



**CAIRO APARECIDO CAMPOS**

**RHBS: ROTEAMENTO HIERÁRQUICO BASEADO EM  
SETORES PARA REDES VEICULARES**

**LAVRAS – MG**

**2020**

**CAIRO APARECIDO CAMPOS**

**RHBS: ROTEAMENTO HIERÁRQUICO BASEADO EM SETORES PARA REDES  
VEICULARES**

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

Prof. DSc. Tales Heimfarth

Orientador

Prof. DSc. João Carlos Giacomini

Coorientador

**LAVRAS – MG**

**2020**

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

Campos, Cairo Aparecido.

RHBS: Roteamento Hierárquico Baseado em Setores para  
Redes Veiculares / Cairo Aparecido Campos. - 2020.

85 p. : il.

Orientador(a): Prof. DSc. Tales Heimfarth.

Coorientador(a): Prof. DSc. João Carlos Giacomini.

Dissertação (mestrado acadêmico) - Universidade Federal  
de Lavras, 2020.

Bibliografia.

1. Redes Veiculares. 2. VANET. 3. Roteamento em  
Ambiente Urbano. I. Heimfarth, Tales. II. Giacomini, João  
Carlos. III. Título.

**CAIRO APARECIDO CAMPOS**

**RHBS: ROTEAMENTO HIERÁRQUICO BASEADO EM SETORES PARA REDES VEICULARES**

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

APROVADA em 30 de Novembro de 2020.

Prof. DSc. Raphael Winckler de Bettio UFLA  
Prof. DSc. Edison Pignaton de Freitas UFRGS

Prof. DSc. Tales Heimfarth  
Orientador

Prof. DSc. João Carlos Giacomini  
Co-Orientador

**LAVRAS – MG  
2020**

*Dedico este trabalho a todas as pessoas que acreditam na educação e na ciência como processos de transformação.*

## **AGRADECIMENTOS**

A minha família que sempre me deu força e incentivo para a realização dos meus estudos.

Ao IFSULDEMINAS – Campus Machado que possibilitou o afastamento das minhas atividades para a realização do mestrado.

A Gerência de Tecnologia da Informação e Comunicação (GTIC) pelo apoio e compreensão durante o tempo que estive ausente.

Aos professores Tales Heimfarth e João Carlos Giacomini por todos os ensinamentos e orientações dadas durante o projeto.

A Diego Vinícius Natividade, um verdadeiro amigo e companheiro de mestrado, que sempre esteve presente para discutir assuntos relacionados a programação, redes veiculares, matemática e corpos celestes.

A todos colegas de mestrado e do GRUBi que me possibilitaram boas risadas e novos conhecimentos que vão muito além da área da ciência da computação.

A Universidade Federal de Lavras, especialmente ao Programa de Pós-Graduação em Ciência da Computação, pela oportunidade.

A Deus, o maior arquiteto, matemático e programador que já existiu!

*O que sabemos é uma gota, o que ignoramos é um oceano.*

*(Isaac Newton)*

## RESUMO

Em redes veiculares *ad hoc*, uma das maiores dificuldades encontradas é manter a comunicação estável entre os veículos. A alta mobilidade e topologia dinâmica dos veículos interrompem as rotas estabelecidas entre dois veículos em comunicação. Para tentar minimizar esse problema, um grande número de protocolos de roteamento foram desenvolvidos e classificados de acordo com suas propriedades. Entre eles, os protocolos baseados em cluster e os protocolos geográficos ganharam destaque. Essas duas classes de protocolos, em geral, permitem um menor *overhead* de mensagens de controle e um menor tempo de latência. No entanto, nem sempre estão adaptados às características únicas das cidades. Visando um melhor desempenho para o ambiente urbano, este trabalho propõe o protocolo de *Roteamento Hierárquico Baseado em Setores* (RHBS). Neste protocolo cada cluster é um setor delimitado por um conjunto de ruas que possuem estacionamentos. Os veículos estacionados nestes estacionamentos formam um backbone que interligam todos os setores. O backbone é utilizado para rotear de forma geográfica, mensagens de controle e dados. Em cada setor há também um veículo estacionado com a função de *Cluster Head* (CH). O CH recebe a posição dos veículos móveis do seu setor e fornece a posição de um veículo quando requisitada. Além deste veículo de status especial, há também um veículo estacionado com o status de *Cluster Head de Segundo Nível* (CH2) próximo ao centro da área geográfica onde se localiza os setores. Este veículo recebe tabelas de atualização dos CHs e portanto, conhece todos os veículos em movimento em cada setor. O CH2 também é consultado quando a posição de um veículo não é conhecida por um CH. No RHBS, quando um veículo móvel de origem recebe a posição de um veículo móvel de destino por meio de um CH ou CH2, envia uma mensagem de dados em direção a esta posição. O RHBS após seu desenvolvimento foi comparado com o protocolo reativo *Ad-hoc On-Demand Distance Vector* (AODV) mostrando um menor tempo de latência e uma menor quantidade de mensagens em quatro das cinco comunicações avaliadas.

**Palavras-chave:** Rede Veicular, VANET, Backbone, Roteamento Baseado em Cluster, Roteamento Geográfico, Ambiente Urbano.

## ABSTRACT

In vehicular networks ad hoc, one of the biggest difficulties encountered is maintaining stable communication between vehicles. The high mobility and dynamic topology of the vehicles interrupt routes established between two communicating vehicles. To try to minimize this problem, a large number of routing protocols have been developed and classified according to their properties. Among them, cluster-based protocols and geographical protocols have gained prominence. These two classes of protocols, in general, allow for a smaller overhead of control messages and a shorter latency time. However, they are not always adapted to the unique characteristics of cities. Aiming at better performance for the urban environment, this work proposes the *Sector-Based Hierarchical Routing* (SBHR) protocol. In this protocol, each cluster is a sector bounded by a set of streets that have parking lots. Vehicles parked in these car parks form a backbone that interconnects all sectors. The backbone is used to route control messages and data messages geographically. In each sector, there is also a vehicle parked with the Cluster Head (CH) function. The CH receives the mobile vehicles' position in its sector and provides a vehicle's position when requested. In addition to this special status vehicle, there is also a vehicle parked with Second Level Cluster Head (CH2) status close to the center of the geographical area where the sectors are located. This vehicle receives update tables from the CHs and therefore knows all the vehicles in motion in each sector. CH2 is also consulted when a CH does not know a vehicle's position. In the SBHR, when a source mobile vehicle receives a destination mobile vehicle's position via a CH or CH2, it sends a data message towards this position. The SBHR after its development was compared with the reactive Ad-hoc On-Demand Distance Vector (AODV) protocol showing a shorter latency time and a smaller number of messages in four of the five evaluated communications.

**Keywords:** Vehicular Network, VANET, Backbone, Cluster-based Routing, Geographic Routing, Urban Environment.

## LISTA DE FIGURAS

Figura 2.1 – Tipos de Comunicação . . . . .	24
Figura 2.2 – Arquitetura WAVE . . . . .	26
Figura 2.3 – Algoritmo de Cluster <i>Cluster Based Location Routing</i> (CBLR) . . . . .	28
Figura 2.4 – Taxonomia de Protocolos de Roteamento em <i>Vehicular Ad hoc Network</i> (VANET) . . . . .	31
Figura 4.1 – Troca de Mensagens Usando o CH . . . . .	45
Figura 4.2 – Troca de Mensagens Usando CH e CH2 . . . . .	45
Figura 4.3 – Algoritmo 1: Descoberta e Atualização de CH . . . . .	47
Figura 4.4 – Anúncio de Posição . . . . .	48
Figura 4.5 – Etapas do Encaminhamento da Mensagem <i>Hello</i> . . . . .	50
Figura 4.6 – Envio de uma Mensagem <i>Hello</i> para o CH do Setor . . . . .	51
Figura 4.7 – Etapas de Encaminhamento da Requisição até um CH . . . . .	52
Figura 4.8 – Etapas de Encaminhamento da Requisição até um CH2 . . . . .	53
Figura 4.9 – Etapas do Encaminhamento da Mensagem de Resposta . . . . .	54
Figura 4.10 – Envio de uma Mensagem Solicitando a Posição do Veículo Móvel de Destino	55
Figura 4.11 – Envio de uma Mensagem Solicitando a Posição do Veículo Móvel de Destino para o CH2 . . . . .	55
Figura 4.12 – Resposta do CH para o Veículo Móvel de Origem . . . . .	56
Figura 4.13 – Resposta do CH2 para o Veículo Móvel de Origem . . . . .	56
Figura 4.14 – Etapas do Encaminhamento da Mensagem de Dados . . . . .	57
Figura 4.15 – Envio de Mensagem do Veículo Móvel de Origem para o Veículo Móvel de Destino em um Mesmo Setor . . . . .	58
Figura 4.16 – Envio de Mensagem do Veículo Móvel de Origem para o Veículo Móvel de Destino em Setores Diferentes . . . . .	58
Figura 4.17 – Algoritmo 2: Descoberta de um Nó Próximo a uma Interseção . . . . .	61
Figura 4.18 – Roteamento Geográfico com Alteração de Modos . . . . .	61
Figura 4.19 – Máximo Local no Envio de Mensagem do CH para o Veículo Móvel de Origem . . . . .	62
Figura 4.20 – Máximo Local no Envio de Mensagem do Veículo Móvel de Origem para o Veículo Móvel de Destino . . . . .	63
Figura 5.1 – Cenário com CHs Utilizado pelo RHBS . . . . .	66

Figura 6.1 – Latência Fim-a-fim . . . . .	70
Figura 6.2 – Média Geral da Quantidade de Mensagens para cada Comunicação . . . . .	71
Figura 6.3 – Média de Saltos - Entrega da Mensagem de Dados . . . . .	71
Figura 6.4 – Média de Requisições Recebidas . . . . .	72
Figura 6.5 – Latência Média no Envio de Mensagens, com o Uso do CH2 e Sem o Uso do CH2 . . . . .	73
Figura 6.6 – Média de Saltos . . . . .	74
Figura 6.7 – Quantidade Média de Mensagens . . . . .	74
Figura 6.8 – Latência média no envio de mensagens, com o uso do CH2 e sem o uso do CH2 . . . . .	75
Figura 6.9 – Média de Saltos . . . . .	76
Figura 6.10 – Média da Quantidade de Mensagens . . . . .	76
Figura 6.11 – Latência Fim-a-fim . . . . .	77
Figura 6.12 – Média Geral da Quantidade de Mensagens . . . . .	78
Figura 6.13 – Média de Saltos . . . . .	79

## LISTA DE QUADROS

Quadro 3.1 – Comparação entre os Trabalhos Relacionados . . . . .	42
Quadro 5.1 – Parâmetros Utilizados no RHBS . . . . .	67

