é enviado, outra variável “*latência*” armazena o momento que o GNET\_RESPONSE é recebido pelo nó de origem. A diferença entre os instantes de tempo permite determinar a latência da rota (Figura 13).

Após armazenar as rotas e latências atualizadas, estas entradas de rotas do “*route cache*” do G-Net são convertidas para o formato de cromossomos. Os cromossomos são usados como população do algoritmo genético (AG) e as latências são utilizadas para o cálculo do *Fitness*. As rotas geradas pelo AG são reconvertidas para o formato de “IP\_VECTOR” (vetor de rota), e avaliadas através do envio de GNET\_REQUEST e GNET\_RESPONSE.

Para converter as rotas em cromossomos, cada nó da rede é considerado um gene e o conjunto de nós de uma rota completa (vetor de rota) representa o cromossomo. Para isso, foi



**Figura 13: Cálculo da latência**

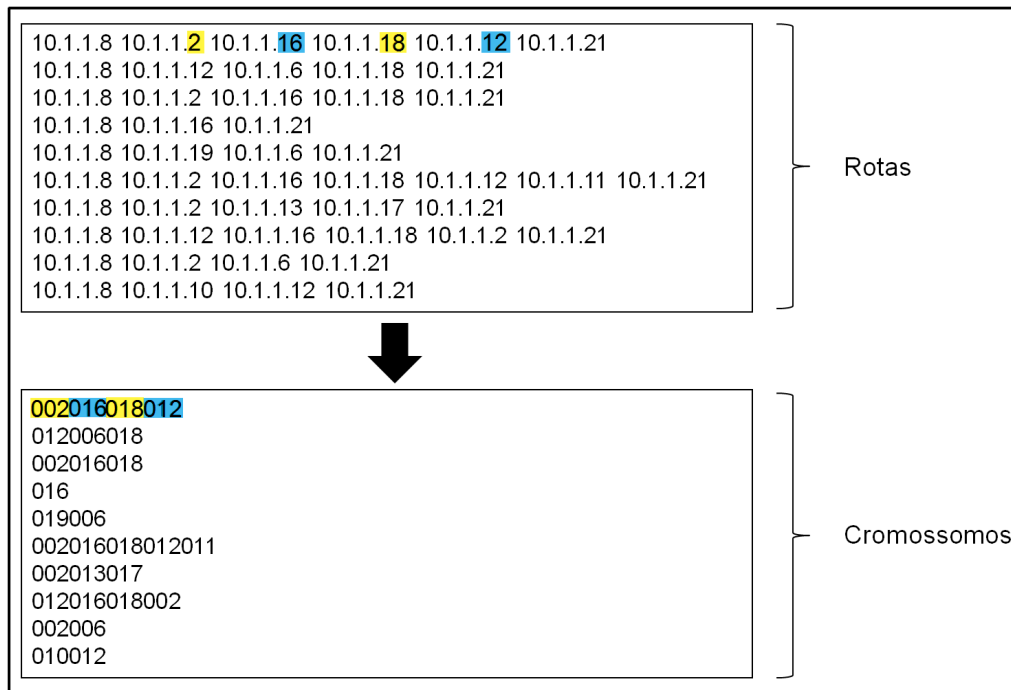
**Fonte: Autoria própria.**

criado um mapa de rotas do G-Net, para cada destino. Neste sentido, como cada rota contém um vetor de endereços IP da origem até o destino, os IPs precisam ser convertidos em genes. Um problema encontrado foi definir como os IPs seriam representados. Foi definido então, que o último campo do endereço IP representa um gene, porém como as unidades tem uma casa decimal, as dezenas duas e as centenas possuem três casas decimais, houve a necessidade de padronizar todos os genes com três casas decimais para poder aceitar até cem ou um número superior de veículos. A Figura 14 permite visualizar como foi realizada a conversão de rotas em cromossomos e como os nós representam os genes.

Desta forma, a cada três caracteres temos um gene e a *string* inteira representa o cromossomo. Através da Figura 14 pode-se observar que os nós de origem e destino foram retirados para aplicar o algoritmo genético. Isto deve ser feito porque a origem e o destino nunca devem mudar suas posições. Por isso o cromossomo é representado apenas pelos nós intermediários da rota completa.

Outra dificuldade encontrada nesta proposta é o fato das rotas (cromossomos) terem tamanhos diferentes. Assim, podem haver rotas em que o nó de origem e o nó de destino são vizinhos e por isso não há nó intermediário (comunicação direta). Neste caso, estas rotas possuem apenas dois nós que são a própria origem e o destino. Por isso, estas rotas não entram no algoritmo genético, mas suas latências são calculadas e, tanto a rota quanto sua latência são armazenadas no mapa de rotas do G-Net. Isto é necessário porque, neste caso, muito provavelmente estas rotas serão melhores do que as rotas geradas pelo AG, pois dificilmente uma rota com mais nós terá uma latência menor. Sendo assim, houve a necessidade





**Figura 14: Exemplo de conversão de rotas em cromossomos**

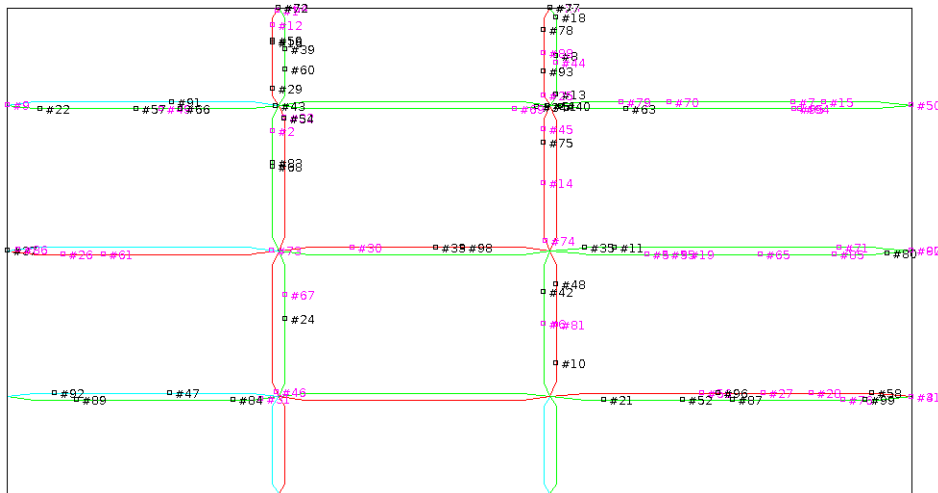
**Fonte: Autoria própria.**

de configurar o G-Net de tal forma que, as rotas só são enviadas para o AG quando existem pelo menos duas rotas com no mínimo um nó intermediário. Mesmo assim, as rotas do AG são sempre consideradas, pois como nas VANETs os nós se deslocam muito rapidamente, é de extrema importância manter rotas de proteção para o caso de falhas.

## 6.2 TRACES DE MOBILIDADE

Conforme já informado no item 4.4, neste trabalho optou-se pelo VanetMobiSim com IDM\_IM e o IDM.LC. A Figura 15 mostra um exemplo de simulação com 100 veículos, onde é possível perceber um acúmulo de automóveis nas vias com sinais fechados. Também foi adotado o modelo de ultrapassagens (MOBIL) (KESTING et al., 2007) que interage com IDM\_IM para gerenciar mudanças de faixa, acelerações e frenagens do veículo. Através da Figura 15, verifica-se que as vias vermelhas representam semáforos com o sinal fechado, ocasionando o acúmulo de veículos, enquanto as vias verdes representam sinal aberto. Além disso, foram configuradas vias de mão dupla com duas faixas em cada sentido, permitindo uma maior fluidez do trânsito.

Durante a simulação, o VanetMobiSim envia os dados de movimento para um arquivo de saída. Este arquivo contém a identificação de cada veículo (ID do nó), a posição inicial,



**Figura 15: Imagem de movimento de 100 veículos simulados no VanetMobisim.**

**Fonte: Autoria própria.**

o tempo decorrido de toda a simulação, as posições dos veículos durante seus deslocamentos, bem como a velocidade dos mesmos para cada alteração de posição no decorrer do tempo. Este arquivo de saída é conhecido como *trace* de mobilidade (Figura 16) do VanetMobiSim e estas informações são essenciais para importação pelo ns-3.

```

$node (46) set X_ 1300.000001
$node (46) set Y_ 1500.000001
$node (46) set Z_ 0.0
$node (47) set X_ 600.000001
$node (47) set Y_ 1.0E-6
$node (47) set Z_ 0.0
$node (48) set X_ 600.000001
$node (48) set Y_ 1.0E-6
$node (48) set Z_ 0.0
$node (49) set X_ 1.0E-6
$node (49) set Y_ 800.000001
$node (49) set Z_ 0.0
$ns_ at 21.0 "$node (0) setdest 607.000001 132.64861059677025 11.973682"
$ns_ at 21.0 "$node (1) setdest 1467.8300866612624 807.000001 11.979815"
$ns_ at 21.0 "$node (2) setdest 595.7239155474653 1363.3478399782723 13.089251"
$ns_ at 21.0 "$node (3) setdest 1295.7363997781497 1363.4548476695181 13.0690775"
$ns_ at 21.0 "$node (6) setdest 593.000001 1418.3753880222719 8.927308"
$ns_ at 21.0 "$node (7) setdest 132.11731784952974 793.000001 11.9669075"
$ns_ at 21.0 "$node (8) setdest 593.000001 1455.1891181616736 6.1176486"
$ns_ at 21.0 "$node (9) setdest 1293.000001 1418.3834403552971 8.922651"
$ns_ at 21.0 "$node (12) setdest 73.1187472597154 793.000001 8.598771"
$ns_ at 21.0 "$node (13) setdest 30.000001 793.000001 4.8590765"
$ns_ at 21.0 "$node (14) setdest 1526.8812624865602 807.000001 8.598561"
$ns_ at 21.0 "$node (15) setdest 1570.000001 807.000001 4.859092"
$ns_ at 21.0 "$node (16) setdest 1293.000001 1455.1904281046516 6.117511"
$ns_ at 21.0 "$node (17) setdest 1584.4706001987345 803.6235278536741 1.7413143"
$ns_ at 21.0 "$node (19) setdest 593.000001 1479.3923646205728 3.204681"
$ns_ at 21.0 "$node (20) setdest 15.529354766915661 796.3764851210291 1.7413013"
$ns_ at 21.0 "$node (21) setdest 607.000001 73.15755843905902 8.567746"
$ns_ at 21.0 "$node (22) setdest 593.000001 1490.000001 0.30598915"
$ns_ at 21.0 "$node (36) setdest 607.000001 34.77166955102405 5.421609"
$ns_ at 21.0 "$node (38) setdest 604.2278397237117 15.099425013195088 1.6597519"
$ns_ at 21.0 "$node (46) setdest 1293.000001 1479.3934948067106 3.2044713"

```

**Figura 16: Exemplo de “trace de mobilidade” do VanetMobiSim mostrando dados sobre o movimento de veículos.**

**Fonte: Autoria própria.**

Toda a estrutura de comunicação nos veículos, de tal forma que pudessem se comunicar através de uma rede sem fio, foi estabelecida utilizando a versão 3.17 do simulador de rede ns-3 de modo a simular os nós *ad hoc* de acordo com o número de veículos configurados no VanetMobiSim. Como o VanetMobiSim gera traces compatíveis com o ns-2, foi utilizado o assistente 'Ns2MobilityHelper' para importação da mobilidade dos veículos no ns-3.