## LISTA DE SIGLAS

**5G** *Fifth Generation*

**AODV** *Ad-hoc On-Demand Distance Vector*

**BN** *Boundary Node*

**CCH** *Control Channel*

**CH** *Cluster Head*

**CH2** *Cluster Head de Segundo Nivel*

**CBLR** *Cluster Based Location Routing*

**CBR** *Cluster Based Routing*

**C-V2X** *Cellular vehicle-to-everything*

**CBDRP** *Cluster-Based Directional Routing Protocol*

**DSRC** *Dedicated Short Range Communications*

**DSR** *Dynamic Source Routing*

**DBA-MAC** *Dynamic Backbone-Assisted MAC*

**DTN** *Delay Tolerant Network*

**EDCA** *Enhanced Distributed Channel Access*

**GPS** *Global Position System*

**GPSR** *Greedy Perimeter Stateless Routing*

**GRUBi** *Grupo de Redes Ubíquas*

**HCB** *Hierarchical Cluster Based*

**HSPA** *High Speed Packet Access*

**LLC** *Logical Link Control*

**LREP** *Location Reply*

**LTE** *Long Term Evolution*

**LORA-CBF** *Location Routing Algorithm with Cluster - Based Flooding*

**LREQ** *Location Request*

**MANET** *Mobile Ad hoc Network*

**non-DTN** *non-delay tolerant network*

**OBU** *On-Board Unit*

**OFDM** *Orthogonal Frequency-Division Multiplexing*

**OMNeT++** *Objective Modular Network Testbed in C++*

**QoS** *Quality of Service*

**RSU** *Road Side Unit*

**RSUs** *Road Side Units*

**RREQ** *route request*

**RMAC** *Robust Mobility-Aware Clustering*

**RREP** *Route Reply*

**REER** *Route Error*

**RHBS** *Roteamento Hierárquico Baseado em Setores*

**SCH** *Service Channel*

**SUMO** *Simulation of Urban Mobility*

**SBHR** *Sector-Based Hierarchical Routing*

**TIBCRPH** *Traffic Infrastructure Based Cluster Routing Protocol with Handoff*

**TTL** *Time to Live*

**UTC** *Coordinated Universal Time*

**WAVE** *Wireless Access in the Vehicular Environment*

**WSMP** *WAVE ShortMessage Protocol*

**WiMAX** *Worldwide Interoperability for Microwave Access*

**VANET** *Vehicular Ad hoc Network*

**V2V** *Vehicle-to-Vehicle*

**V2I** *Vehicle-to-Infrastructure*

**V2X** *Vehicle-to-X*

**V2P** *Vehicle-to-pedestrian*

**V2N** *Vehicle-to-network*

**VRU** *Vulnerable Road Users*

**Veins** *Vehicular in Network Simulation*

**ZOR** *Zone of Relevance*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
1.1	Motivação	18
1.2	Definição do Problema	19
1.3	Solução Proposta	19
1.4	Objetivo Geral	19
1.5	Objetivos Específicos	19
1.6	Estrutura do Trabalho	20
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>21</b>
2.1	Características das VANETs	21
2.2	Aplicações	22
2.3	Tipos de Comunicação e Dispositivos Utilizados	24
2.4	Wireless Access in the Vehicular Environment (WAVE)	25
2.4.1	A Família de Padrões IEEE 1609	25
2.4.2	O Padrão IEEE 802.11p	27
2.5	Clustering de Veículos	28
2.6	Classificação dos algoritmos de roteamento em VANET	30
2.6.1	Protocolos de Roteamento Baseados em Topologia (Ad hoc)	31
2.6.2	Protocolos de Roteamento Baseados em Posição	32
2.6.3	Protocolos de Roteamento Baseados em Cluster	33
2.6.4	Protocolos de Roteamento Baseados em <i>Geocast</i>	33
2.6.5	Protocolos de Roteamento Baseados em <i>Multicast</i>	33
2.6.6	Protocolos de Roteamento Baseados em <i>Broadcast</i>	34
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>35</b>
3.1	CBLR/LORA-CBF	35
3.2	Hierarchical Cluster Based (HCB)	36
3.3	Cluster Based Routing (CBR)	37
3.4	Traffic Infrastructure Based Cluster Routing Protocol with Handoff (TIB-CRPH)	38
3.5	Cluster-Based Directional Routing Protocol (CBDRP)	38
3.6	Algoritmo Baseado em Distância Euclidiana	39
3.7	Comparação entre os Protocolos e a Solução Proposta	40

<b>4</b>	<b>ROTEAMENTO HIERÁRQUICO BASEADO EM SETORES (RHBS)</b>	<b>43</b>
4.1	Visão Geral	43
4.1.1	Tipos de Nós e Definição de Setor	43
4.1.2	Localização e Eleição de CHs	44
4.1.3	Troca de Mensagens de Controle e Envio da Mensagem de Dados	44
4.1.4	Tabelas Utilizadas pelo Protocolo	46
4.2	Etapas do Algoritmo Proposto	46
4.2.1	Descoberta de CH	46
4.2.2	Anúncio de Posição de Carros Estacionados	47
4.2.3	Descoberta de um Nó Vizinho Estacionado por um Veículo em Movimento	49
4.2.4	Envio de Posição de Veículos em Movimento para um CH	50
4.2.5	Descoberta da Posição de um Nó de Destino	51
4.2.6	Recebimento de uma Mensagem de Dados	57
4.2.7	Envio de Posição de Veículos em Movimento para o CH2	59
4.3	Roteamento Geográfico Detalhado	60
4.3.1	Modos do Roteamento Geográfico	60
4.3.2	Situações de Ocorrência de Máximo Local	61
<b>5</b>	<b>METODOLOGIA</b>	<b>64</b>
5.1	Simuladores Utilizados	64
5.1.1	OMNeT++	64
5.1.2	SUMO	64
5.1.3	Veins	65
5.1.4	GrubiX	65
5.2	Descrição do Cenário	65
5.3	Configurações Utilizadas	67
5.3.1	RHBS	67
5.3.2	Geral	67
<b>6</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>69</b>
6.1	AODV X RHBS	69
6.2	RHBS	72
6.2.1	Etapa de Requisição de Posição	72
6.2.2	Etapa de Entrega da Mensagem de Dados	74

<b>6.2.3</b>	<b>Avaliação Geral</b>	<b>76</b>
<b>6.3</b>	<b>Análise Final</b>	<b>79</b>
<b>7</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	<b>80</b>
	<b>REFERÊNCIAS</b>	<b>81</b>

## 1 INTRODUÇÃO

A evolução dos dispositivos de comunicação sem fio possibilitou a troca de informação de forma *ad hoc* entre nós móveis em uma rede na qual a infraestrutura é inexistente ou limitada. Entre as áreas de pesquisa voltadas para redes *ad hoc* móveis ou *Mobile Ad hoc Network* (MANET), a VANET tem ganhado destaque ao possibilitar aplicações que compartilham informações para prever e prevenir acidentes de trânsito, além de possibilitar aplicações que visam melhorar a eficiência do tráfego, o conforto de passageiros e a disseminação de anúncios comerciais (EZE; ZHANG; LIU, 2014). A comunicação em redes VANET pode ser classificada em *Vehicle-to-Vehicle* (V2V) na qual dois veículos comunicam diretamente entre si, *Vehicle-to-Infrastructure* (V2I) na qual os veículos comunicam com *Road Side Units* (RSUs) que fazem parte da infraestrutura das vias e em *Vehicle-to-X* (V2X) na qual é possível uma comunicação híbrida que mistura os tipos V2V e V2I (ALVES et al., 2009). Embora diferentes tipos de aplicações tenham surgido usando os diferentes tipos de comunicação, o roteamento na VANET ainda é um grande desafio por causa da alta mobilidade e topologia dinâmica dos veículos que causa mudanças constantes na rede (FONSECA; FESTAG, 2006).

Para propiciar um roteamento mais eficaz alguns algoritmos foram desenvolvidos pensando nas limitações e características exclusivas das VANETs ou adaptados para essas redes. Entretanto, protocolos de roteamento como o AODV (Perkins; Royer, 1999) projetados para redes MANET também são utilizados em redes veiculares ao demonstrarem eficiência em determinados cenários de comunicação. Para tentar organizar os protocolos desenvolvidos de acordo com a sua forma de funcionamento, classificações foram criadas. Entre as classes, os protocolos baseados em posição e baseados em cluster têm ganhado destaque por possibilitarem baixa latência no roteamento e baixo *overhead* de mensagens de controle embora também possuam suas limitações.

No roteamento baseado em posição ou roteamento geográfico como também é chamado, não é necessário armazenar ou descobrir a topologia de rede como ocorre em protocolos de roteamento proativos e reativos. Para o seu funcionamento, os nós utilizam dispositivos de localização como *Global Position System* (GPS) para determinar sua posição que é posteriormente compartilhada com os nós vizinhos de um salto de distância por meio de *beacons*. Já a posição do veículo de destino a ser inserida no cabeçalho do pacote a ser enviado, pode ser fornecida por um serviço de localização (Cadger et al., 2013) e (BOUSSOUFA-LAHLAH; SEMCHEDINE; BOUALLOUCHE-MEDJKOUNE, 2018).

Após a troca de mensagens de posição entre nós, duas estratégias de roteamento podem ser utilizadas. Na primeira chamada de roteamento guloso, um pacote é sempre encaminhado para o nó a um salto de distância que esteja mais próximo do destino. Porém, esta estratégia sofre com o problema de máximo local que consiste em o nó de encaminhamento não ser capaz de encontrar um nó vizinho mais próximo do destino do que ele mesmo. Este problema ocorre por causa da existência de espaços “vazios” ou “buracos” entre nós. Para resolvê-lo, uma das soluções empregadas é eliminar o pacote para evitar loops nos quais ele é enviado para trás. Na segunda estratégia chamada de roteamento de face, as faces de um grafo planar são atravessadas usando uma técnica denominada como regra da mão direita (*right hand rule*) até que o destino seja atingido. Ao serem comparadas as duas técnicas, o roteamento guloso é mais rápido e mais simples de implementar, já o roteamento de face é mais lento, mais difícil de implementar, porém sempre entrega o pacote ao destino (Cadger et al., 2013). Para tentar sanar as deficiências dessas duas estratégias alguns algoritmos híbridos de roteamento geográfico como *Greedy Perimeter Stateless Routing* (GPSR) (KARP; KUNG, 2000) as combinaram. Nos algoritmos geográficos híbridos, o roteamento inicia-se usando a estratégia gulosa, ao ser identificado um máximo local, muda-se para a estratégia do roteamento de face e posteriormente retorna-se novamente para a estratégia de roteamento guloso ao ser encontrado um nó mais próximo do destino. Deste modo a entrega do dado é garantida e o tempo gasto para tal ação é o menor possível.

Nos protocolos baseados em cluster, grupos virtuais são criados entre os veículos. Em cada cluster um nó é eleito como cluster head. Este nó será responsável por gerenciar a comunicação em cada grupo (intra-cluster) e entre clusters (inter-cluster) (Bengag; El Boukhari, 2018). Porém em redes VANET, o tempo de vida de um cluster pode ser pequeno também por causa da alta velocidade dos veículos e mudanças de direção. Pensando nisso, em Rawashdeh e Mahmud (2012) é apresentada uma técnica de clustering de veículos para o ambiente de rodovia que além de utilizar localização e direção de veículos, considera também a diferença de velocidades deles para formação de clusters. Nesta técnica veículos rápidos são agrupados em um cluster e veículos lentos em outro cluster. Embora essa técnica tenha se mostrado interessante para manter a estabilidade e um tempo maior de vida para clusters, o ambiente urbano possui suas próprias características como menor velocidade e mudanças de direção constantes.

Diante disso e ao tentar superar os protocolos de roteamento reativos, neste trabalho é proposto o protocolo de roteamento RHBS. Neste protocolo, cada cluster é um setor delimitado

por um conjunto de ruas que possuem estacionamentos. Os nós estacionados nos estacionamentos dessas ruas formam backbones que interligam os setores. O backbone de carros estacionados são utilizados para rotear mensagens de controle e de dados por meio do roteamento geográfico guloso. Cada setor possui um CH de primeiro nível que armazena a posição dos veículos móveis de seu setor, além de existir um CH2 próximo ao centro da área geográfica na qual se localiza os setores. O CH2 armazena a posição dos veículos em movimento de todos os setores.

No RHBS, veículos móveis enviam periodicamente mensagens *hello* com atualização de suas posições para o CH do seu setor. Posteriormente as posições armazenadas pelo CH são enviadas para o CH2. Neste protocolo quando um nó em movimento deseja enviar uma mensagem de dados para um nó de destino também em movimento, deve primeiro enviar uma mensagem de requisição de posição para o CH do seu setor. Caso a posição seja desconhecida a requisição é reencaminhada para o CH2. Em ambos os casos, ao saberem a posição do nó de destino, o CH ou CH2 respondem para o nó de origem que então envia a mensagem de dados em direção da posição recebida.

## 1.1 Motivação

Protocolos de roteamento reativos (sob demanda), tem se apresentado como uma melhor opção que protocolos baseados em *broadcast* e proativos (orientados por tabela) por possibilitarem uma latência menor e um menor *overhead* de mensagens de controle para cada troca de mensagem de dados. Porém, os protocolos reativos tem mostrado um desempenho inferior quando comparados com protocolos baseados em cluster ou baseados em posição. Por sua vez protocolos baseados em cluster podem ser afetados pelo tempo de vida dos clusters e o *re-clustering* constante também pode gerar um *overhead* de mensagens significativo na rede. O roteamento geográfico guloso por sua vez sofre com o problema de local máximo no qual um método de recuperação deve ser usado. Entre os métodos utilizados o roteamento de face é o mais empregado, porém é mais lento que o roteamento guloso. Portanto, o desenvolvimento de protocolos de roteamento para VANET ainda é algo que tem se mostrado promissor.

## 1.2 Definição do Problema

Embora vários tipos de protocolos de roteamento tenham surgido, ainda é um grande desafio o desenvolvimento de um protocolo eficiente que possa ser escalável em um ambiente urbano e que tenha uma baixa latência aliada a um menor *overhead* de mensagens de controle.

## 1.3 Solução Proposta

O trabalho apresenta o protocolo RHBS para ambientes urbanos a fim de se possibilitar uma menor latência e um menor *overhead* de mensagens de controle. Este protocolo funde roteamento em cluster com roteamento geográfico guloso. Nós estacionados denominados como CHs presentes em cada setor e um nó estacionado denominado como CH2 próximo ao centro da área geográfica onde se encontram os setores, são consultados para fornecer a posição de um nó móvel. Este protocolo possui dois modos de roteamento geográfico guloso que utilizam o backbone de carros estacionados para o encaminhamento de mensagens. O primeiro modo sempre encaminha o pacote para o nó vizinho mais próximo do destino. Quando um máximo local é encontrado ele utiliza o segundo tipo de roteamento guloso para encaminhar o pacote sempre para o nó que esteja mais próximo de uma interseção. Quando o pacote chega no último nó estacionado da interseção, o roteamento retorna para o primeiro tipo de roteamento guloso e envia o pacote para um nó de uma rua vizinha que esteja mais próximo do destino.