### 6.3 CONFIGURAÇÃO DE CENÁRIOS E PROTOCOLOS

Na literatura, são encontrados cenários de simulação VANET com áreas que variam de 200m x 200m (SPAHO et al., 2013) até alguns quilômetros (KORKMAZ et al., 2004; PENG et al., 2011). Também são encontrados ambientes com tempo da simulação a partir de 30 segundos (FONSECA et al., 2012), *range* de comunicação da antena que iniciam com 100m (LABIOD et al., 2010) e experimentos a partir de 30 veículos (SPAHO et al., 2012) para um ambiente tipicamente urbano.

Nesta dissertação os protocolos foram analisados em dois tipos de cenário. Um de 600m x 600m de área, alcance de transmissão de 100 metros e com 20, 30, 40 e 50 veículos (Tabela 3).

**Tabela 3: Resumo dos parâmetros utilizados nas simulações do cenário 1**

Parâmetros das simulações - Cenário 1	
Área da simulação	600m x 600m
Alcance de comunicação	100m
Modelo de propagação	Nakagami (m = 1)
Simulador de mobilidade	VanetMobiSim
Aplicação	CBR
Transporte	UDP
MAC e PHY	802.11p
Tamanho do pacote	512 bytes
Taxa de transmissão	4 pacotes/segundo
Fila da interface	20 pacotes
Tempo da simulação	100 segundos
Número de veículos	20, 30, 40 e 50
Velocidade máxima dos veículos	60 km/h
Protocolos de roteamento	AODV, DSR e G-Net
Intervalo de confiança	95%

**Fonte: Autoria própria**

O outro ambiente possui uma área de 1km x 1km, alcance de transmissão de 250 metros, com 25, 50, 75 e 100 veículos, conforme pode ser observado na Tabela 4.

Os dois cenários tem o intuito de verificar se o comportamento dos protocolos apresentam uma divergência considerável em diferentes situações.

**Tabela 4: Resumo dos parâmetros utilizados nas simulações do cenário 2**

Parâmetros das simulações - Cenário 2	
Área da simulação	1000m x 1000m
Alcance de comunicação	250m
Modelo de propagação	Nakagami (m = 1)
Simulador de mobilidade	VanetMobiSim
Aplicação	CBR
Transporte	UDP
MAC e PHY	802.11p
Tamanho do pacote	512 bytes
Taxa de transmissão	4 pacotes/segundo
Fila da interface	20 pacotes
Tempo da simulação	100 segundos
Número de veículos	25, 50, 75 e 100
Velocidade máxima dos veículos	60 km/h
Protocolos de roteamento	AODV, DSR e G-Net
Intervalo de confiança	95%

**Fonte: Autoria própria**

Após definir os cenários, foi configurada toda a pilha de protocolos desde a camada física e enlace, com o protocolo OFDM e o protocolo IEEE 802.11p. Na camada de rede foi configurado o protocolo IP com os respectivos endereços IPs de cada nó e os protocolos de roteamento AODV, DSR e G-Net. No final do modelo OSI foi utilizado o protocolo UDP para transmissão de dados da camada de aplicação.

#### 6.4 PROTOCOLOS DE ROTEAMENTO ANALISADOS

Para analisar o desempenho do protocolo proposto (G-Net), é importante comparar seu comportamento em relação a outros protocolos já conhecidos. Neste trabalho foi realizada uma comparação com o AODV, por ser um protocolo *ad hoc* bastante utilizado em estudos de ambientes móveis e o DSR, pelo fato de ter servido como base para a criação do G-Net.

##### 6.4.1 AODV

Protocolo de roteamento MANET largamente utilizado na comunidade científica, tanto para comparação com outros protocolos MANET como VANET (BISOYI; SAHU, 2010; PAUL et al., 2011b; PENG et al., 2011). Por ser utilizado em vários cenários, o AODV permite que seja comparado com outros trabalhos, de tal forma que outros protocolos possam ser avaliados a partir de seu comportamento de acordo com o ambiente estudado.

O AODV é uma melhoria do DSDV com o esquema sob demanda (reativo). Ele minimiza a sobrecarga da rede porque envia pacote de *broadcast* para criação de rota apenas

quando é necessário enviar dados. Quando o nó de origem quer enviar dados para o nó de destino, ele primeiro inicia o processo de descoberta de rota. Neste processo, o nó de origem transmite o pacote *Route Request* (RREQ) para os seus vizinhos que encaminham para os próximos vizinhos até que RREQ alcance o destino ou o nó que conhece o caminho para o destino. Quando os nós intermediários recebem RREQ, eles registram em suas tabelas o endereço dos vizinhos, estabelecendo assim um caminho inverso. Quando o nó que conhece o caminho para o destino ou o próprio nó de destino recebe RREQ, ele envia de volta o pacote *Route Reply* (RREP) para o nó de origem. Este pacote RREP é transmitido pelo caminho inverso. Quando o nó de origem recebe o pacote RREP, ele aprende o caminho para o nó de destino e armazena as informações deste caminho descoberto em sua tabela de roteamento (SPAHO et al., 2013).

Quanto ao tipo de mensagem de controle, o AODV trabalha com pacotes de requisição de rota (RREQs), pacotes de respostas de rota (RREPs) e pacotes de erros de rota (RERRs). Ele permite que os nós móveis obtenham rotas rapidamente para novos destinos, e não necessita que os nós mantenham rotas para destinos que não estão em comunicação ativa. No AODV, os nós móveis respondem a quebra de *links* e mudanças na topologia da rede em tempo hábil. Quando ocorre quebra de *link*, o protocolo faz com que o conjunto de nós afetados sejam notificados para que tenham capacidade de invalidar as rotas que estão com *links* quebrados (PERKINS et al., 2003). Caso exista alguma falha no *link*, o AODV envia relatório de erro e outro RREQ é realizado (PAUL et al., 2011b).

Percebe-se que, por ser reativo, o AODV é um protocolo que reduz a sobrecarga de roteamento porque só envia pacotes de broadcast para descoberta de rotas quando um nó deseja enviar dados. Desta forma, mesmo que haja atualização e otimização de rotas constantes com o protocolo proposto (G-Net), é possível verificar se o mesmo é capaz de manter uma sobrecarga de roteamento aceitável em relação ao AODV e ao DSR.

#### 6.4.2 DSR

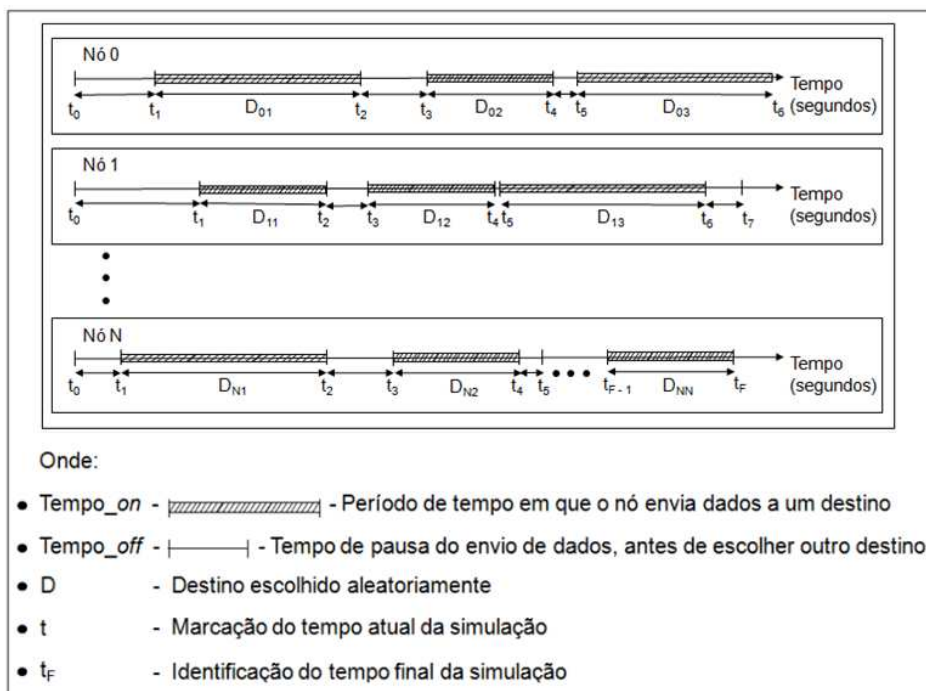
Protocolo de roteamento MANET reativo utilizado como base para a criação do G-Net. O DSR tem a característica de ser “*source routing*” e costuma apresentar bons resultados com relação a sobrecarga de roteamento (RANI et al., 2011; CHOWDHURY et al., 2011).

O protocolo DSR é composto por dois mecanismos principais que trabalham em conjunto para a descoberta e manutenção de rotas na rede ad hoc (JOHNSON et al., 2007). Quando um nó precisa enviar dados e não sabe como chegar em um destino, ele utiliza o mecanismo de descoberta de rotas. De acordo com Singh et al. (2011), o mecanismo de

manutenção de rotas do DSR é capaz de detectar quando uma rota pára de responder ao longo do tempo. Quando o mecanismo de manutenção de rotas indica que uma rota está quebrada, o nó pode tentar usar qualquer outra rota conhecida de sua tabela de rotas, ou pode reativar o mecanismo de descoberta para encontrar uma nova rota.

## 6.5 TRÁFEGO DE DADOS

Duas estratégias de geração de tráfego foram consideradas, ambas utilizando fontes *On/Off* com taxa de bits constante (CBR) e protocolo UDP. Na primeira estratégia cada nó envia dados a um destino aleatório por um tempo aleatório entre 40 e 50 segundos. Depois, o nó fica sem enviar dados por um período aleatório entre 0 e 5 segundos, conforme pode ser observado na Figura 17. Após o período de pausa, o nó escolhe outro destino aleatoriamente e repete o processo até o final da simulação.



**Figura 17: Representação gráfica do tráfego de dados da simulação**

**Fonte: Autoria própria.**

Desta forma, procura-se simular de forma mais fiel a transmissão de dados entre veículos. Pois, em um ambiente real, um veículo nem sempre envia dados para um único destino no decorrer do tempo. Além disso, obstáculos e problemas técnicos também podem

proporcionar pequenas pausas durante a transmissão de dados.

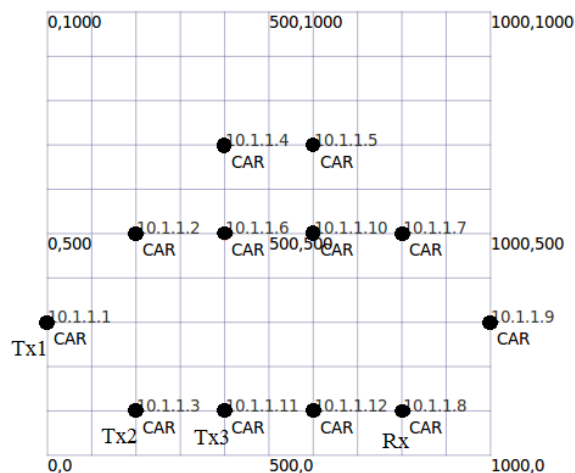
Na segunda estratégia cada nó escolhe um destino aleatório e transmite de forma ininterrupta durante todo o tempo de simulação.

## 6.6 ANÁLISE DE CAMINHOS PERCORRIDOS

Com o intuito de validar a solução proposta quanto a escolha de rotas em função do tráfego de dados e latência, foi criado um cenário com 12 nós para visualizar melhor o comportamento do G-Net.

### 6.6.1 ANÁLISE DE CAMINHOS PERCORRIDOS DE ACORDO COM A LATÊNCIA

Neste cenário foram criados três nós de origem e um único nó de destino (Figura 18). Assim, o nó 1 (10.1.1.1) envia dados para o nó 8 (10.1.1.8) desde o início até o fim da simulação. No instante 30 segundos, o nó 3 (10.1.1.3) começa a transmitir dados para o nó 8 e, o nó 11 (10.1.1.11) transmite dados ao mesmo destino à partir de 60 segundos até o final da simulação.



**Figura 18: Cenário para análise de caminhos percorridos de acordo com a latência das rotas**

**Fonte: Autoria própria.**

Com três nós transmissores e apenas um nó de destino, este cenário permite gerar sobrecarga de transmissão de pacotes nos enlaces próximos ao destino. A sobrecarga de pacotes num *link* acarreta no aumento de latência e espera-se que o protocolo seja capaz de escolher uma rota mais adequada. O objetivo é analisar o comportamento do G-Net quanto a escolha das rotas de acordo com a latência. Por isso, os nós permaneceram fixos para evitar qualquer interferência dos movimentos em relação a estas latências.

## 6.6.2 RESULTADOS DA ANÁLISE DE CAMINHOS PERCORRIDOS

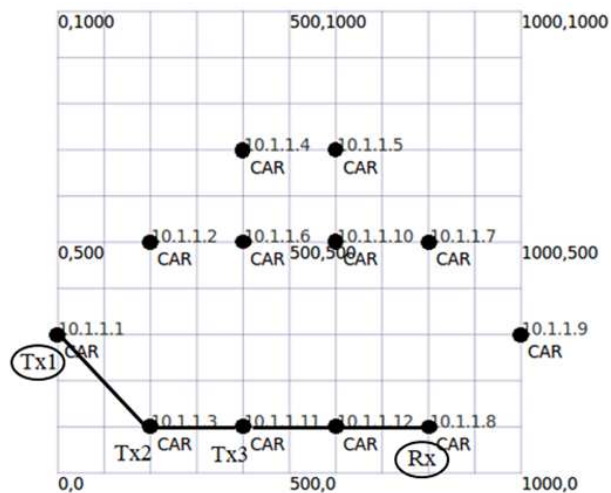
Para a análise dos caminhos percorridos, a cada 20 segundos de simulação são exibidas as rotas e latências utilizadas. Com isso, é possível verificar qual o caminho percorrido e a latência da rota escolhida pelo G-Net durante as alterações de tráfego de dados. A Tabela 5 mostra um exemplo de caminho percorrido pelo G-Net para a entrega de dados do nó 1 para o nó 8, bem como suas respectivas latências.

**Tabela 5: Exemplo do resultado de caminhos percorridos**

Tempo da simulação	Nó de origem	Rota utilizada	Latência (seg.)
20 segundos	Nó 1	1 - 3 - 11 - 12 - 8	0,004564
40 segundos	Nó 1	1 - 3 - 11 - 12 - 8	0,083701
	Nó 3	3 - 11 - 12 - 8	0,036479
60 segundos	Nó 1	1 - 2 - 4 - 10 - 7 - 9 - 8	0,047314
	Nó 3	3 - 11 - 12 - 8	0,015064
80 segundos	Nó 1	1 - 2 - 4 - 10 - 7 - 9 - 8	0,065578
	Nó 3	3 - 11 - 12 - 8	0,016917
	Nó 11	11 - 12 - 8	0,014627

Fonte: Autoria própria

De acordo com a Tabela 5, as Figuras 19, 20, 21 e 22 mostram as rotas com as latências escolhidas nos instantes 20, 40, 60 e 80 segundos, respectivamente. A Figura 19 mostra que o G-Net escolhe o caminho mais curto da origem até o destino no instante 20 segundos. Como não há nenhum outro nó transmitindo neste momento, o protocolo escolhe a rota mais próxima conforme esperado.

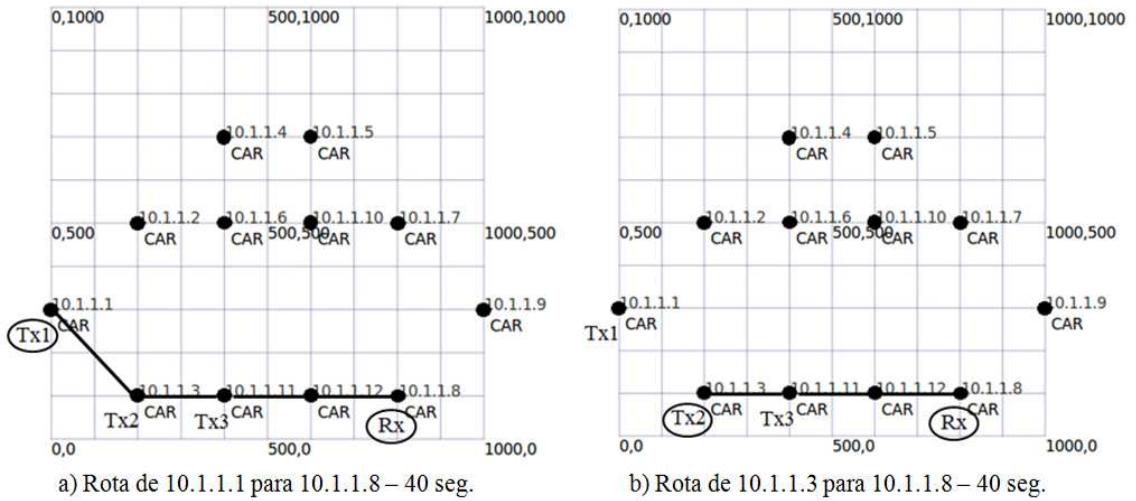


**Figura 19: Exemplo de caminho percorrido no instante 20 segundos**

Fonte: Autoria própria.

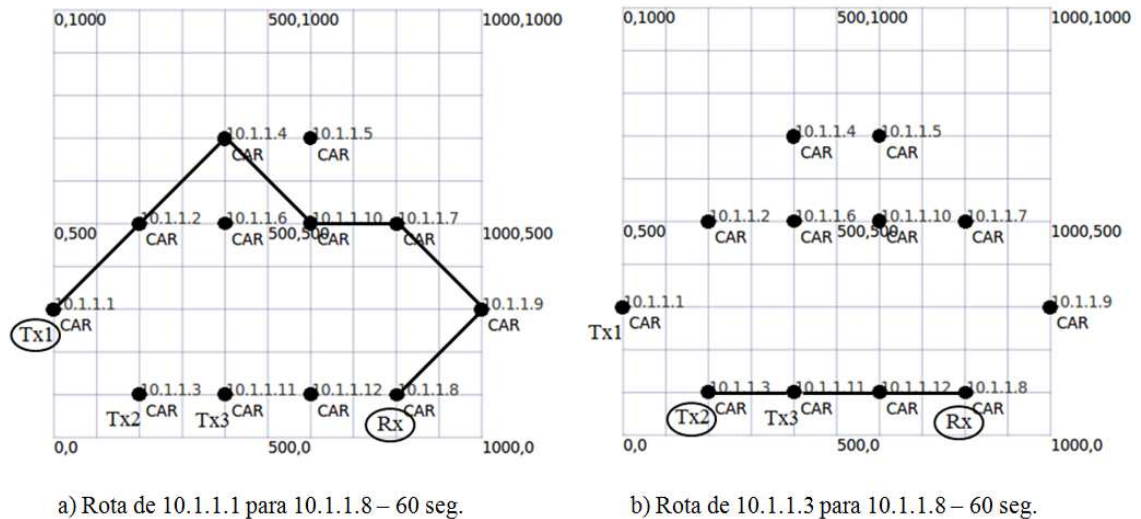


No instante 40 segundos, de acordo com a Figura 20, observa-se que o nó 1 continua usando o mesmo caminho do instante 20 segundos. Entretanto, a Figura 21 demonstra que o G-Net escolheu outra rota após identificar o aumento de latência na rota mais curta.



**Figura 20: Exemplo de caminhos percorridos no instante 40 segundos**

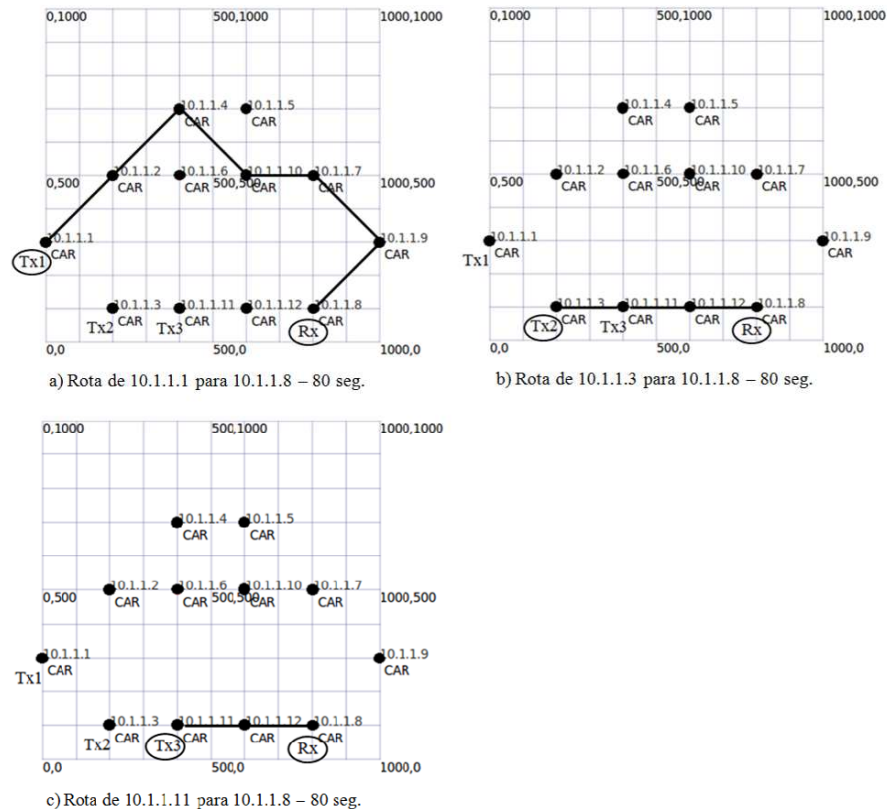
Fonte: Autoria própria.