## 1.4 Objetivo Geral

Desenvolver um protocolo de roteamento eficiente para o ambiente urbano.

## 1.5 Objetivos Específicos

Os objetivos específicos para o desenvolvimento do protocolo, consistem em:

1. Utilizar somente a comunicação V2V ao se pensar em custos futuros de implementação;
2. Criar e utilizar um mapa de ruas, que de acordo com a via trafegada informe o setor/-cluster, além do Id e posição do CH deste setor;
3. Criar um backbone de carros estacionados que será utilizado no roteamento geográfico guloso para a troca de mensagens de controle e dados;

4. Veículos em movimento devem enviar periodicamente mensagens *hello* com a atualização de sua posição para o CH do seu setor;
5. CHs ao possuírem atualizações de posição de veículos móveis devem enviar suas tabelas atualizadas para o CH2;
6. Quando um veículo móvel de origem deseja enviar uma mensagem para um veículo móvel de destino, ele deve requisitar a posição deste nó para seu CH;
7. Quando o CH do setor não conhece a posição requisitada de um veículo de destino, a requisição deve ser encaminhada para o CH2;
8. O veículo móvel de origem ao obter a posição do veículo móvel de destino do seu CH ou CH2, deve em seguida enviar a mensagem de dados;
9. Definir um modo de recuperação para o roteamento que encaminhe a mensagem para um nó mais próximo de uma interseção quando um local máximo for encontrado;
10. Comparar o protocolo desenvolvido com o protocolo baseado em topologia e reativo AODV.

## 1.6 Estrutura do Trabalho

O presente trabalho está organizado da seguinte forma: O Capítulo 2 apresenta o referencial teórico que contém os conceitos utilizados para o desenvolvimento do trabalho. O Capítulo 3 apresenta os trabalhos relacionados sobre protocolos de roteamento em cluster. O Capítulo 4 apresenta o protocolo proposto RHBS. O Capítulo 5 apresenta a metodologia utilizada para o desenvolvimento do protocolo. O Capítulo 6 apresenta os resultados obtidos. Por fim, o Capítulo 7, apresenta a conclusão e os trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

A VANET é uma subclasse da MANET onde os veículos são os nós (EZE; ZHANG; LIU, 2014). Segundo Alves et al. (2009) as VANETs, são redes formadas entre veículos automotores (automóveis pessoais, caminhões, ônibus etc.) ou entre os veículos e a infraestrutura fixa existente nas vias de transporte, que trocam dados ao utilizarem interfaces de comunicação sem fio. Para Wangham et al. (2014) a ideia básica das VANETs é adaptar tecnologias existentes como redes sem fio e das redes de sensores e instalá-las em veículos que serão capazes de coletar, gerar, analisar e compartilhar dados.

A seguir neste capítulo, são definidas as características das redes VANET. Posteriormente são descritas as principais aplicações utilizadas, os tipos de comunicação e dispositivos utilizados, além da arquitetura de comunicação WAVE. Por fim, o *clustering* de veículos e as classes de algoritmos de roteamento são apresentados.

### 2.1 Características das VANETs

As VANETs possuem características como a auto-organização dos nós e a falta de um controle central, assim como as redes ad hoc convencionais e de sensores (FONSECA; FESTAG, 2006). Porém, as Redes Ad hoc Veiculares possuem características exclusivas que as definem e que as diferenciam de outros tipos de redes sem fio. Entre elas:

1. **Alta Mobilidade e Topologia Dinâmica:** Nós podem moverem-se com alta velocidade ao longo de ruas e rodovias. Durante a comunicação entre veículos alguns nós são utilizados como nós intermediários de encaminhamento em uma cadeia de múltiplos saltos. Porém quando este nós de encaminhamento se movem rapidamente, as rotas estabelecidas entre dois nós que se comunicam são interrompidas (FONSECA; FESTAG, 2006);
2. **Não Possui Restrições de Energia:** O veículo possui bateria que pode fornecer energia de forma contínua para a *On-Board Unit* (OBU) (YOUSEFI; MOUSAVI; FATHY, 2006) e (AL-SULTAN et al., 2014);
3. **Densidade Variável de Rede:** Devido à densidade veicular, as redes veiculares podem ser densas ou esparsas, o que influencia na adoção de diferentes abordagens de roteamento (YOUSEFI; MOUSAVI; FATHY, 2006) e (AL-SULTAN et al., 2014);

4. **Rede de Grande Escala:** A rede pode ser maior em áreas centrais, rodovias e também em entradas e saídas de cidades (YOUSEFI; MOUSAVI; FATHY, 2006) e (AL-SULTAN et al., 2014);
5. **Grande Capacidade Computacional:** Os veículos podem ser equipados com sensores e ter melhores recursos de processamento e memória. A detecção da posição atual e do movimento dos nós utilizando sensores como o GPS, auxilia em decisões de comunicação e roteamento (FONSECA; FESTAG, 2006) e (AL-SULTAN et al., 2014);
6. **Mobilidade Previsível:** Diferentemente de outros tipos de rede móveis, onde os nós se movem de forma aleatória, na VANET o trajeto realizado pelos veículos é previsível devido a existência das vias de circulação já estabelecidas e dos sinais de trânsito (AL-SULTAN et al., 2014);
7. **Confidencialidade Inexistente para Informações de Segurança:** Em aplicações de segurança de trânsito, as informações existentes em uma mensagem, são de interesse de todos os usuários de uma via e portanto não devem ser confidenciais (FONSECA; FESTAG, 2006).

## 2.2 Aplicações

No que se refere às aplicações, Karagiannis et al. (2011), as classifica em três tipos, aplicações ativas de segurança rodoviária (*Active road safety applications*), aplicações de eficiência e gerenciamento de tráfego e aplicações de informação e entretenimento (*Infotainment applications*). Classificação parecida foi realizada por Hossain et al. (2010) e Papadimitratos et al. (2009). Outra nomenclatura classificatória utilizada foi feita por Eze, Zhang e Liu (2014) e Liang et al. (2015), que definem as aplicações como relacionadas à segurança e não relacionadas à segurança, que por sua vez também obteve classificação parecida no trabalho de Wischhof, Ebner e Rohling (2005). De forma geral, pode-se dizer que as aplicações classificadas são as mesmas. Porém, o modelo classificatório de Eze, Zhang e Liu (2014) e Liang et al. (2015) une os dois últimos tipos de aplicações propostas por Karagiannis et al. (2011), em uma só classe. Eles uniram as aplicações de eficiência e gerenciamento de tráfego e as aplicações de informação e entretenimento na classe de aplicações não relacionadas à segurança.

As aplicações ativas de segurança rodoviária ou as aplicações relacionadas à segurança de trânsito, têm o objetivo de preservar vidas ao diminuir a probabilidade de acidentes de trân-

sito por meio de colisões entre veículos. As colisões podem ser evitadas através do compartilhamento de informações como posição atual do veículo, posição de intersecção, velocidade, aceleração e direção de movimento com outros veículos e RSUs, que as processaram para prevenir acidentes de trânsito e colisões. Porém, para um melhor êxito, as informações devem ser entregues de forma confiável e em um curto período de tempo (KARAGIANNIS et al., 2011) e (EZE; ZHANG; LIU, 2014).

Alguns exemplos de aplicações relacionadas a segurança são: aviso de possibilidade de colisão de intersecção, aviso de ultrapassagem de veículo, aviso de possibilidade de colisão na extremidade traseira, aviso de veículo estacionário, notificação de local perigoso, aviso de risco de colisão, aviso de violação de sinal, aviso de condução de veículo em direção errada (contramão), aviso de frenagem brusca (*Emergency electronic brake lights*), assistência para mudança de faixa, aviso de perda de controle de veículo, aviso de veículo de emergência, aviso em cooperação da possibilidade de colisão frontal, detecção e aviso de pré-colisão com ativação de equipamentos do veículo para diminuir os efeitos de uma batida (KARAGIANNIS et al., 2011).

As aplicações não relacionadas à segurança, visam melhorar a eficiência do tráfego, o conforto de passageiros e a disseminação de anúncios comerciais. Alguns exemplos destas aplicações são a que tem como finalidade informações sobre condições de trânsito, estacionamentos mais próximos, informações meteorológicas, localização e informações sobre postos de combustíveis, restaurantes, shoppings e hotéis, além do fornecimento de acesso à internet (EZE; ZHANG; LIU, 2014) e (WISCHHOF; EBNER; ROHLING, 2005). No que diz respeito a divulgação de anúncios, gateways estacionários podem ser instalados nas vias para transmitir informações sobre preços, produtos e serviços das instituições antes destacadas, para os veículos da VANET (EZE; ZHANG; LIU, 2014).

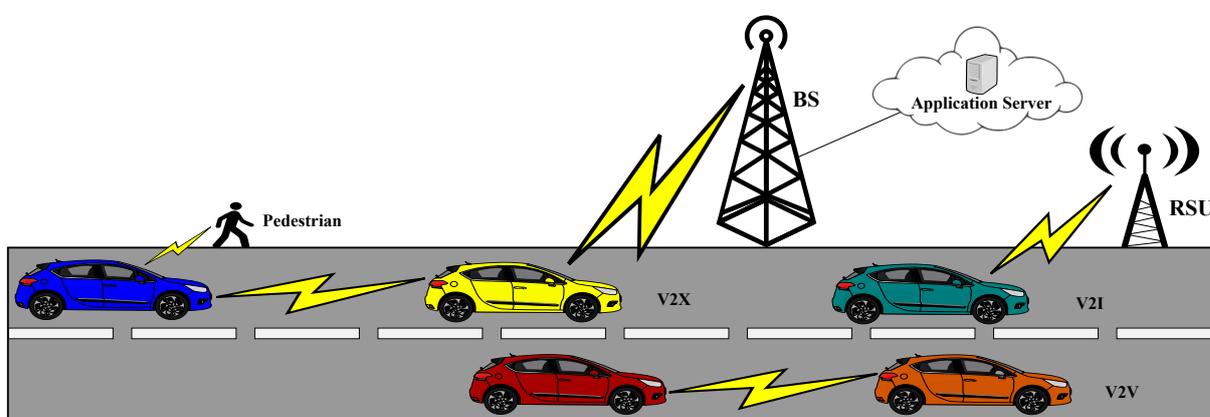
Embora os serviços não relacionados à segurança tragam praticidade aos motoristas da VANET, estes não podem interferir ou sobressair sobre os serviços relacionados à segurança. Diante disso, Eze, Zhang e Liu (2014) sugerem que em redes veiculares onde aplicações relacionadas à segurança e não relacionadas à segurança devam coexistir, deve-se utilizar canais diferentes para cada tipo de aplicação ou deve ser aplicado priorização de tráfego no qual mensagens relacionadas à segurança possuam mais prioridade do que mensagens não relacionadas a segurança.

### 2.3 Tipos de Comunicação e Dispositivos Utilizados

As redes veiculares possuem três tipos de comunicação (FIGURA 2.1) que definem a forma como os nós se organizam e se comunicam (ALVES et al., 2008) e (ALVES et al., 2009). No primeiro tipo de comunicação que é puramente *ad hoc*, V2V, devido à ausência de infraestrutura, os veículos comunicam diretamente entre si e por múltiplos saltos entre nós. No segundo tipo que envolve infraestrutura, V2I, os veículos comunicam e trocam informações com nós estáticos conhecidos como *Road Side Unit* (RSU)s, que também são nomeados como Unidades de Acostamento (ALVES et al., 2009).

No terceiro e último tipo de comunicação que é híbrida, Vegni e Little (2011) ao prestarem sua contribuição, chamam esse modelo de V2X e explicam que nele existe uma fusão entre os dois modelos anteriores, o que possibilita uma alternância entre comunicação *ad hoc* e comunicação que utiliza infraestrutura. Uma visão mais atualizada do termo V2X, de acordo com Five G Americas (2016) e Five G Americas (2018), denomina seu significado como "*Vehicle to Everything*", em português, "Veículo para Tudo". De acordo com estes dois documentos, o V2X além de possibilitar comunicações V2V e V2I permite também comunicações do tipo *Vehicle-to-pedestrian* (V2P) que consiste da comunicação entre veículos e usuários vulneráveis da estrada, *Vulnerable Road Users* (VRU), como pedestres ou ciclistas e *Vehicle-to-network* (V2N) que consiste da comunicação entre um veículo e um servidor de aplicações. Ao utilizar rede celular *Long Term Evolution* (LTE) e *Fifth Generation* (5G), o V2X passa a ser chamado também de *Cellular vehicle-to-everything* (C-V2X), para diferenciá-lo do modelo inicial que empregou 802.11p na comunicação.

Figura 2.1 – Tipos de Comunicação



Fonte: autor (2020)

No que se refere aos dispositivos utilizados para a comunicação na VANET ao utilizar o padrão *Wireless Access in the Vehicular Environment (WAVE)*, descrito no tópico a seguir, Zeadally et al. (2012) introduz que existem dois tipos. O primeiro é a RSU que, como visto anteriormente, é um dispositivo estacionário e o segundo é a OBU que é um dispositivo móvel. Ainda, de acordo com este autor, ambos os dispositivos podem operar conforme a arquitetura cliente-servidor, alternando entre esses modos.