**Figura 21: Exemplo de caminhos percorridos no instante 60 segundos**

Fonte: Autoria própria.

A Figura 22 mostra que o nó 3 continua usando o caminho mais curto neste exemplo, bem como o nó 11. Além disso, apesar do nó 1 continuar adotando a mesma rota do instante 60 segundos, verifica-se que o G-Net continua escolhendo uma rota alternativa com o aumento de latência do caminho mais curto.



**Figura 22: Exemplo de caminhos percorridos no instante 80 segundos**

**Fonte: Autoria própria.**

Isso confirma que o G-Net foi capaz de escolher os melhores caminhos de acordo com a latência das rotas. Verifica-se portanto, que o cálculo da latência e a escolha do caminho está acontecendo de acordo com esta latência e por isso o G-Net é recomendado em ambientes onde o tráfego de dados é grande e precisam-se buscar caminhos alternativos de acordo com a menor latência.

## 6.7 MÉTRICAS ANALISADAS

Para analisar o comportamento dos protocolos G-Net, DSR e AODV, foram considerados dois indicadores de desempenho: *taxa média de entrega de dados* e *sobrecarga de roteamento* de acordo com a variação do número de veículos no ambiente configurado.

### 6.7.1 TAXA MÉDIA DE ENTREGA DE DADOS

O *Average Delivery Ratio* (ADR) corresponde a taxa média de entrega de pacotes de dados ao longo da simulação. Para calcular o ADR, foi desenvolvido um filtro do arquivo *trace* para identificar a quantidade de pacotes de dados recebidos (*rxPackets*) e perdidos (*lostPackets*) de cada fluxo. Primeiramente foi calculada a taxa de entrega de dados de cada fluxo,  $DR_j$ , conforme observado na Equação 2.

$$DR_j = \frac{rxPackets_j}{(rxPackets_j + lostPackets_j)}, \quad (2)$$

Onde:

$rxPackets_j$  - Quantidade total de pacotes de dados recebidos para o fluxo  $j$ .

$lostPackets_j$  - Quantidade total de pacotes de dados perdidos para o fluxo  $j$ .

O ADR (Equação 3) é calculado como:

$$ADR = \frac{\sum_{j=1}^{TF} DR_j}{TF}, \quad (3)$$

Onde:

$TF$  (Total de Fluxos) - Quantidade total de fluxos  $j$  válidos.

### 6.7.2 SOBRECARGA DE ROTEAMENTO

A sobrecarga de roteamento (RO) é uma métrica utilizada para analisar a quantidade de pacotes de roteamento transmitidos pelo protocolo em relação a quantidade de pacotes de dados recebidos com sucesso no destino. Foi realizado um filtro do arquivo *trace* do ns-3 para extrair o total de pacotes de roteamento transmitidos e a quantidade de pacotes de dados recebidos com sucesso nos nós de destino.

O RO é calculado conforme a Equação 4 a seguir:

$$RO = \frac{\sum txPacketRO}{\sum rxPacketDATA}, \quad (4)$$

Onde:

$\Sigma txPacketRO$  - Total de pacotes de roteamento transmitidos.

$\Sigma rxPacketDATA$  - Total de pacotes de dados recebidos com sucesso.

### 6.7.3 RESULTADOS DAS MÉTRICAS ANALISADAS

Quanto as métricas *taxa média de entrega de dados e sobrecarga de roteamento*, os protocolos analisados foram simulados em duas condições de tráfego para cada cenário, sendo um tráfego ininterrupto e outro com pausa no tráfego durante a simulação. Além do tamanho da área, os cenários criados para análise destas métricas também diferem quanto ao alcance da transmissão, alterando significativamente o comportamento dos protocolos analisados.

O protocolo proposto foi implementado para realizar atualização e otimização de rotas de duas formas diferentes:

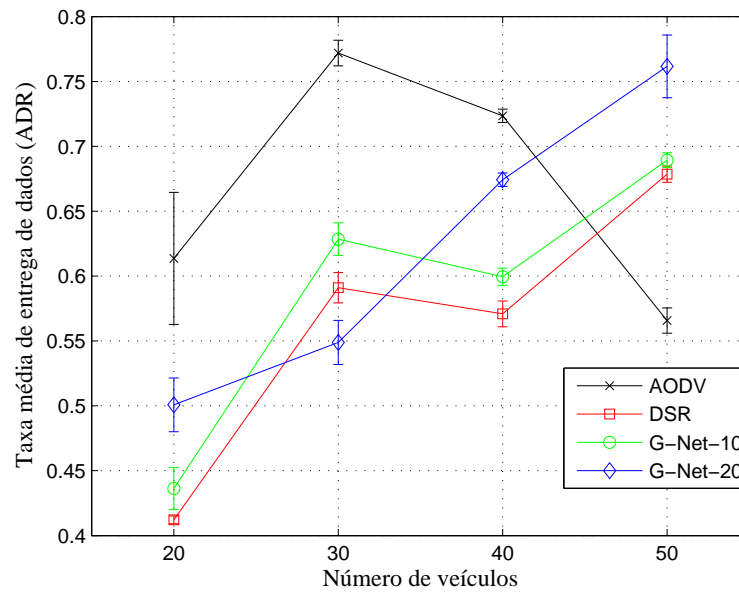
- No primeiro caso, o G-Net considera  $Np = 10$  pacotes transmitidos e nos gráficos dos resultados é chamado de G-Net-10.
- No segundo caso, o G-Net considera  $Np = 20$  pacotes transmitidos. Nos gráficos é chamado de G-Net-20.

#### 6.7.3.1 RESULTADOS COM TRÁFEGO CBR ININTERRUPTO

Neste experimento os resultados foram analisados com um tráfego de dados CBR ininterrupto do início ao fim da simulação. Desta forma, é possível verificar o desempenho contínuo do protocolo proposto.

##### 6.7.3.1.1 RESULTADOS CENÁRIO 1 COM TRÁFEGO CBR ININTERRUPTO

De acordo com os resultados apresentados na Figura 23 é possível observar que o protocolo AODV mantém uma taxa de entrega mais elevada que os outros protocolos com 20, 30 e 40 veículos, porém o G-Net supera o DSR em quase todos os casos, principalmente acima de 30 veículos, comprovando que a solução tem mais eficácia conforme aumenta o número de nós da rede. Além disso, deve-se verificar o porque do comportamento do G-Net-20. Acredita-se que devido a atualizações menos constantes, o G-Net-20 pode continuar usando rotas boas acima de 30 veículos, porém com 30 veículos ele pode ter usado rotas piores enquanto o G-Net-10 substitui por rotas melhores.

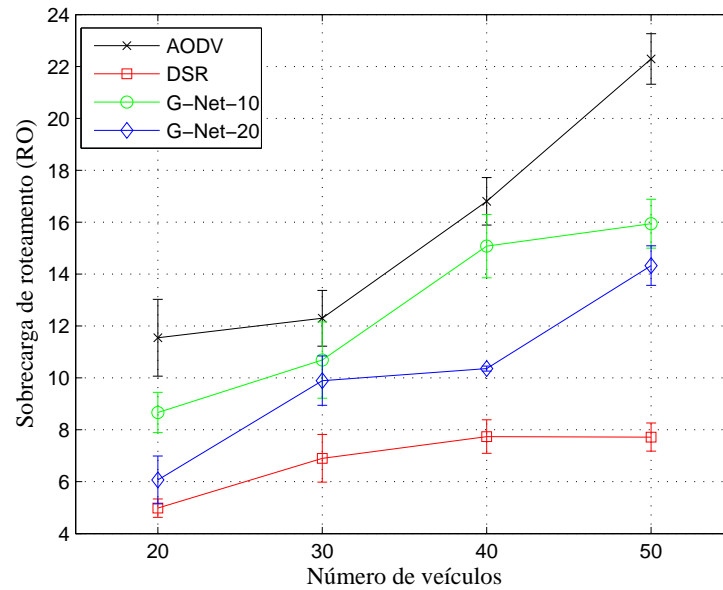


**Figura 23: Gráfico de resultados ADR do cenário 1 com tráfego CBR ininterrupto**

**Fonte: Autoria própria.**

Com 20 veículos o G-Net-20 conseguiu um valor de 50,08% de taxa de entrega, melhor que o G-Net-10 e o DSR. Apesar do G-Net-20 ficar com um ADR de 54,89%, inferior aos outros protocolos com 30 veículos, observa-se que ele melhorou seu desempenho consideravelmente até alcançar uma taxa acima de 75% com 50 veículos, superior a todos os outros protocolos analisados. Verifica-se que o G-Net-10 não apresentou um desempenho ADR tão bom quanto o G-Net-20 com 20, 40 e 50 veículos. Porém, além de conseguir um resultado superior ao G-Net-20 com 30 veículos, ele mantém um comportamento semelhante e sempre superior ao DSR conforme aumenta o número de veículos.

Observa-se, neste cenário, que o comportamento dos protocolos G-Net-10 e G-Net-20 foi bem diferente com 20, 30 e 40 veículos, necessitando fazer uma melhor análise durante o momento em que estes protocolos atualizam suas rotas. Pois, deve-se levar em conta que, apesar das mesmas condições, no momento em que a atualização de rotas é ativada, pode ser que os veículos estejam em boas condições para gerar novas rotas ou não. Entretanto, pode-se aferir que mesmo com diferentes comportamentos, ambos os protocolos G-Net-10 e G-Net-20 apresentaram melhores resultados que o DSR durante toda a variação de veículos, no caso do G-Net-10 e com 20, 30 e 40 veículos, no caso do G-Net-20. Mesmo assim, é possível observar que a partir de 40 veículos eles permanecem sempre melhor que o DSR e com 50 veículos, tanto o G-Net-10 quanto o G-Net-20 ficaram melhor que os outros protocolos AODV e DSR, no ambiente analisado.