Ao prestarem sua contribuição, Wangham et al. (2014), dizem que a OBU ou Unidade de Bordo como também é chamada, é um componente embarcado presente nos veículos que realiza funções como roteamento entre os nós da rede, controle de congestionamento e a transferência de mensagens confiáveis. Já uma RSU é um componente de infraestrutura que pode ser instalado em postes de luz, semáforos e nas margens das vias de trânsito.

De acordo Al-Sultan et al. (2014), a RSU tem a finalidade de estender o alcance da rede veicular, ao redistribuir informações para OBUs e ao enviar informações para outras RSUs que também as encaminham para outras OBUs. Além disso, Al-Sultan et al. (2014) e Mejri, Ben-Othman e Hamdi (2014), dizem que por meio de uma RSU, uma OBU pode se conectar a redes externas, como a internet.

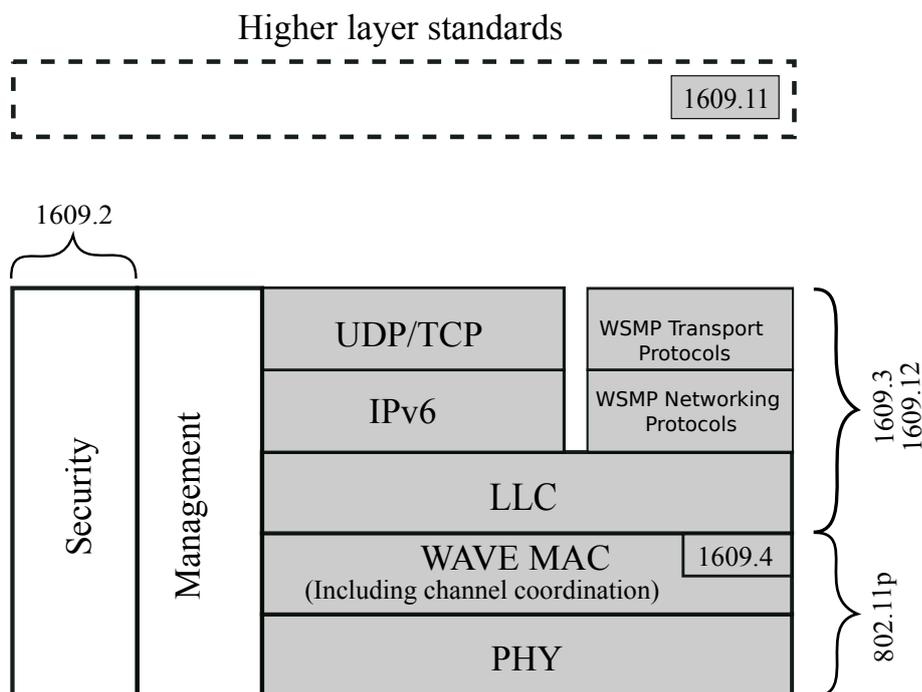
## **2.4 Wireless Access in the Vehicular Environment (WAVE)**

O WAVE consiste em uma arquitetura para comunicação em redes veiculares composta por padrões da família IEEE 1609 e do padrão IEEE 802.11p (ALVES et al., 2009), (WANGHAM et al., 2014) e (UZCÁTEGUI; ACOSTA-MARUM, 2009). Ele também pode ser considerado como um sistema de comunicação de rádio destinado a fornecer serviços interoperáveis e ininterruptos aos usuários de um sistema de transporte IEEE... (2019).

### **2.4.1 A Família de Padrões IEEE 1609**

A família de padrões IEEE 1609 é composta por documentos que descrevem dispositivos, protocolos, canais, endereços, identificadores, além de recursos de segurança que podem ser utilizados na troca de mensagens. A Figura 2.2 de acordo com o padrão IEEE 1609.0 de 2019 (*IEEE Guide for Wireless Access in Vehicular Environments (WAVE) Architecture*), demonstra a organização das camadas da arquitetura WAVE, bem como a suas referências aos documentos da família de padrões IEEE 1609 e padrão IEEE 802.11p.

Figura 2.2 – Arquitetura WAVE



Fonte: adaptado de (IEEE... , 2019)

De acordo com Alves et al. (2009), Wangham et al. (2014) e IEEE... (2019), de forma resumida, os principais documentos da família de padrões IEEE 1609 são:

- **IEEE 1609.0:** Fornece uma visão geral do sistema WAVE, seus componentes e sua operação, além de estabelecer um contexto para que se possa entender melhor os padrões da família IEEE 1609 e o padrão IEEE 802.11p.
- **IEEE 1609.2 - Serviços de Segurança:** Ao objetivar prover segurança para mensagens de aplicações e de gerenciamento, este padrão define o formato para mensagens seguras, quando elas devem ser utilizadas e como elas devem ser processadas. Este padrão define também mecanismos de criptografia ao utilizar chaves criptográficas e certificados.
- **IEEE 1609.3 – Serviços de Rede:** Especifica os serviços da camada de controle de enlace lógico, ou seja, *Logical Link Control* (LLC) e para duas camadas semelhantes as camadas de rede e de transporte do modelo OSI/ISO. Para essas duas últimas camadas os serviços oferecidos são roteamento e endereçamento com suporte a troca de mensagens seguras. Este padrão especifica para o WAVE duas pilhas de protocolos, o IPv6 e o WAVE *ShortMessage Protocol* (WSMP), assim pode suportar comunicações de alta prioridade e sensíveis ao tempo e também aplicações menos exigentes que utilizam TCP/UDP em suas transmissões.

- **IEEE 1609.4 – Operações Multicanais:** Na arquitetura WAVE este padrão descreve as operações em múltiplos canais que utilizam o protocolo 802.11p. Ele descreve a utilização de dois tipos de canais, um canal de controle que é o *Control Channel* (CCH) e múltiplos canais de serviço denominados como *Service Channel* (SCH).
- **IEEE 1609.11 – Pagamento Eletrônico:** É um padrão em nível de aplicação que especifica um protocolo de pagamento eletrônico, ao definir os serviços e o formato de mensagens para este fim.
- **IEEE 1609.12 – Identificadores:** Define a alocação de alguns identificadores usados pelo padrão WAVE.

#### 2.4.2 O Padrão IEEE 802.11p

O IEEE 802.11p especifica na arquitetura WAVE, as camadas Física (PHY) inspirada no padrão 802.11a e de Controle de Acesso ao Meio (MAC) inspirada no padrão 802.11e.

A camada PHY é baseada no *Orthogonal Frequency-Division Multiplexing* (OFDM) que possui taxas de fluxo que variam entre 3 à 27 Mbps ao utilizar canais com largura de 10 MHz (MEJRI; BEN-OTHTMAN; HAMDI, 2014). Além disso, adota a faixa de operação de 5,9 GHz do *Dedicated Short Range Communications* (DSRC), esses fatores possibilitam uma melhor comunicação em ambientes veiculares conhecidos por sua alta mobilidade e constantes mudanças de canal (RASOOL; ZIKRIA; KIM, 2017).

A camada MAC usa o protocolo *Enhanced Distributed Channel Access* (EDCA). Para transmitir mensagens de segurança dentro de um prazo razoável, o EDCA introduz o conceito de gerenciamento com qualidade de serviço, ou seja, *Quality of Service* (QoS), ao definir quatro categorias de acesso de acordo com o tipo de tráfego que são: tráfego de segundo plano, tráfego de melhor esforço, tráfego de vídeo e tráfego de voz (MEJRI; BEN-OTHTMAN; HAMDI, 2014) e (WANGHAM et al., 2014).

A atualização da norma IEEE 802.11 em 2012, incorporou a emenda p publicada anteriormente em 2010, porém devido a comunidade de pesquisa ainda utilizar o termo IEEE 802.11p para se referir exclusivamente ao padrão utilizado em Redes Veiculares Ad hoc, neste trabalho, o termo também será utilizado no decorrer do documento quanto este for necessário.

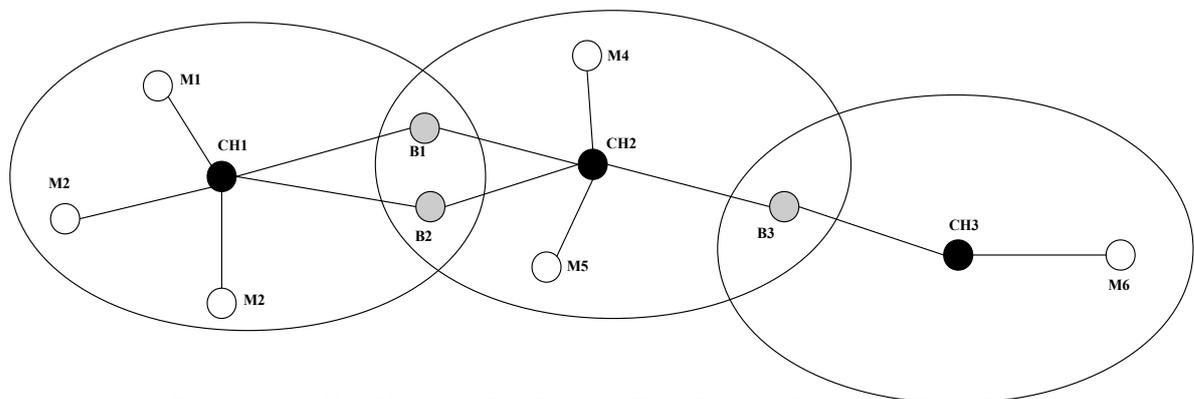
## 2.5 Clustering de Veículos

Para minimizar os efeitos nocivos do *flooding* (AlMheiri; AlQamzi, 2015) e resolver problemas como tempo de vida de rede reduzido e aumento de *overhead* de roteamento, ocasionados pela mudança frequente de topologia devido a alta mobilidade dos veículos (HANDE; MUDDANA, 2016), a adoção de algoritmos de *clustering* tem sido uma solução comum adotada em redes VANET.

Em uma rede Ad hoc, o *clustering* consiste em um processo de agrupamento no qual grupos virtuais são criados a partir dos nós (VODOPIVEC; BEŠTER; KOS, 2012), (BALI; KUMAR; RODRIGUES, 2014) e (SUCASAS et al., 2016).

Em uma rede veicular em cluster, os veículos são os nós que podem receber diferentes funções ou status, como *Cluster Head* (CH), gateway e membro (BALI; KUMAR; RODRIGUES, 2014), (HANDE; MUDDANA, 2016) e (COOPER et al., 2017). Em cada cluster existe pelo menos um CH. Ele é o nó responsável por coordenar os membros do seu cluster e por mediar entre o cluster e o restante da rede da mesma maneira que um ponto de acesso sem fio utilizado em uma rede de infraestrutura. O nó gateway, chamado também de bridge (SANTOS; EDWARDS; SEED, 2002), é utilizado para encaminhar informações entre clusters ao conectar dois CHs. Já o nó membro ao ser um nó comum não possui links entre clusters e não possui funções de CH (BALI; KUMAR; RODRIGUES, 2014), (HANDE; MUDDANA, 2016) e (COOPER et al., 2017). Devido às leis de propagação de sinal de rádio, o cluster descrito como ideal é representado como um círculo com um CH no centro e demais nós em torno dele (VODOPIVEC; BEŠTER; KOS, 2012). A Figura 2.3, tendo como exemplo o algoritmo de cluster CBLR (SANTOS; EDWARDS; SEED, 2002), ilustra os tipos de nós e suas relações.

Figura 2.3 – Algoritmo de Cluster CBLR



**Description: M = Member, B = Bridge, CH = Cluster Head**

Fonte: adaptado de (SANTOS; EDWARDS; SEED, 2002)

De acordo com Cooper et al. (2017), os algoritmos de roteamento em cluster tendem a formar estruturas hierárquicas de três formas diferentes. A primeira delas, como já visto consiste em uma hierarquia que é constituída a partir de nós gateway que são membros de mais de um cluster. Na segunda maneira um cluster head é membro de outro cluster, tendo-se uma estrutura semelhante a uma árvore. No terceiro e último tipo, uma estrutura de backbone dinâmica é formada, na qual os CHs trocam mensagens de roteamento entre si. De forma respectiva, o autor cita os algoritmos CBLR (SANTOS; EDWARDS; SEED, 2002), *Robust Mobility-Aware Clustering* (RMAC) (Goonewardene; Ali; Stipidis, 2009) e *Dynamic Backbone-Assisted MAC* (DBA-MAC) (Bononi; di Felice, 2007) como representantes de cada modelo hierárquico. Diante destes modelos, como pode se observar, o nó gateway nem sempre é utilizado em uma estrutura hierárquica de roteamento em cluster.

No que se refere ao processo de formação de um cluster, os nós de forma geral, transmitem informações de controle, como identidade do nó, lista de vizinhos, mobilidade, etc. Após esta troca de informações, os CHs de acordo com uma métrica predefinida são selecionados. Posteriormente o restante dos nós se unem aos CHs, sendo que os membros do cluster que interconectam clusters se tornam gateways. Porém estas mensagens de controle, ou de sinalização como também são chamadas, não são somente usadas para a seleção de CHs ou formação de cluster, elas são usadas também para a manutenção do cluster e para tarefas de roteamento (SUCASAS et al., 2016).

No que diz respeito ao tamanho do cluster, ele é influenciado principalmente pelo alcance de transmissão do dispositivo de comunicação sem fio que é usado pelos nós próximos para a formação dos grupos virtuais. Porém, alguns algoritmos de cluster também implementam filtros que impedem que os nós ingressem em um cluster. Um exemplo de filtro é o filtro de direção de movimento, no qual um nó não se junta a um cluster onde o CH se move na direção oposta (VODOPIVEC; BEŠTER; KOS, 2012).

Entre os benefícios da utilização do *clustering*, o primeiro é a criação de um *backbone* formado por cluster heads e gateways que é utilizado para rotear pacotes. No *backbone*, a comunicação entre clusters é conduzida pelos CHs, de forma que os nós membros do cluster transmitem os seus dados primeiramente para o seu CH, o que reduz a complexidade do roteamento ao reduzir o número e o tamanho das rotas na rede (SUCASAS et al., 2016). Um segundo benefício diz respeito a uma melhor alocação de canais para evitar interferências. Em redes Ad hoc clusterizadas, os nós podem compartilhar um canal de controle comum para a formação e

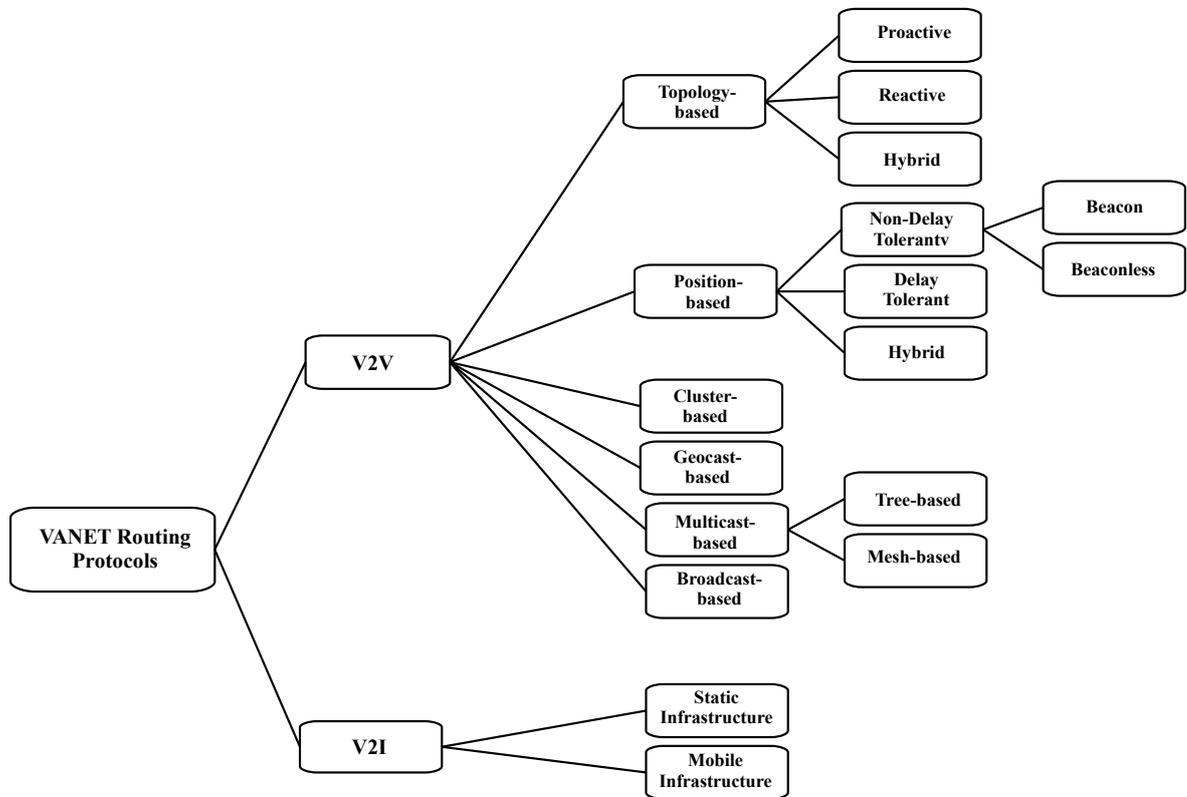
manutenção de cluster, enquanto diferentes canais podem ser alocados para a transmissão de dados (Ting-Chao Hou; Tzu-Jane Tsai, 2001) e (SUCASAS et al., 2016).

## 2.6 Classificação dos algoritmos de roteamento em VANET

Durante o desenvolvimento das redes VANET, várias pesquisas tentaram classificar os protocolos de roteamento utilizados nestas redes. Entre elas pode-se destacar os trabalhos de Li e Wang (2007), Dua, Kumar e Bawa (2014), Sharef, Alsaqour e Ismail (2014), Senouci, Zibouda e Harous (2017), Devangavi e Gupta (2017) e Bengag e El Boukhari (2018). Porém, estas classificações quando comparadas apresentam divergências, o que demonstra que uma classificação comum que consiga dividir os algoritmos de roteamento em VANET de acordo com suas características principais parece ser algo que ainda deve ser alcançado.

A taxonomia proposta por Sharef, Alsaqour e Ismail (2014) (FIGURA 2.4), embora seja mais antiga que algumas classificações de outros trabalhos citados, possui uma melhor organização o que possibilita um melhor entendimento. Ela classifica os protocolos de roteamento usados na arquitetura V2V em baseados em topologia, baseados posição, baseados em cluster, baseados em *Geocast*, baseados em *Multicast* e baseados em *Broadcast*. Além disso, classifica os protocolos usados na arquitetura V2I em protocolos de roteamento baseados em infraestrutura estática e protocolos de roteamento de infraestrutura móvel. De acordo com essa taxonomia, as classes de algoritmos de roteamento utilizados na arquitetura V2V são detalhadas a seguir.

Figura 2.4 – Taxonomia de Protocolos de Roteamento em VANET



Fonte: adaptado de (SHAREF; ALSAQOUR; ISMAIL, 2014)

### 2.6.1 Protocolos de Roteamento Baseados em Topologia (Ad hoc)

Os protocolos dessa classe mantêm tabelas de roteamento para armazenar as informações de link de rede. Com base nesses dados armazenados, encaminha pacotes do veículo de origem para o veículo de destino (SHAREF; ALSAQOUR; ISMAIL, 2014) e (Devangavi; Gupta, 2017). Esta classe é dividida nas subclasses de protocolos de roteamento proativo (orientado por tabela), protocolos de roteamento reativo (sob demanda) e protocolos híbridos.

1. **Protocolos de Roteamento Proativo (Orientado por Tabela):** Nesta classe cada nó mantém uma tabela de roteamento que possui informações sobre os nós presentes na topologia de rede. As tabelas de roteamento são atualizadas regularmente devido a mudanças na topologia e trocadas entre os nós. Desta forma, não é necessário descobrir a rota de destino, pois ela já estará armazenada na tabela (SHAREF; ALSAQOUR; ISMAIL, 2014) e (Devangavi; Gupta, 2017).

2. **Protocolos de Roteamento Reativo (Sob Demanda):** Nestes protocolos, as rotas são estabelecidas sob demanda, ou seja, uma rota é descoberta quando tem-se um dado que deve ser transmitido. Na fase de descoberta de rota é realizado um *flooding* de pacotes de solicitação de rota na rede. Quando o pacote atinge o nó de destino, um pacote de resposta contendo as informações da rota é enviado de forma reversa para o nó de origem, usando a comunicação unicast (SHAREF; ALSAQOUR; ISMAIL, 2014) e (Devangavi; Gupta, 2017).
3. **Protocolos Híbridos:** São uma combinação dos protocolos de roteamento reativo e proativo para tornar o roteamento mais escalável e eficiente.

### 2.6.2 Protocolos de Roteamento Baseados em Posição

Os protocolos desta classe, também são chamados de protocolos de roteamento geográfico. Nela os nós ao utilizarem dispositivos GPS, conhecem sua posição geográfica e por meio de solicitações de informações de posição de nós vizinhos, também passa a conhecer a posição deles. As informações de posição dos nós são utilizadas para decisões de roteamento. Esta estratégia de roteamento tem um desempenho melhor, pois nele não é necessário criar e manter uma rota global do nó de origem para o nó de destino. Esta classe é dividida ainda em três subclasses que são: protocolos de roteamento de rede não tolerante a atraso (*non-delay tolerant network (non-DTN) routing protocols*), protocolos de roteamento de rede tolerante a atraso (*Delay Tolerant Network (DTN) routing protocols*) e protocolos de roteamento híbrido.

1. **Protocolos de Roteamento de Rede não Tolerante a Atraso:** Em protocolos non-DTN, ao utilizar uma abordagem gananciosa, os nós encaminham o pacote de dados sempre para o nó vizinho mais próximo do destino. Porém, se os vizinhos de um nó que quer transmitir o pacote não forem os mais próximos do destino que o próprio nó, o pacote atingiu o “máximo local”, diante disso cada protocolo dessa categoria utiliza uma estratégia de recuperação para que este problema seja resolvido e o dado chegue ao destino. Essa subclasse ainda se divide em outras duas subclasses. Protocolos de roteamento que transmitem *beacons hello* para identificar e localizar outros nós e protocolos de roteamento sem *beacon (Beaconless)* que não usam mensagens de sinalização regulares para identificar e localizar outros nós.

2. **Protocolos de Roteamento de Rede Tolerante a Atraso:** Em protocolos DTN, devido a constante desconexão de rede entre os nós que impossibilita a existência de um caminho de fim-a-fim, ao ser utilizada a técnica de *carry-and-forward*, os veículos armazenam o pacote de dados quando perdem contato com outros veículos e o carregam por uma determinada distância até que outros nós sejam encontrados, quando isso ocorre o pacote é transmitido para os nós vizinhos.
3. **Protocolos Híbridos:** Nesta subclasse as estratégias DTN e non-DTN são mescladas. Isso ocorre porque mesmo que nas redes DTN utilizem os modos de roteamento guloso e de recuperação, em alguns momentos em que a rede esteja desconectada ou particionada em redes esparsas, estes modos de roteamento geográfico não serão eficientes. Com isso, a utilização da estratégia de roteamento DTN possibilita a solução para o problema.

### 2.6.3 Protocolos de Roteamento Baseados em Cluster

Como visto no tópico 2.5, cluster são agrupamentos virtuais de veículos. Cada cluster possui um cluster head responsável pelo gerenciamento intra-cluster e inter-cluster.

### 2.6.4 Protocolos de Roteamento Baseados em Geocast

O *Geocast* é um roteamento *multicast* baseado em posição empregado para encaminhar uma mensagem de um nó de origem para todos os nós de uma área geográfica específica, também chamada de zona de relevância, *Zone of Relevance (ZOR)* (ZEADALLY et al., 2012) e (SHAREF; ALSAQOUR; ISMAIL, 2014).

### 2.6.5 Protocolos de Roteamento Baseados em Multicast

Nas redes tradicionais com fio, o conceito de transmissão *multicast*, refere-se a transmissão de uma única fonte para um grupo de nós de destino sem nenhuma limitação geográfica. A transmissão *multicast* na VANET normalmente é uma transmissão de uma única fonte para vários destinos dentro de uma região geográfica específica, sendo geralmente realizada pelo roteamento *geocast* (Hassan; Ahmed; Rahman, 2011) e (SHAREF; ALSAQOUR; ISMAIL, 2014). Essa classe é dividida em duas subclasses.

1. **Protocolos de Roteamento *Multicast* Baseados em Árvore (*Tree-based multicast routing protocols*):** Estes protocolos matem uma árvore de roteamento *multicast* compartilhada para fornecer dados de uma origem para os receptores em um grupo *multicast*.
2. **Protocolos de Roteamento *Multicast* Baseado em Malha (*Mesh-based multicast routing protocol*):** Nesta subclasse, os protocolos de roteamento *multicast* criam uma malha que consiste em um componente conectado da rede que possui todos os receptores em um grupo.

#### 2.6.6 Protocolos de Roteamento Baseados em *Broadcast*

Nos protocolos de roteamento baseados em *broadcast*, as mensagens são enviadas entre os nós usando o *flooding* como o método de transmissão. No *flooding*, cada nó na rede recebe uma mensagem e a retransmite para os nós próximos, o que gera o problema de *broadcasting storm*. Devido a esse problema, várias cópias de uma mesma mensagem são recebidas redundantemente por nós na rede, o que consome largura de banda e recursos de processamento. Além disso, existe uma alta contenção para acessar o canal sem fio, pois todos os nós querem retransmitir a mensagem ao mesmo tempo, o que gera um grande número de colisões na rede e atraso (NI et al., 1999) e (AlMheiri; AlQamzi, 2015).