**Figura 24: Gráfico de resultados RO do cenário 1 com tráfego CBR ininterrupto**

**Fonte: Autoria própria.**

Quanto a sobrecarga de roteamento, a Figura 24 mostra que, neste cenário, o AODV tem o RO próximo a 12, mais alto que o RO de todos os protocolos analisados. Ele mantém um RO um pouco mais alto com 30 veículos e depois apresenta uma forte crescente, chegando a alcançar um RO superior a 22 com 50 veículos. Isso permite inferir que no ambiente estudado, o AODV tem uma sobrecarga de roteamento muito alta em relação ao DSR e ao protocolo proposto, tanto no caso do G-Net-10 quanto do G-Net-20.

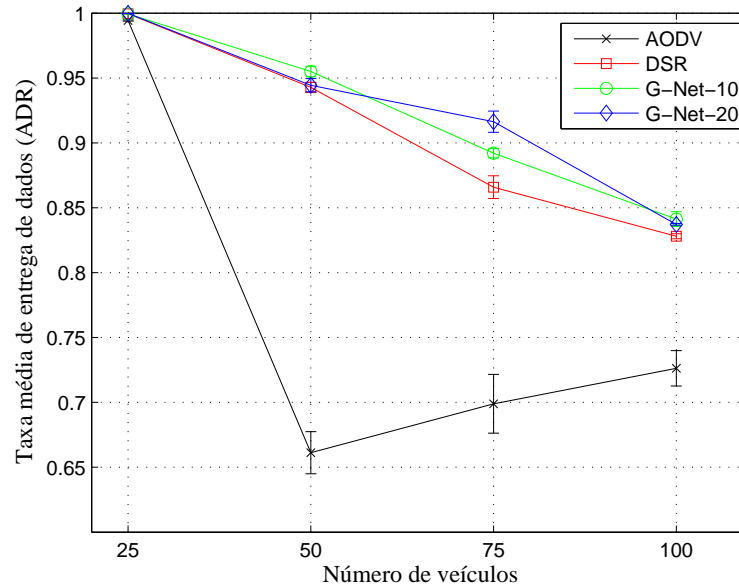
Na Figura 24 é possível perceber ainda, que o G-Net-20 tem o RO superior ao DSR desde 20 até 50 veículos e apesar do G-Net-10 apresentar um RO superior ao G-Net-20 com a alteração de veículos, ambos continuam com o RO inferior ao AODV.

Como era esperado, o DSR apresenta uma sobrecarga de roteamento baixa em relação ao G-Net. Isto se justifica porque o G-Net atualiza rotas periodicamente, enquanto o DSR só ativa a descoberta de rotas quando deseja enviar dados.

### **6.7.3.1.2 RESULTADOS CENÁRIO 2 COM TRÁFEGO CBR ININTERRUPTO**

No cenário de 1000m x 1000m com *range* de comunicação de 250 metros, pode-se observar através da Figura 25 que com 25 veículos, todos os protocolos apresentam uma taxa média de entrega de dados próximo a 100%. Isso se explica porque, como existe pouco

veículo em relação ao tamanho do ambiente, é provável que a maior parte da comunicação ocorra com os poucos nós que se encontram próximos. Principalmente, porque o *range* de alcance das antenas permite mais conexões diretas entre os veículos, reduzindo o número de nós intermediários.



**Figura 25: Gráfico de resultados ADR do cenário 2 com tráfego CBR ininterrupto**

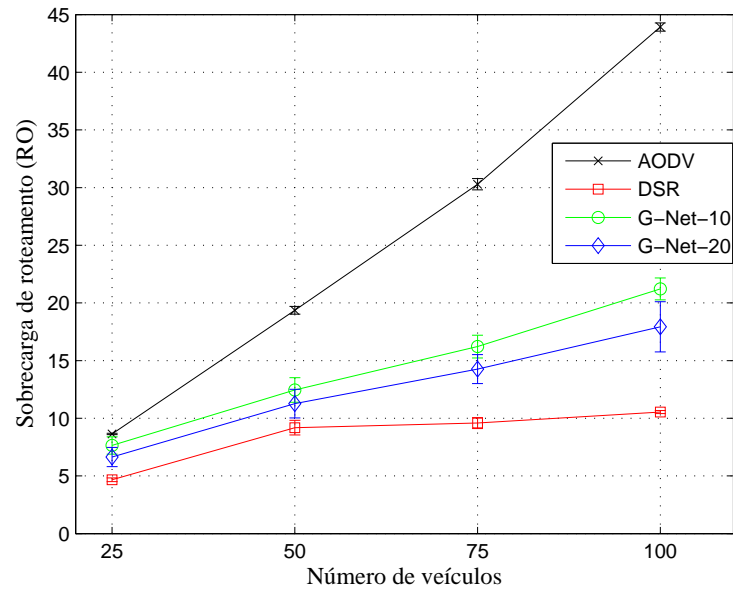
**Fonte: Autoria própria.**

Conforme aumenta o número de veículos, pode-se visualizar na Figura 25 que a taxa média de entrega de dados de todos os protocolos vai diminuindo. Percebe-se uma grande queda de ADR, cujos valores vão de aproximadamente 100% do AODV, com 25 veículos, para aproximadamente 66% com 50 veículos. Depois, o AODV apresenta uma pequena melhora para 75 veículos até chegar a um ADR de 72,62% com 100 veículos.

Os outros protocolos exibem uma leve perda de desempenho ADR, reduzindo de aproximadamente 100% com 25 veículos para aproximadamente 95% com 50 veículos. Com 75 veículos o DSR tem um ADR 86,58% enquanto o G-Net-10 tem 89,20% e o G-Net-20 tem uma taxa média de entrega de 91,63%.

Com 100 veículos, novamente observa-se uma leve queda de desempenho do DSR (82,80%), do G-Net-10 (84,13%) e G-Net-20 (83,72%). Mesmo assim os três protocolos apresentam um ADR bem superior ao AODV (72,62%) e tanto o G-Net-10 quanto o G-Net-20 continuam com taxas superiores ao DSR com 50, 75 e 100 veículos, neste cenário.

Com tráfego CBR ininterrupto, no cenário 2 pode-se visualizar que o AODV tem o RO



**Figura 26: Gráfico de resultados RO do cenário 2 com tráfego CBR ininterrupto**

**Fonte: Autoria própria.**

mais alto que todos os protocolos desde 25 veículos, com RO de 8,64, até 100 veículos, onde se visualiza um RO de 43,92 (Figura 26). Assim como no cenário 1, o DSR mostra boas curvas, com RO de 4,66 para 25 veículos aumentando pouco sua sobrecarga até alcançar 10,54 com 100 veículos.

O G-Net-10 varia seu RO de 7,64 com 25 veículos a 21,21 com 100. Enquanto o G-Net-20, apesar de ser pior do que o DSR, apresenta uma sobrecarga um pouco melhor que o G-Net-10, onde observa-se RO de 6,64 com 25 veículos até 17,93 com 100 veículos.

### 6.7.3.2 RESULTADOS COM PAUSA NO TRÁFEGO CBR

Neste experimento os resultados foram analisados com um tráfego de dados CBR, com pausa de transmissão e possível troca de destino a cada 50 segundos. Com isso, procura-se verificar o desempenho do protocolo proposto em uma situação mais realista de trânsito urbano, onde pausas entre as transmissões podem ser encontradas.

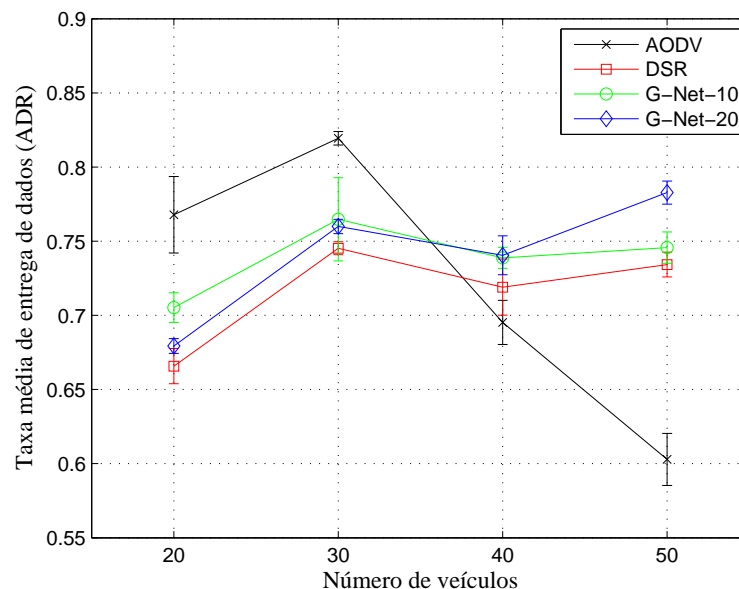
#### 6.7.3.2.1 RESULTADOS CENÁRIO 1 COM PAUSA NO TRÁFEGO DE DADOS CBR

Num cenário mais realista com pausa no tráfego CBR, foi observado que, de modo



geral, os protocolos possuem uma taxa média de entrega de dados superior ao mesmo cenário 1 com tráfego CBR ininterrupto. Assim, enquanto no ambiente com tráfego ininterrupto, o cenário 1 apresenta o menor ADR com 41,21% com 20 veículos e a maior taxa, de 77,19% com 30 veículos, neste mesmo cenário com pausa no tráfego CBR, de acordo com a Figura 27 verifica-se que o menor ADR foi de 60,28% para 50 veículos e o maior ADR foi de 81,94% com 30 veículos.

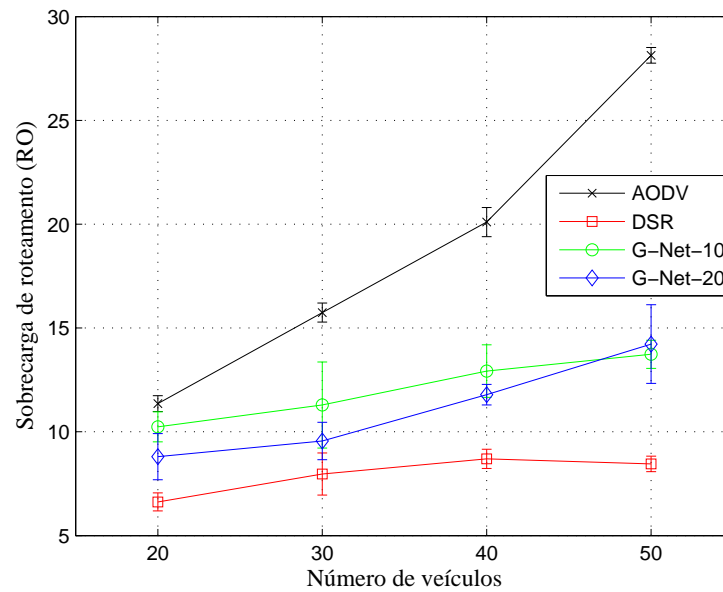
De acordo com a Figura 27 pode-se inferir que o AODV tem um ADR bom de 76,78% para 20 veículos e 81,94% para 30 veículos. Porém, assim como em outros cenários, seu desempenho diminui conforme aumenta o número de veículos. Desta forma, o AODV mostra um resultado inferior com 40 e 50 veículos, onde apresenta taxas de 69,52% e 60,28%, respectivamente.