### 3 TRABALHOS RELACIONADOS

Em redes veiculares cada algoritmo de cluster utilizado possui suas próprias características. Entre elas, a comunicação pode ser V2V e V2I ou somente V2V, a posição do veículo de destino já pode ser conhecida ao ser utilizado um sistema de localização ou métodos para descobrir a localização do veículo de destino são utilizados. Alguns destes algoritmos são projetados para ambientes urbanos e outros para ambiente de rodovia e diferentes métodos de eleição de CH e formação de cluster podem ser utilizados. A seguir são apresentados alguns protocolos de roteamento em cluster.

#### 3.1 CBLR\LORA-CBF

Entre os algoritmos de cluster projetados para redes veiculares, o CBLR (SANTOS; EDWARDS; SEED, 2002) e o *Location Routing Algorithm with Cluster - Based Flooding* (LORA-CBF) (Santos et al., 2004) possuem o mecanismo de formação de cluster e método de roteamento semelhantes, portanto neste trabalho eles são referidos como CBLR\LORA-CBF.

No CBLR\LORA-CBF cada veículo determina sua posição ao utilizar o GPS ou algum sistema de coordenadas local e mensagens *hello* são trocadas periodicamente via *broadcast* entre veículos de um salto de distância. Estas mensagens possuem informações como endereço, tipo e localização de um nó vizinho e são utilizadas para a formação de clusters e roteamento.

Nestes dois algoritmos, após a formação do cluster, quando o nó de origem deseja enviar um pacote de dados para um nó de destino, primeiro ele verifica a tabela de roteamento para determinar se sabe a localização do destino. Caso saiba a posição, o nó de origem envia o pacote para o nós mais próximo do destino. Quando a posição do nó de destino não está em sua tabela, o nó de origem armazena o pacote de dados em seu buffer, inicia um *timer* e transmite um pacote *Location Request* (LREQ). Ao receber uma LREQ, um CH verifica se o nó de destino é membro do seu cluster. Ao ser encontrado o nó de destino, o CH envia um pacote *Location Reply* (LREP) para o nó de origem usando um nó mais próximo dele.

O nó de origem ao receber o pacote LREP com a localização do destino, recupera o pacote de dados do seu buffer e o envia para o vizinho mais próximo do destino. Porém, se o nó de destino não for encontrado no cluster do CH, retransmissões do pacote LREQ deste CH para CHs adjacentes são realizadas por meio de nós do tipo gateway. Os gateways retransmitem apenas um pacote de um gateway para outro para minimizar retransmissões desnecessárias e

apenas se o gateway pertencer a um cluster head diferente, além disso caso um pacote LREQ que já foi recebido anteriormente, seja recebido novamente por um CH, ele é descartado.

### 3.2 Hierarchical Cluster Based (HCB)

O protocolo de roteamento *Hierarchical Cluster Based* (HCB) (Xia; Yeo; Lee, 2009), inicialmente proposto para rede MANET heterogênea altamente móvel, possui uma hierarquia de comunicação de duas camadas e considera dois tipos de nós na rede e que nenhuma informação de GPS está disponível. Os nós do tipo 1 possuem apenas uma interface sem fio IEEE 802.11g cuja o alcance de transmissão é de 100 m. Estes nós formam clusters na camada 1. Os nós do tipo 2, também chamados de *supernodes*, possuem uma interface sem fio de curto alcance IEEE 802.11g como os nós do tipo 1 e uma interface de longo alcance como *Worldwide Interoperability for Microwave Access* (WiMAX) ou *High Speed Packet Access* (HSPA) que possui um alcance de 1000 m. *Supernodes* existem na camada 1 e na camada 2 e agem como cluster heads na camada 1, além disso esses nós podem comunicar entre si diretamente ou por meio de uma estação base na camada 2.

Durante a formação e manutenção de clusters, na camada 1, onde a comunicação é estabelecida através de caminhos multi-hop, cada cluster head transmite periodicamente mensagens do tipo *HEARTBEAT*. Uma mensagem *HEARTBEAT* é identificada por um número de sequência e contém o endereço do cluster head e a distância em número de saltos até ele. Inicialmente esta distância é zero e a topologia é desconhecida. Na formação do cluster os nós do tipo 1 ingressam no cluster mais próximo ao receber uma mensagem *HEARTBEAT*. Isso é feito examinando a distância em saltos e o endereço do cluster head, recebidos na mensagem. Ao ser verificado pelos nós, o número de sequência e a distância em saltos, cada mensagem *HEARTBEAT* é retransmitida apenas uma vez, sempre em direção ao limite do cluster.

Na descoberta de topologia, um nó detecta a presença de seus vizinhos quando retransmite mensagens *HEARTBEAT* e ao examinar esse tipo de mensagem ele pode saber a distância em saltos de seus vizinhos até o CH. Com base nessas informações, um nó se torna um *Boundary Node* (BN) caso nenhum de seus vizinhos do mesmo cluster tenha uma maior distância em saltos do que ele próprio. O *boundary node* inicia o processo de descoberta de topologia periodicamente usando mensagens do tipo *SCAN*. Esta mensagem contém a topologia de rede parcial de um *boundary node* para o cluster head e a medida que ela é propagada passa a incluir mais nós e links. O cluster head ao receber a mensagens *SCAN*, pode combinar a topologia par-

cial de todas as mensagens desse tipo recebidas e enviá-la na próxima mensagem *HEARTBEAT*. Porém no que trata do roteamento inter-cluster, os cluster heads trocam entre si periodicamente informações de associação usando uma mensagem do tipo *MEMBERSHIP* por meio de um link de longo alcance. As mensagens *MEMBERSHIP*, além de serem usadas para atualizações periódicas, elas também são enviadas sob demanda quando há alterações dos membros do cluster.

No encaminhamento de dados, cada nó em um cluster pode preencher sua tabela de roteamento para fazer o roteamento intra-cluster com base na topologia das mensagens *HEARTBEAT* usando algoritmos de caminho mais curto. Cada rota calculada a partir das informações de topologia possui um tempo de vida útil. Sempre uma rota é removida quando o seu tempo expira. Caso um nó não encontre a rota para o destino de um pacote, os pacotes de dados serão encaminhados para uma rota padrão e eventualmente, para o cluster head. Isso ocorre quando o nó de destino está em um cluster diferente. Neste caso, o cluster head encaminha o pacote para o cluster head do cluster onde se encontra o nó de destino. Ao receber o pacote de dados este CH o encaminha para o nó de destino.

### 3.3 Cluster Based Routing (CBR)

O *Cluster Based Routing* (CBR) proposto por Luo, Zhang e Hu (2010) e implementado e testado por Alowish et al. (2017), é um protocolo de roteamento baseado em posição e cluster. Ao ser implementado, a área geográfica é dividida em *grids* lógicos quadrangulares que possuem um ID exclusivo e a posição central de cada *grid* é fixada de acordo com o GPS ou mapa digital. Os veículos em cada *grid* formam um cluster e em cada cluster além de nós comuns, existe apenas um cluster head que pode ser um veículo ou uma RSU. No CBR quando um nó de origem S deseja enviar um pacote para um nó de destino D, ele não precisa descobrir a rota e armazená-la em tabela. O nó S define no pacote a localização do nó de destino e o envia para o cluster head identificado na sua tabela de cluster head.

Quando o cluster head recebe o pacote de dados, ele pesquisa em sua tabela de cluster head e seleciona o cluster head vizinho ideal entre os clusters vizinhos de acordo com ângulo mínimo e encaminha o pacote de dados para este CH. Porém se o nó S for um CH de um cluster, ele seleciona o CH ideal de um cluster vizinho também de acordo com o critério de ângulo mínimo e envia o pacote de dados diretamente para ele.

O pacote de dados é encaminhado entre CHs dos grids sempre tendo como premissa a seleção do próximo cluster head vizinho ideal, até que o pacote chegue ao cluster head do

cluster onde se encontra o nó de destino D. Então o CH do cluster onde se encontra este nó, envia o pacote de dados diretamente para ele.

### 3.4 Traffic Infrastructure Based Cluster Routing Protocol with Handoff (TIBCRPH)

No *Traffic Infrastructure Based Cluster Routing Protocol with Handoff* (TIBCRPH) (Tiecheng Wang; Gang Wang, 2010), os nós conhecem a sua localização por meio de algum tipo de serviço de posicionamento, como GPS. Além disso, este protocolo considera um serviço de localização que possibilita ao nó de origem detectar a posição e a velocidade do nó de destino.

Neste protocolo, os equipamentos de infraestrutura de tráfego existentes são utilizados como cluster heads para dividir a rede em clusters. Cada cluster possui um CH e um grupo de membros. A união dos CHs de cada cluster formam uma rede de *backbone* que abrange uma grande área com várias ruas. A comunicação entre os CHs pode ser com ou sem fio por serem equipamentos de infraestrutura fixos. Porém, devido a aproximação dos clusters que é causada pela movimentação dos veículos, existem áreas de sobreposição entre eles que são definidas como áreas de interferência.

Uma métrica de *handoff* é utilizada quando os veículos estão em áreas de interferência. Ao utilizar esta métrica, ela é calculada por meio do produto escalar do vetor de velocidade de um veículo e do vetor de direção dos dois CHs vizinhos. Por meio do produto escalar é possível estimar o ID de um CH mais adequado para o veículo. Posteriormente, o CH deste ID é selecionado como o CH do veículo.

Em um cluster do TIBCRPH, os membros comunicam diretamente entre si e quando uma mensagem de dados precisa ser enviada de um nó de origem para um nó de destino que está em outro cluster, a posição do destino assim como ocorre no CBR, também é armazenada na mensagem que é transmitida entre os CHs do backbone até chegar ao CH do destino que entregará a mensagem para ele.

### 3.5 Cluster-Based Directional Routing Protocol (CBDRP)

O *Cluster-Based Directional Routing Protocol* (CBDRP) (Tao Song et al., 2010) é um protocolo proposto para o cenário de rodovia. Nele os veículos que possuem uma mesma direção de movimento são divididos em clusters e em cada cluster é selecionado um CH. No procedimento de roteamento desse protocolo, o CH encaminha os pacotes *route request* (RREQ),

pacotes *Route Reply* (RREP) e pacotes de dados para o CH vizinho, além de manter uma tabela de roteamento e informações de membros de seu cluster. Este procedimento é dividido em quatro partes, a solicitação de roteamento, estabelecimento de roteamento, manutenção de roteamento e desconexão de link.

Quando um nó de origem S deseja enviar pacotes de dados ao nó de destino D, ele primeiramente pergunta para o CH se o nó D está no mesmo cluster, caso esteja o nó S envia a mensagem de dados diretamente para o nó D. Porém, se o nó D não estiver no mesmo cluster, o nó S inicia a solicitação de roteamento, enviando um pacote RREQ para o CH do seu cluster. O CH então seleciona um CH de encaminhamento de acordo com a direção de transmissão. Posteriormente, o CH de encaminhamento ao receber o pacote RREQ, verifica se o nó D está no seu cluster. Caso esteja, este CH encaminha o pacote RREQ para D, caso não estiver, o CH de encaminhamento continuará verificando se o nó D está em seu cluster até que o *Time to Live* (TTL) seja igual a zero. Posteriormente, um novo CH de encaminhamento é selecionado para encaminhar o pacote RREQ. Nesta etapa todos os CHs precisam adicionar a lista de rotas de encaminhamento e a lista de rotas reversa à tabela de rotas ao receber pacotes RREQ. No estabelecimento de roteamento, o nó de destino D ao receber o pacote RREQ, responde com um pacote RREP ao nó de origem S, de acordo com a tabela de roteamento reverso, o que permite o estabelecimento de um link da origem para o destino.

Na manutenção de roteamento, sempre que um link é interrompido devido à velocidade veicular, um nó intermediário ao detectar tal problema, usa a estratégia *store-and-forward*, ou seja, armazena brevemente a mensagem de dados e envia o RREQ para o nó de destino D que responde com um pacote RREP no tempo definido, dessa forma o reparo é realizado. Caso o nó intermediário não receba uma resposta do nó de destino D, ele envia um pacote *Route Error* (REER) ao nó de origem S, que fará um novo procedimento de solicitação de roteamento. Por último, na desconexão de link, quando a comunicação é encerrada, o link é desconectado.

### 3.6 Algoritmo Baseado em Distância Euclidiana

No algoritmo baseado em distância euclidiana, proposto por Tian et al. (2010), para o ambiente de rodovia, cada veículo possui um GPS que fornece a posição do veículo, velocidade e tempo preciso baseado em *Coordinated Universal Time* (UTC). Neste algoritmo, informações topológicas sobre a rede são obtidas por meio de *hello beacons* que são transmitidos pelos veículos.