**Figura 27: Gráfico de resultados ADR do cenário 1 com pausa no tráfego de dados CBR**

**Fonte: A autoria própria.**

Nesta Figura 27, os outros três protocolos apresentam comportamentos bem semelhantes, aumentando a taxa média de entrega de dados de 20 para 30 veículos, com uma leve queda de 30 para 40 veículos, e melhorando o ADR com 50 veículos. Observa-se ainda, que tanto o G-Net-10 quanto o G-Net-20 permanecem com ADR superior ao DSR desde 20 até 50 veículos. Além disso, o G-Net-10 começa melhor que o G-Net-20 com ADR de 70,52% contra 67,93% com 20 veículos, mas com 50 veículos o G-Net-20 (78,28%) supera o G-Net-10 (74,57%), mostrando um desempenho conforme aumenta o número de veículos, neste cenário.



**Figura 28: Gráfico de resultados RO do cenário 1 com pausa no tráfego de dados CBR**

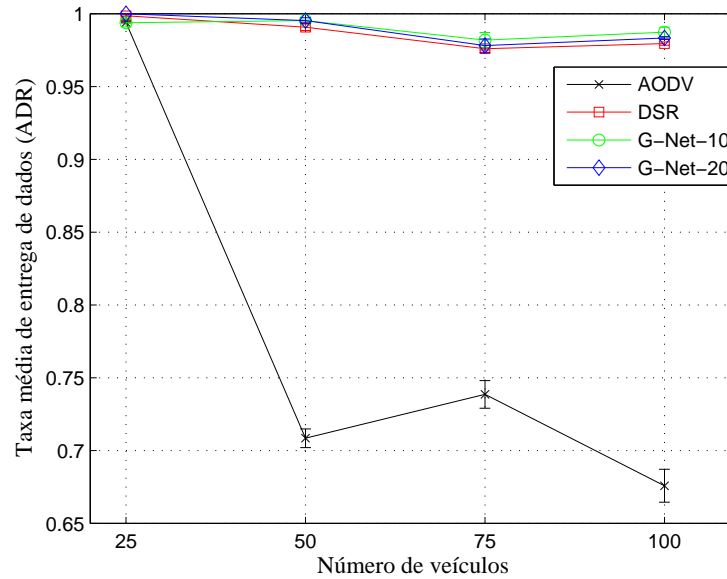
**Fonte: Autoria própria.**

Novamente, através da Figura 28 observa-se que o AODV mostra uma sobrecarga de roteamento superior a todos os protocolos analisados desde 20 veículos (RO=11,36) até 50 veículos (RO=28,13) e o DSR com RO variando de 6,62 até 8,45 tem os melhores resultados. O G-Net-10 continua com ROs superiores ao G-Net-20 de 20 até 40 veículos. Entretanto, neste cenário, o G-Net-20 (RO=14,22) teve uma forte crescente com 50 veículos, apresentando um resultado pior que o G-Net-10 (RO=13,73).

#### **6.7.3.2.2 RESULTADOS CENÁRIO 2 COM PAUSA NO TRÁFEGO DE DADOS CBR**

No cenário 2 com pausa no tráfego de dados CBR, pode-se visualizar que o DSR, o G-Net-10 e o G-Net-20 tiveram ótimo desempenho ADR do início ao fim com uma pequena queda na taxa entre 25 e 100 veículos (Figura 29). O AODV também teve um ótimo ADR com 25 veículos (99,40%). Porém, mostra uma forte queda na curva piorando sua taxa com 50 veículos (70,85%), com uma pequena melhora com 75 veículos (73,86%) reduzindo ainda mais seu ADR com 100 veículos (67,59%), demonstrando que este protocolo tem a tendência de diminuir a taxa média de entrega conforme aumenta o número de veículos neste cenário.

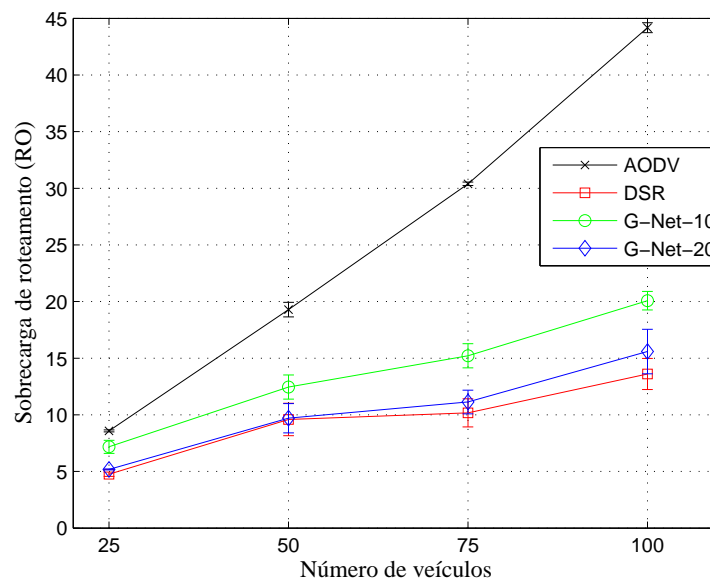
Conforme observado na Figura 29, apesar do G-Net-10 (99,38%) começar com um



**Figura 29: Gráfico de resultados ADR do cenário 2 com pausa no tráfego de dados CBR**

**Fonte: Autoria própria.**

ADR inferior que o DSR (99,86%) e o G-Net-20 (100%), mesmo com taxas bastante elevadas, verifica-se que ele teve um melhor crescimento atingindo um ADR de 98,73%, ligeiramente superior ao G-Net-20 (98,33%) e ao DSR (97,95%).



**Figura 30: Gráfico de resultados RO do cenário 2 com pausa no tráfego de dados CBR**

**Fonte: Autoria própria.**

A Figura 30 mostra que o AODV continua piorando sua sobrecarga de roteamento conforme aumenta o número de veículos, apresentando RO maior do que todos os outros protocolos e variação de 8,58 com 25 veículos a 44,17 com 100 veículos. Neste experimento, o G-Net-20 tem um RO bem baixo, próximo ao DSR do início ao fim do gráfico, variando de 5,18 para 25 veículos a 15,59 para 100 veículos. Mesmo assim, o DSR tem o melhor resultado de RO desde 25 (4,75) até 100 veículos (13,60).

## 7 CONCLUSÃO

Este trabalho apresentou uma visão geral sobre VANET, informando o seu conceito, classificações, sua importância e aplicações. No documento foram apresentados alguns problemas enfrentados pelas VANETs e informado que, basicamente, esses problemas são decorrentes da complexidade dos movimentos dos veículos que os protocolos MANETs não conseguem resolver. Entretanto, ao analisá-los de acordo com as camadas do modelo OSI, o texto demonstra que, normalmente, os pesquisadores estão buscando soluções VANETs realizando modificações em protocolos da camada física, enlace, rede e transporte. Mas algumas soluções de protocolos totalmente novos também estão sendo criados especificamente para as redes veiculares.

Quanto às ferramentas de simulação, o trabalho explica a diferença entre os geradores de mobilidade, os simuladores de rede e os simuladores VANET. Além de apresentar alguns exemplos destas ferramentas, descreve as principais características e permite entender as vantagens e desvantagens de cada uma delas.

Foi apresentada a proposta de um novo protocolo de roteamento VANET, batizado de G-Net, criado com base no DSR, com atualizações periódicas e otimização de rotas através de algoritmo genético. O protocolo proposto (G-Net) foi simulado em diferentes ambientes para analisar sua taxa média de entrega e sobrecarga de roteamento em relação ao AODV e o DSR. Os experimentos foram realizados com área de 600m x 600m com *range* de comunicação de 100m e área de 1000m x 1000m e *range* de 250m. Ambos os cenários foram testados com tráfego de dados CBR ininterrupto do início ao fim das simulações e com pausa no tráfego CBR para representar uma situação mais dinâmica.

Os cenários foram criados com a união do VanetMobiSim e o ns-3, adotando a mobilidade com mudança de faixas, frenagem, aumento e diminuição de velocidades dos nós (veículos) em um ambiente tipicamente urbano. Com variação de velocidade até 60km/h e semáforos, com o intuito de analisar os resultados numa situação mais próxima da realidade.

Neste trabalho, o G-Net foi implementado para realizar atualização e otimização de

rotas de duas formas diferentes: Ativando a descoberta e o algoritmo genético a cada 10 pacotes transmitidos (G-Net-10) e num segundo caso, ativando a descoberta e o algoritmo genético a cada 20 pacotes transmitidos (G-Net-20).

De acordo com os experimentos realizados foi possível compreender a importância do *range* de comunicação. Neste sentido, pôde-se inferir que no caso dos cenários com *range* de 100m, as taxas médias de entrega, de maneira geral, tiveram os valores ADR quase sempre inferiores ao cenário com *range* de 250m. Isto se explica porque, devido a configuração do ambiente com variação de velocidade e semáforos, os carros podem se afastar para distâncias superiores a 100m com mais frequência do que em ambientes com velocidades constantes. Além disso, o autor Benin et al. (2012) apresenta um gráfico no qual os pacotes recebidos reduzem drasticamente conforme aumenta a distância com modelo Nakagami-m, em ambiente urbano e no ns-3.

Como foi implementado um sistema de atualização de rotas com envio periódico de pacotes de controle durante o envio de pacotes de dados, o G-Net apresentou uma sobrecarga de roteamento ruim em relação ao DSR. Relativamente a sobrecarga, mesmo com o aumento, o G-Net foi sempre melhor que o AODV em todos os cenários analisados e, em alguns casos, foi observado um RO bem inferior ao AODV e apenas um pouco acima do DSR dependendo do cenário e do número de veículos.

O trabalho mostrou que o protocolo proposto teve o melhor desempenho de taxa de entrega em quase todas as situações, e conforme aumentou-se o número de veículos seu desempenho foi melhorando em relação ao DSR e ao AODV. Além disso, os resultados mostraram que no cenário de 600m x 600m tanto o G-Net-10 quanto o G-Net-20 apresentaram as melhores taxas ADR com 50 veículos, com e sem pausa no tráfego de dados e acima de 40 veículos foram sempre melhores que o DSR. Proporcionalmente, ambos G-Net-10 e G-Net-20, também foram superiores ao AODV e ao DSR a partir de 50 veículos em todos os cenários testados com área de 1000m x 1000m.

## 7.1 TRABALHOS FUTUROS

As análises e conclusões mostraram que o protocolo proposto apresenta boas taxas de entrega de dados, principalmente em ambientes acima de 40 veículos e espera-se que novos avanços tragam resultados ainda mais promissores. Porém, os cenários e métricas avaliados não permitem averiguar as deficiências e todo o potencial que o G-Net pode proporcionar. Fica claro que melhorias são possíveis e algumas situações ainda merecem ser investigadas, tais como:

- Avaliar novas métricas, como por exemplo, atraso médio fim-a-fim, vazão (*throughput*) e *goodput* podem mostrar maiores vantagens do G-Net. Somente uma investigação destas métricas pode confirmar esta teoria.
- Avaliar o comportamento do G-Net no caso de alterações nas taxas de tráfego. Num ambiente real podem haver diversos tipos de tráfego de dados que devem alterar significativamente o comportamento do protocolo. Com alterações nas taxas de tráfego pode-se verificar se o protocolo apresenta deficiências ou se existe melhora no desempenho de acordo com as métricas analisadas.
- Diferentes formas de seleção, *crossover* e mutação do algoritmo genético também podem ser propostas e analisadas. Alterações internas no algoritmo genético podem proporcionar melhores resultados. Novamente, apenas uma análise mais detalhada pode confirmar esta teoria e por isso recomenda-se maiores estudos com este objetivo.
- Efetuar simulações com mais veículos em diferentes cenários e comparar o G-Net com mais protocolos pode mostrar seu potencial num sentido mais amplo. Os experimentos com mais veículos permitem verificar se os resultados tendem a estabilizar no decorrer das simulações e a comparação com outros protocolos pode proporcionar uma visão mais abrangente de seu desempenho em relação a outras soluções já conhecidas.
- Atualizar e otimizar rotas a cada  $Np$  pacotes transmitidos tem influência direta, tanto na evolução das rotas geradas quanto na sobrecarga de roteamento. Aumentar o valor de  $Np$  pode reduzir a sobrecarga de roteamento e diminuí-lo pode proporcionar melhores resultados em outras métricas analisadas.

## REFERÊNCIAS

- BEHRISCH, M. et al. Sumo-simulation of urban mobility-an overview. In: IARIA. **SIMUL 2011, The Third International Conference on Advances in System Simulation**. Barcelona, 2011. p. 55–60.
- BENIN, J.; NOWATKOWSKI, M.; OWEN, H. Vehicular Network simulation propagation loss model parameter standardization in ns-3 and beyond. In: IEEE. **Southeastcon, 2012 Proceedings of IEEE**. Orlando, 2012. p. 1–5.
- BERNSEN, J.; MANIVANNAN, D. Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification. **Pervasive and Mobile Computing**, Elsevier, v. 5, n. 1, p. 1–18, 2009.
- BHAKTHAVATHSALAM; NAYAK, S. Operational inferences on VANETs in 802.16e and 802.11p with improved performance by Congestion Alert. In: IEEE. **Consumer Communications and Networking Conference (CCNC), 2011 IEEE**. Las Vegas, 2011. p. 467–471.
- BISOYI, S. K.; SAHU, S. Performance analysis of Dynamic MANET On-demand (DYMO) Routing protocol. **Special Issue of IJCCT**, v. 1, n. 2, p. 3, 2010.
- BITAGHSIR, S. A. **An Intelligent Routing Protocol for Delay Tolerant Networks using Genetic Algorithm**. Dissertação (Mestrado) — Sharif University of Technology, 2009.
- BLUM; ESKANDARIAN, A.; HOFFMAN, L. Challenges of intervehicle ad hoc networks. **Intelligent Transportation Systems, IEEE Transactions on**, v. 5, n. 4, p. 347–351, 2004.
- BURIOL, L. S. et al. A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. **Networks**, Wiley Online Library, v. 46, n. 1, p. 36–56, 2005.
- CANUMOBISIM. **Canu Project**. <http://canu.informatik.uni-stuttgart.de/mobisim/index.html>. Setembro 2013. Disponível em: <<http://canu.informatik.uni-stuttgart.de>>.
- CAVALCANTI, S. R. **Veer: Um algoritmo de seleção de pares em redes ad hoc veiculares**. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2008.
- CHANDRAN, K. et al. A feedback-based scheme for improving TCP performance in ad hoc wireless networks. **IEEE Pers Commun**, v. 8, n. 1, p. 34–39, 2001.
- CHEN, Y.-S.; LIN, Y.-W.; LEE, S.-L. A mobicast routing protocol in vehicular ad-hoc networks. **Mobile Networks and Applications**, Springer-Verlag New York, Inc., v. 15, n. 1, p. 20–35, 2010.
- CHOWDHURY, S. I. et al. Performance evaluation of reactive routing protocols in VANET. In: IEEE. **Communications (APCC), 2011 17th Asia-Pacific Conference on**. Sabah, 2011. p. 559–564.



CORREIA; CELESTINO, J.; CHERKAOUI, O. Mobility-aware Ant Colony Optimization routing for vehicular ad hoc networks. In: IEEE. **Wireless Communications and Networking Conference (WCNC), 2011 IEEE**. Cancun, 2011. p. 1125–1130.

COSTA, H. F. **Redes veiculares de emergência**. Dissertação (Mestrado) — 74 f. Universidade Técnica de Lisboa (UTL). Instituto Superior Técnico, Lisboa, 2010.

DONGRE, M. M.; BAWANE, N. G.; JAWADE, N. R. Routing Protocols for V2V communications used in Urban VANET to improve ITS. **IJCA Proceedings on National Conference on Innovative Paradigms in Engineering and Technology (NCIPET 2012)**, ncipet, n. 6, p. 24–28, March 2012. Published by Foundation of Computer Science, New York, USA.

FONSECA, A.; CAMÕES, A.; VAZÃO, T. Geographical routing implementation in NS3. In: ICST (INSTITUTE FOR COMPUTER SCIENCES, SOCIAL-INFORMATICS AND TELECOMMUNICATIONS ENGINEERING). **Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques**. Lugano, 2012. p. 353–358.

FREESIM. **FREESIM**. <http://www.freewaysimulator.com/>. Agosto 2013. Disponível em: <<http://www.freewaysimulator.com>>.

GREFENSTETTE, J. J. et al. Genetic algorithms for changing environments. In: ACM. **PPSN**. Brussels, 1992. v. 2, p. 137–144.

GUNASEKAR, M.; HINDUJA, S. Automatic Tuning Of OLSR Routing Protocol Using IWD in VANET. **International Journal of Innovative Research in Computer and Communication Engineering**, v. 2, n. 1, 2014.

HARRI, J.; FIORE, M. VanetMobiSim–Vehicular Ad hoc Network mobility extension to the CanuMobiSim framework. **Institut Eurécom Department of Mobile Commu**, v. 6904, 2006.

HÄRRI, J. et al. Vehicular mobility simulation with VanetMobiSim. **Simulation**, Sage Publications, 2009.

HE et al. Reliable Blast UDP: predictable high performance bulk data transfer. In: IEEE. **Cluster Computing, 2002. Proceedings. 2002 IEEE International Conference on**. Chicago, 2002. p. 317–324.

HISTORY, N. C. **NS: CHANGE HISTORY**. <http://www.isi.edu/nsnam/ns/CHANGES.html>. Agosto 2013. Disponível em: <<http://www.isi.edu>>.

HOLLAND, G.; VAIDYA, N. Analysis of TCP performance over mobile ad hoc networks. **Wireless Networks**, Springer-Verlag New York, Inc., v. 8, n. 2/3, p. 275–288, 2002.

HONG, X.; XU, K.; GERLA, M. Scalable routing protocols for mobile ad hoc networks. **Network, IEEE**, v. 16, n. 4, p. 11–21, 2002.

HUANG, J.; LIU, Y. MOEAQ: A QoS-aware multicast routing algorithm for MANET. **Expert Systems with Applications**, Elsevier, v. 37, n. 2, p. 1391–1399, 2010.

IEEE. **IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages**. 2006.

- IEEE. **IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services**. 2010. 1–144 p.
- IEEE. **IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-channel Operation**. 2011. 1–89 p.
- IWATA, A. et al. Scalable routing strategies for ad hoc wireless networks. **Selected Areas in Communications, IEEE Journal on, IEEE**, v. 17, n. 8, p. 1369–1379, 1999.
- JIANG et al. Design of 5.9 Ghz DSRC-based vehicular safety communication. **Wireless Communications, IEEE**, v. 13, n. 5, p. 36–43, 2006.
- JOHNSON, D. et al. **The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4**. RFC 4728, 2007.
- JOHNSON, D. A.; MALTZ, D. B.; BROCH, J. DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. **Computer Science Department Carnegie Mellon University Pittsburgh, PA**, p. 15213–3891, 2001.
- JOSHI, H. P.; SICHITIU, M. L.; KIHLE, M. Distributed robust geocast multicast routing for inter-vehicle communication. In: **INL. Proceedings of WEIRD workshop on WiMax, wireless and mobility**. Coimbra, 2007. p. 9–21.
- KAMINI, K.; KUMAR, R. Vanet parameters and applications: A review. **Global Journal of Computer Science and Technology**, v. 10, n. 7, 2010.
- KAMOLTHAM; NAKORN, K. N.; ROJVIBOONCHAI, K. From NS-2 to NS-3 - Implementation and evaluation. In: **IEEE. Computing, Communications and Applications Conference (ComComAp), 2012**. Hong Kong, 2012. p. 35–40.
- KARNADI; MO, Z. H.; LAN, K.-C. Rapid Generation of Realistic Mobility Models for VANET. In: **IEEE. Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE**. Hong Kong, 2007. p. 2506–2511.
- KAUR, J.; SINGH, E. P. Performance Comparison Between Unicast and Multicast Protocols VANETS. **International Journal of Advanced Technology & Engineering Research (IJATER)**, 2013.
- KESTING, A.; TREIBER, M.; HELBING, D. General lane-changing model MOBIL for car-following models. **Transportation Research Record: Journal of the Transportation Research Board**, Trans Res Board, v. 1999, n. 1, p. 86–94, 2007.
- KHORASHADI, B. et al. Impact of Transmission Power on the Performance of UDP in Vehicular Ad Hoc Networks. In: **IEEE. Communications, 2007. ICC '07. IEEE International Conference on**. Glasgow, 2007. p. 3698–3703.
- KIM, D.; TOH, C. K.; CHOI, Y. TCP-BuS: improving TCP performance in wireless ad hoc networks. In: **IEEE. Communications, 2000. ICC 2000. 2000 IEEE International Conference on**. New Orleans, 2000. v. 3, p. 1707–1713.
- KONISHI et al. MobiREAL simulator-evaluating MANET applications in real environments. In: **IEEE. Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005. 13th IEEE International Symposium on**. Atlanta, 2005. p. 499–502.