Após a descoberta de topologia usando *hello beacons*, o algoritmo de *clustering* gera os CHs e divide os veículos que se movem em uma mesma direção em clusters calculando a distância euclidiana entre cada veículo e cada CH. Se a distância euclidiana entre um veículo e um CH for menor que a distância limite, eles fazem parte de um mesmo cluster. O processo de formação de clusters é executado periodicamente.

Ao ser formados os clusters, quando um veículo de origem S quer enviar uma mensagem para um veículo de destino D, ele primeiro verifica o veículo D em sua tabela de topologia. Se o veículo S encontrar o registro do veículo D, ele calcula a qual cluster pertence o veículo D. Se eles estiverem no mesmo cluster, o veículo S usará um método de roteamento reativo como AODV ou *Dynamic Source Routing (DSR)* para descobrir a rota para o veículo D. Neste caso, a mensagem RREQ é transmitida apenas entre os veículos que pertencem ao mesmo cluster. Porém, se o veículo S e o veículo D estiverem em clusters diferentes, mas se moverem na mesma direção, a mensagem RREQ será transmitida entre os veículos em direção ao veículo D.

### 3.7 Comparação entre os Protocolos e a Solução Proposta

Após serem analisados os algoritmos descritos, são destacados quatro problemas que podem afetar a implementação das redes VANET ou prejudicar a comunicação destas redes.

Primeiro, devido à mudança de trajetória e mudança de velocidade constantes dos veículos, o tempo de vida de um cluster pode ser pequeno e o processo de *reclustering* pode ocorrer com uma maior frequência, algo que é dispendioso quando mensagens são trocas esporadicamente. Com a exceção do TIBCRPH que utiliza CHs estáticos baseados em infraestrutura e uma técnica de *handoff* que permite aos veículos encontrarem novos CHs e assim novos clusters rapidamente, os demais algoritmos descritos devido as mudanças de CHs e de veículos entre clusters, podem sofrer com este problema.

Segundo, deseja-se no futuro que os veículos se comuniquem durante sua locomoção pelas cidades o que implica gastos com infraestrutura caso seja utilizada a comunicação V2I. A implementação de infraestrutura em grandes áreas urbanas pode inviabilizar a implementação de redes VANET devido ao seu alto custo. Alguns algoritmos que podem usar infraestrutura são o HCB (Estação base utilizada na comunicação entre CHs) e CBR (CH como RSU). No caso do TIBCRPH ele é totalmente dependente de infraestrutura para operar (CHs são equipamentos de infraestrutura).

Terceiro, a comunicação entre nós em áreas urbanas usando um sinal de longo alcance pode sofrer com interferências causadas por prédios e árvores. Neste caso, o algoritmo HCB ao possibilitar uma comunicação entre CHs diretamente ao ser utilizada uma interface sem fio que possui um alcance de 1000 metros, pode sofrer com este problema.

Quarto, o roteamento entre veículos de clusters que viajam em uma mesma direção de trânsito em uma rodovia é interessante, pois embora exista o problema de aumento de velocidade, existe a possibilidade de permanecerem por mais tempo juntos, diferente do que ocorre quando veículos de sentidos diferentes de trânsito querem se comunicar ou formar um cluster neste ambiente. Porém quando o ambiente é urbano, veículos que desejam se comunicar podem estar em regiões diferentes da cidade e em pistas nas quais a direção do trânsito também é diferente, portanto este tipo de roteamento poderia não ser eficiente ou adaptável. Exemplos de algoritmos de roteamento direcional em rodovia são o CBDRP e Algoritmo Baseado em Distância Euclidiana.

Ao observar estes problemas é proposto neste trabalho, um algoritmo de roteamento para área urbana que possui as seguintes características:

- Os clusters são delimitados e definidos por ruas e não por direção e velocidade, o que aumenta o tempo de vida de cada cluster;
- Um backbone de carros estacionados é utilizado para rotear mensagens entre carros em movimento, o que diminui gastos na implementação de redes veiculares já que somente o tipo de comunicação V2V é utilizado;
- Quando um dado precisa ser enviado de um CH para um CH2, isto é feito usando roteamento geográfico guloso, o que dispensa um sinal de longo alcance;
- O roteamento é feito em qualquer direção de trânsito;

No quadro Quadro 3.1 a seguir, são comparadas algumas métricas dos algoritmos descritos com a solução proposta. Estas métricas são: O tipo de comunicação, ambiente de comunicação, se mensagens de controle são trocadas entre CHs de um mesmo nível, se o roteamento baseado em posição é utilizado e se a posição do nó de destino já é conhecida pelo o nó de origem ao ser utilizado um sistema de localização.

Quadro 3.1 – Comparação entre os Trabalhos Relacionados

<b>Protocolo</b>	<b>Tipo de Comunicação</b>	<b>Ambiente de Comunicação</b>	<b>Mensagens de Controle entre CHs</b>	<b>Roteamento Baseado em Posição</b>	<b>Posição do nó de Destino Conhecida</b>
CBLR/LORA-CBF	V2V	Rodovia	Sim	Sim	Não
HCB	V2V e V2I	-	Sim	Não	Não
CBR	V2V e V2I	Ambiente Urbano	Não	Sim	Sim
TIBCRPH	V2V e V2I	Ambiente Urbano	Não	Sim	Sim
CBDRP	V2V	Rodovia	Sim	Não	Não
Algoritmo Baseado em Distância Euclidiana	V2V	Rodovia	Não	Não	Não
Solução Proposta	V2V	Ambiente Urbano	Não	Sim	Não

Fonte: autor (2020)

## 4 ROTEAMENTO HIERÁRQUICO BASEADO EM SETORES (RHBS)

Neste capítulo inicialmente é apresentada uma visão geral do algoritmo proposto e posteriormente suas etapas são detalhadas.

### 4.1 Visão Geral

O protocolo de *Roteamento Hierárquico Baseado em Setores* (RHBS) é proposto para o ambiente urbano. Ele não utiliza infraestrutura para a troca de mensagens e possibilita roteamento em qualquer direção. O RHBS, possui uma hierarquia de dois níveis de CHs na implementação testada, porém pode ser escalável em mais níveis. Esta hierarquia ao ser utilizada possibilita que veículos de um mesmo setor ou de setores diferentes comuniquem entre si, além disso diminui a sobrecarga ocasionada por requisições feitas a um controlador e também diminui o *overhead* de mensagens de controle quando comparada com métodos de roteamento que utilizam *flooding*. Para o roteamento de mensagens controle e de dados, o RHBS utiliza dois modos de roteamento geográfico guloso sobre um *backbone* formado por veículos estacionados.

#### 4.1.1 Tipos de Nós e Definição de Setor

No algoritmo proposto, denominado como RHBS, são considerados dois tipos de nós na rede veicular. O primeiro tipo é composto por carros moveis em circulação que desejam comunicar entre si e o segundo por carros estacionados que formam um backbone que é utilizado para roteamento. O segundo tipo ainda é dividido em carros estacionados comuns que somente reencaminham uma mensagem e CHs que além de reencaminhar uma mensagem, possuem informações sobre os carros moveis. Por fim, estes CHs ainda podem ser classificados como Cluster Head (CH) ou Cluster Head de primeiro nível e Cluster Head de Segundo Nível (CH2). Os CHs de primeiro nível possuem informações sobre nós moveis do seu setor e fornecem a posição de um veículo móvel de destino quando requisitado. Já um CH2 possui informações de veículos moveis de todos os setores e fornece a posição requisitada de um veículo móvel de destino quando um CH a desconhece. Um setor é composto e delimitado por um conjunto de ruas de duplo sentido de trânsito que em sua maioria possuem estacionamento e que ao agrupar veículos moveis e estacionados sobre a gerência de um CH, também é um cluster.

### 4.1.2 Localização e Eleição de CHs

Um veículo ao locomover-se em uma rua, periodicamente consulta um sistema de localização que combina mapa de ruas e GPS, o que possibilita saber em que setor se está, além de também descobrir o *id* e posição do CH deste setor. No contexto do RHBS, um veículo ao mudar de rua pode continuar em um mesmo setor, já que um setor é composto por um conjunto de ruas ou mudar de setor quando em seu trajeto entra em uma rua que não pertence ao setor pelo qual se trafegava antes. O tamanho do setor que é utilizado no protocolo impacta no roteamento de mensagens até o CH. Diversos tamanhos de setores podem ser utilizados. Para fins de avaliação deste trabalho foi utilizado o tamanho de uma quadra para o setor.

No que refere-se a eleição de nós como CH, a ideia é que o nó mais próximo de uma determinada posição de um setor seja eleito nesta função. Porém, como a rede é dinâmica o CH pode ser alterado. Portanto faz-se necessário um algoritmo para tratar a manutenção do Cluster Head. No entanto, isto não é tratado no escopo do trabalho sendo então utilizados veículos estacionários como CHs e CH2.

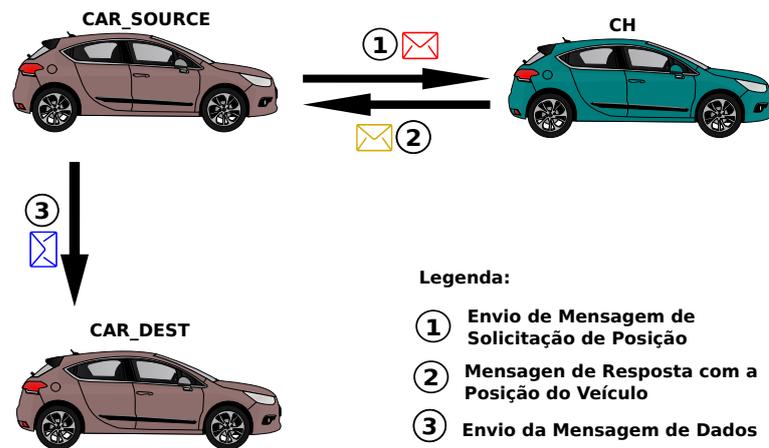
### 4.1.3 Troca de Mensagens de Controle e Envio da Mensagem de Dados

Quando um veículo estaciona, ele se anuncia para outros veículos estacionados e passa a fazer parte de um backbone. O backbone de carros estacionados é utilizado no roteamento geográfico para encaminhar mensagens *hello*, mensagens de atualização de tabela, mensagens de requisição de posição e mensagens de dados. Uma mensagem do tipo *hello* é enviada por um veículo em movimento para o CH do seu setor a fim de informar sua posição. Porém, para que o envio desta mensagem seja possível, um nó vizinho estacionado é descoberto e usado como primeiro nó de encaminhamento. Mensagens de atualização de tabela são trocadas entre CHs e CH2, para que o CH2 tenha o conhecimento dos veículos móveis de todos os setores. Tanto mensagens *hello* como de atualização de tabela são trocadas periodicamente.

Uma mensagem de requisição de posição é enviada por um veículo de origem em movimento (CAR\_SOURCE) para o CH do seu setor com a finalidade de requisitar a posição de um veículo móvel de destino (CAR\_DEST). Do mesmo modo como ocorre nas mensagens do tipo *hello*, nas mensagens de requisição de posição, um nó vizinho estacionado é descoberto e utilizado como primeiro nó de encaminhamento. O CH ao receber a mensagem verifica se conhece a posição requisitada do veículo móvel de destino. Caso o CH possua o registro do veículo requisitado, ele então envia uma mensagem de resposta com a posição requisitada para

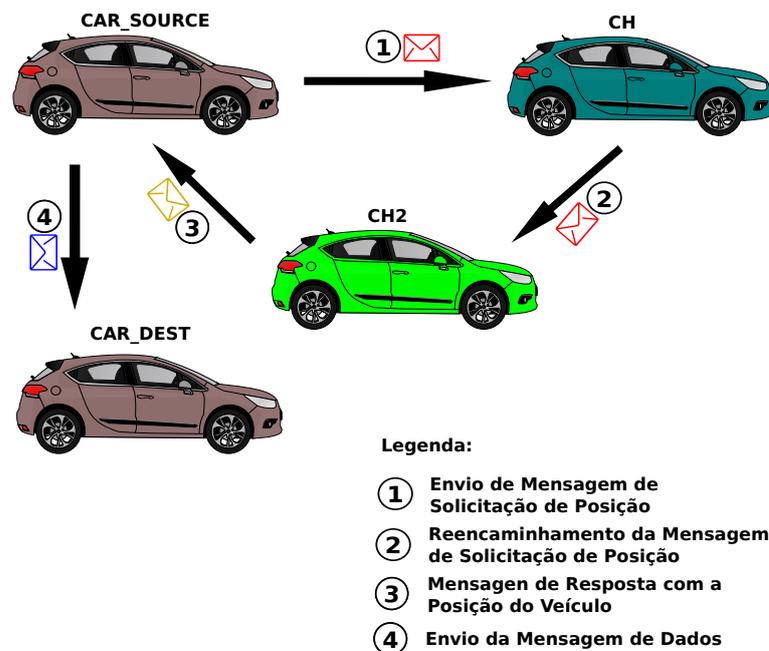
o veículo móvel de origem que então enviará uma mensagem de dados para o veículo móvel de destino (FIGURA 4.1). Porém, caso o CH do setor não possua o registro do veículo móvel de destino, ele reencaminhará a mensagem de requisição de posição para o CH2. Caso o CH2 possua o registro do veículo requisitado, ele então envia uma mensagem de resposta para o veículo móvel de origem que ao obtê-la, então encaminhará a mensagem de dados para o veículo móvel de destino (FIGURA 4.2).