- KOPPARTY et al. Split TCP for mobile ad hoc networks. In: IEEE. **Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE**. Taipei, 2002. v. 1, p. 138–142.
- KORKMAZ, G. et al. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In: ACM. **Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks**. Philadelphia, 2004. p. 76–85.
- LABIOD, H.; ABABNEH, N.; FUENTE, M. G. d. l. An efficient scalable trajectory based forwarding scheme for VANETs. In: IEEE. **Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on**. Perth, 2010. p. 600–606.
- LEE, K. C.; LEE, U.; GERLA, M. Survey of routing protocols in vehicular ad hoc networks. **Advances in vehicular ad-hoc networks: Developments and challenges**, Information Science Reference (IGI Global), p. 149–170, 2010.
- LEE, U. et al. Code torrent: content distribution using network coding in vanet. In: ACM. **Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking**. Los Angeles, 2006. p. 1–5.
- LI, B. et al. Wireless Access for Vehicular Environments. **Internet: <http://www.mehrpouyan.info/Projects/Group>**, v. 205, 2010.
- LI, F.; WANG, Y. Routing in vehicular ad hoc networks: A survey. **Vehicular Technology Magazine, IEEE**, v. 2, n. 2, p. 12–22, 2007.
- LI, Y. J. An overview of the DSRC/WAVE technology. In: **Quality, Reliability, Security and Robustness in Heterogeneous Networks**. Houston: Springer, 2012. p. 544–558.
- LIN, Y.-W.; CHEN, Y.-S.; LEE, S.-L. Routing Protocols in Vehicular Ad Hoc Networks: A Survey and Future Perspectives. **J. Inf. Sci. Eng.**, v. 26, n. 3, p. 913–932, 2010.
- LOCHERT, C. et al. A routing strategy for vehicular ad hoc networks in city environments. In: IEEE. **Intelligent Vehicles Symposium, 2003. Proceedings. IEEE**. Columbus, 2003. p. 156–161.
- LU, K.; QIAN, Y.; CHEN, H.-H. A Secure and Service-Oriented Network Control Framework for WiMAX Networks. **IEEE Communications Magazine**, v. 45, n. 5, p. 124–130, 2007.
- MANGHARAM et al. GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle Networks. In: IEEE. **Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on**. San Jose, 2006. p. 1–8.
- MAOWAD, H.; SHAABAN, E. Efficient routing protocol for Vehicular Ad hoc networks. In: IEEE. **Networking, Sensing and Control (ICNSC), 2012 9th IEEE International Conference on**. Beijing, 2012. p. 209–215.
- MARTINEZ et al. CityMob: A Mobility Model Pattern Generator for VANETs. In: IEEE. **Communications Workshops, 2008. ICC Workshops '08. IEEE International Conference on**. Beijing, 2008. p. 370–374.

MARTINEZ, F. J. et al. A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). **Wireless Communications and Mobile Computing**, Wiley Online Library, v. 11, n. 7, p. 813–828, 2011.

MENOUAR, H.; LENARDI, M.; FILALI, F. An intelligent movement-based routing for VANETs. In: ITS (UK). **ITS world congress**. London, 2006.

MIN, Y. U. H. et al. New Pilot Designs and ICI Mitigation for OFDM Downlink Systems based on IEEE 802.16m Standards over High Speed Vehicular Channels. In: IEEE. **Wireless Communications and Networking Conference, 2009. WCNC 2009**. IEEE. Budapest, 2009. p. 1–5.

MITCHELL, M. **An introduction to genetic algorithms**. [S.l.]: MIT press, 1998.

MOBIREAL. **MobiREAL. A REALISTIC NETWORK SIMULATION**. <http://www.mobireal.net/>. Agosto 2013. Disponível em: <<http://www.mobireal.net/>>.

MONARCH. **MONARCH PROJECT**. <http://www.monarch.cs.rice.edu/>. Agosto 2013. Disponível em: <<http://www.monarch.cs.rice.edu/>>.

NAGARAJ, U.; DHAMAL, P. Broadcasting Routing Protocols in VANET. **Network and Complex Systems**, v. 1, n. 2, p. 13–19, 2012.

NAMBOODIRI, V.; AGARWAL, M.; GAO, L. A study on the feasibility of mobile gateways for vehicular ad-hoc networks. In: ACM. **Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks**. Philadelphia, 2004. p. 66–75.

NCTUNS. **NCTUns 6.0 NETWORK SIMULATOR AND EMULATOR**. <http://NSL.csie.nctu.edu.tw/nctuns.html>. Agosto 2013. Disponível em: <<http://NSL.csie.nctu.edu.tw/>>.

ns-2. **The Network Simulator - ns-2**. <http://www.isi.edu/nsnam/ns>. August 2013. Disponível em: <<http://www.isi.edu/nsnam/ns>>.

NS-3 RASTREAMENTO. **NS-3 RASTREAMENTO, VERSÃO NS-3.16**. <http://www.nsnam.org/docs/release/3.16/tutorial-pt-br/ns-3-tutorial.pdf>. Março 2013. Disponível em: <<http://www.nsnam.org/>>.

PAUL, B.; IBRAHIM, M.; BIKAS, M. A. N. Article: Vanet routing protocols: Pros and cons. **International Journal of Computer Applications**, v. 20, n. 3, p. 28–34, April 2011.

PAUL, B.; IBRAHIM, M.; BIKAS, M. A. N. Experimental Analysis of AODV & DSR over TCP & CBR Connections with Varying Speed and Node Density in VANET. **International Journal of Computer Applications**, v. 24, n. 4, p. 30–37, June 2011.

PEI, G.; GERLA, M.; CHEN, T.-W. Fisheye state routing: a routing scheme for ad hoc wireless networks. In: IEEE. **Communications, 2000. ICC 2000. 2000 IEEE International Conference on**. New Orleans, 2000. v. 1, p. 70–74.

PENG, L.; BO, Z.; ZHONGYONG, Z. Simulation of VANET in a More Realistic Scenario. In: IEEE. **Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on**. Wuhan, 2011. p. 1–3.

PERKINS, C. et al. Rfc 3561-ad hoc on-demand distance vector (AODV) routing. **Internet RFCs**, p. 1–38, 2003.

RANI, P.; SHARMA, N.; SINGH, P. Performance comparison of VANET routing protocols. In: IEEE. **Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on**. Wuhan, 2011. p. 1–4.

ROYER, E. M.; TOH, C.-K. A review of current routing protocols for ad hoc mobile wireless networks. **IEEE Pers Commun**, v. 6, n. 2, p. 46–55, 1999.

SCHMILZ, R. et al. Analysis of Path Characteristics and Transport Protocol Design in Vehicular Ad Hoc Networks. In: IEEE. **Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd**. Melbourne, 2006. v. 2, p. 528–532.

SCHMITZ, R. **Advanced transport protocols for wireless and mobile ad hoc networks**. 1-180 p. Tese (Doutorado) — University of Mannheim, 2007.

SILVA, R. C. d. **Otimização multiobjetivo baseada em colônia de formigas para o roteamento e consumo de energia em redes de sensores sem fio**. Dissertação (Mestrado) — 110 p. Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, 2008.

SINGH, P. K.; LEGO, K.; TUITHUNG, D. T. Simulation based Analysis of Ad hoc Routing Protocol in Urban and Highway Scenario of VANET. **International Journal of Computer Applications**, v. 12, n. 10, p. 42–49, January 2011. Published by Foundation of Computer Science.

SKORDYLIS, A.; TRIGONI, N. Delay-bounded routing in vehicular ad-hoc networks. In: ACM. **Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing**. Hong Kong, 2008. p. 341–350.

SPAHO, E. et al. Performance of OLSR and DSDV protocols in a VANET scenario: evaluation using CAVENET and NS3. In: IEEE COMPUTER SOCIETY. **Proceedings of the 2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications**. Victoria, 2012. p. 108–113.

SPAHO, E. et al. Performance Evaluation of OLSR and AODV Protocols in a VANET Crossroad Scenario. In: IEEE. **Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on**. Barcelona, 2013. p. 577–582.

STRAW. **STRAW - STrEet RANdom Waypoint - vehicular mobility model for network simulations**. <http://www.aqualab.cs.northwestern.edu/projects/144-straw-street-random-waypoint-vehicular-mobility-model-for-network-simulations-e-g-car-networks>. Agosto 2013. Disponível em: <<http://www.aqualab.cs.northwestern.edu>>.

TANENBAUM ANDREW S.; WETHERALL, D. **Redes de computadores**. 5a. ed. São Paulo: Pearson, 2011.

TANOMARU, J. Motivação, fundamentos e aplicações de algoritmos genéticos. In: III ESCOLA DE REDES NEURAIAS. **Anais do II Congresso Brasileiro de Redes Neurais**. Curitiba, 1995. v. 1.

THAKUR, A. et al. Design Approach for VANET Routing Using Real Time Vehicular Traffic Information. **International Journal of Scientific & Engineering Research**, v. 4, n. Issue 5, may 2013.

TIGER. **TIGER Products**. <http://www.census.gov/geo/maps-data/data/tiger.html>. Agosto 2013. Disponível em: <<http://www.census.gov>>.

TOOR, Y.; MUHLETHALER, P.; LAOUITI, A. Vehicle Ad Hoc networks: applications and related technical issues. **IEEE Communications Surveys & Tutorials**, v. 10, n. 3, p. 74–88, 2008.

VANETMOBISIM. **VanetMobiSim Project**. <http://vanet.eurecom.fr>. September 2013. Disponível em: <<http://vanet.eurecom.fr>>.

VAUGHAN-NICHOLS, S. J. Mobile WiMAX: The next wireless battle ground. **Computer, IEEE**, v. 41, n. 6, p. 16–18, 2008.

WANG, Y.-B. et al. A Novel Geographic Routing Strategy over VANET. In: **IEEE. Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on**. Perth, 2010. p. 873–879.

XU, C. H. et al. A unified analysis of routing protocols in MANETs. **Communications, IEEE Transactions on**, v. 58, n. 3, p. 911–922, 2010.

YANG, Q.; LIM, A.; AGRAWAL, P. Connectivity Aware Routing in Vehicular Networks. In: **IEEE. Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE**. Las Vegas, 2008. p. 2218–2223.

YU, X.; TANG, K.; YAO, X. Immigrant schemes for evolutionary algorithms in dynamic environments: Adapting the replacement rate. **Science China Information Sciences**, Springer, v. 54, n. 7, p. 1352–1364, 2011.