Figura 4.1 – Troca de Mensagens Usando o CH



Fonte: autor (2020)

Figura 4.2 – Troca de Mensagens Usando CH e CH2



Fonte: autor (2020)

#### 4.1.4 Tabelas Utilizadas pelo Protocolo

No protocolo desenvolvido, um conjunto de tabelas é utilizado para armazenar informações do nós. As principais tabelas são `neighborsTable`, `responseTable`, `membersTable`, `updateTable` e `l2Table`. A tabela `neighborsTable` é utilizada por veículos estacionados para armazenar informações de veículos vizinhos que também estão estacionados. Estes nós vizinhos durante o roteamento serão usados como saltos até o nó de destino. A tabela `responseTable` é utilizada por carros estacionados para armazenar dados de um veículo móvel para posteriormente se anunciar como um vizinho que será utilizado como primeiro salto do roteamento geográfico. A tabela `membersTable` é utilizada por um CH de setor para armazenar informações de veículos moveis após receber uma mensagem do tipo *hello*, posteriormente esta tabela é consultada pelo próprio CH para informar para um veículo de origem se conhece a posição de um nó procurado que será utilizado como veículo de destino de uma mensagem. Por fim, a tabela `updateTable` do CH de um setor, periodicamente a cada tempo de `SEND_TIME`, se possuir, envia registros atualizados de veículos para o CH2, para que por meio dela possa ser inserido, atualizado ou removido veículos da tabela `l2Table` que posteriormente assim como a tabela `membersTable`, será consultada para informar a posição de um nó procurado que será utilizado com destino de uma mensagem, caso este não esteja no mesmo setor/cluster do nó de origem. No decorrer do texto a utilização destas tabelas será melhor compreendida ao ser explicado as trocas de mensagens do protocolo.

## 4.2 Etapas do Algoritmo Proposto

A seguir as principais etapas do algoritmo proposto são detalhadas.

### 4.2.1 Descoberta de CH

Quando um veículo está em movimento ou estacionado em uma rua do grid, a cada segundo a sua posição é verificada. A cada verificação de posição, o mapa de ruas é consultado e de acordo com a rua onde o veículo se encontra é descoberto em que setor se está. Posteriormente sabendo-se o setor, a tabela `chTable` que é um complemento do mapa de ruas é consultada. Se o CH antes conhecido pelo veículo for diferente do CH resultante da consulta da tabela `chTable`, o Id deste CH e a sua posição são armazenados por ele, o que pode ser visto no algoritmo 1 da Figura 4.3.

Para um veículo móvel um CH diferente indica que o veículo mudou de setor ao mudar de rua. Porém, existe também um caso em particular que a consulta periódica auxilia. Pode ser que um veículo em movimento ou estacionado estejam em um setor no qual nenhum CH foi definido. Verificações periódicas portanto, permitem conhecer o CH assim que ele for definido. Caso o CH de um setor continuasse desconhecido, mensagens de requisição de posição e *hello* não poderiam ser enviadas por carros moveis e nem mesmo roteadas geograficamente pelo backbone de veículos estacionados até ele.

Figura 4.3 – Algoritmo 1: Descoberta e Atualização de CH

---

```

1 Function whoIsCH():
2   lastSector ← mySector(myRoadId());
3   if chTable.containsKey(lastSector) then
4     if currentCH ≠ lastSectorCH then
5       currentCH ← lastSectorCH;
6       posCurrentCH ← lastSectorCHPos;
7     end
8   end
9 End Function

```

---

Fonte: autor (2020)

#### 4.2.2 Anúncio de Posição de Carros Estacionados

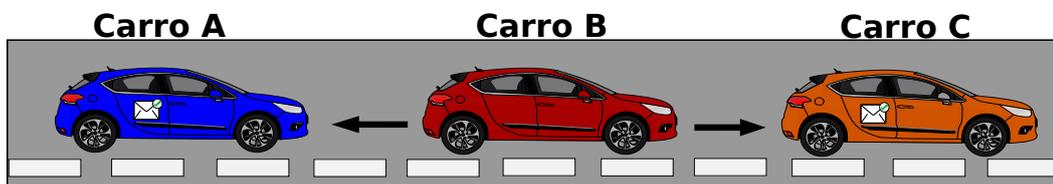
Quando um veículo estaciona, ele envia por *broadcast* um beacon de anúncio de posição. Um nó vizinho ao receber o beacon, inicialmente verifica se está estacionado e depois verifica se o registro do veículo sender está em sua tabela neighborsTable, caso não esteja ele é inserido. Posteriormente, o nó que recebeu o beacon verifica se já passou o tempo de ANNPOS\_CHECK desde que enviou um beacon de anúncio de posição, caso tenha-se passado este tempo, um beacon por *broadcast* também é enviado pelo nó para anunciar sua posição.

Exemplificado, na Figura 4.4, na primeira troca de beacons, o Carro B estaciona então envia um beacon por *broadcast* anunciando sua posição. O Carro A e o Carro C recebem o beacon e o armazenam na tabela neighborsTable. Posteriormente, os dois veículos de acordo com um beckoff aleatório anunciam suas posições via broadcast, pois o tempo passado desde que se anunciaram é maior que ANNPOS\_CHECK. Na segunda troca de beacons, o primeiro

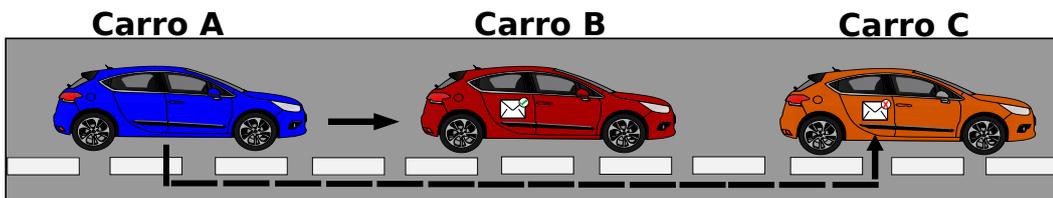
veículo a enviar o beacon após o tempo de *backoff* é o Carro A. O Carro B ao receber o anúncio de posição do Carro A, o armazena na tabela *neighborsTable*. Já o Carro C ao receber o beacon do Carro A o descarta pois o veículo já está registrado em sua tabela. Na terceira troca de beacons, o Carro B recebe o anúncio de posição do Carro C e armazena na tabela *neighborsTable*. Já o Carro A ao receber a mensagem do Carro C a descarta pois o veículo já está registrado em sua tabela. Após as três trocas de beacons, como o tempo decorrido ainda é menor que *ANNPOS\_CHECK*, nenhum outro anúncio de posição será feito pelos três veículos.

Figura 4.4 – Anúncio de Posição

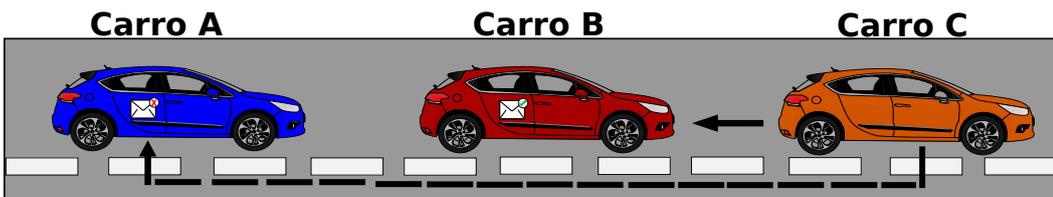
### 1ª Troca de Beacons



### 2ª Troca de Beacons



### 3ª Troca de Beacons



Fonte: autor (2020)

Nos parágrafos acima, pode-se destacar dois aspectos importantes sobre o processo de anúncio de posição. Primeiro, a resposta de anúncio de posição não é unicast, pois ao se ter vários veículos se anunciando e respondendo ao mesmo tempo, várias mensagens são perdidas devido a colisões, o que diminui o número de vizinhos conhecidos por um nó. O envio de respostas de anúncio de posição por *broadcast*, permite não só se anunciar para um novo veículo que estaciona e passa a integrar o backbone, mas também para veículos que já fazem parte do backbone e que perderam a resposta de anúncio anterior. Segundo, o tempo definido de ANN-

POS\_CHECK é para impedir que veículos estacionados reenviem anúncios de forma contínua (em *loop*) um para o outro.

### 4.2.3 Descoberta de um Nó Vizinho Estacionado por um Veículo em Movimento

Quando o CH do setor é conhecido por um veículo em movimento, uma mensagem *broadcast* é enviada para carros estacionados para descobrir um vizinho que possa encaminhar uma mensagem. Esta mensagem a ser encaminhada pode ser uma mensagem de requisição de posição de nó de destino ou uma mensagem *hello*. Quando a mensagem enviada via *broadcast* é recebida por um veículo, ele verifica se está estacionado, caso esteja, ele armazena na tabela *responseTable*, o Id do veículo *sender*. Posteriormente o veículo faz um agendamento pra si com um *backoff* aleatório entre 10 à 100 milissegundos. Ao fim do agendamento, uma mensagem *broadcast* de resposta é enviada informando seu status de vizinho estacionado, caso nenhum outro veículo tenha feito o mesmo antes. Caso um segundo veículo tenha enviado uma mensagem *broadcast* informando seu status de vizinho estacionado, o primeiro cancelará sua resposta e nada será feito ao fim do seu tempo de *backoff*.

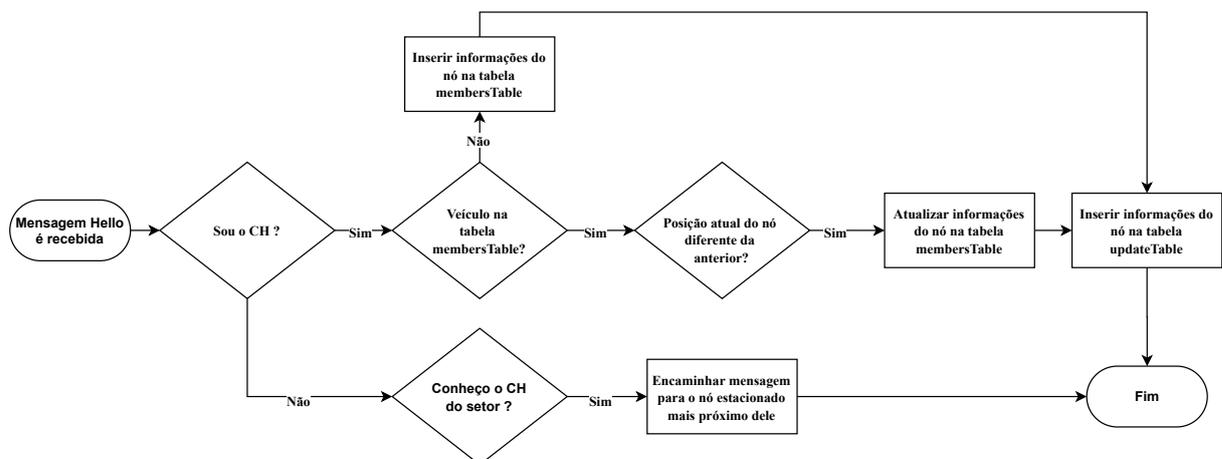
O veículo em movimento ao receber a mensagem de um vizinho estacionado, inicialmente verifica se é o destino da mensagem, posteriormente verifica se está aceitando respostas de vizinhos e se o nó que respondeu é do mesmo setor. Atendidas estas condições, o veículo que respondeu é definido como próximo vizinho de salto e respostas de outros nós estacionados caso sejam recebidas são ignoradas. Após a descoberta de um nó vizinho estacionado uma mensagem de requisição de posição de nó de destino ou uma mensagem *hello* podem ser enviadas usando este nó como primeiro nó de encaminhamento.

É importante salientar que ignorar respostas de outros veículos estacionados e o mecanismo de cancelamento do envio de resposta, são complementares. Quando uma mensagem de resposta é enviada pode ser que um nó próximo ao *sender*, mas distante de quem enviou a mensagem ou ainda devido a colisões, não receba a mensagem e assim não cancele sua resposta. Ao ser ignorar respostas posteriores a primeira, possibilita-se evitar o envio redundante de mensagens usando diferentes nós de salto. A combinação destes dois elementos, diminui o número de colisões já que pode se ter uma grande quantidade de veículos em movimento perguntando sobre seus vizinhos estacionados e uma grande quantidade de veículos estacionados respondendo a questão, caso nenhuma restrição seja definida.

#### 4.2.4 Envio de Posição de Veículos em Movimento para um CH

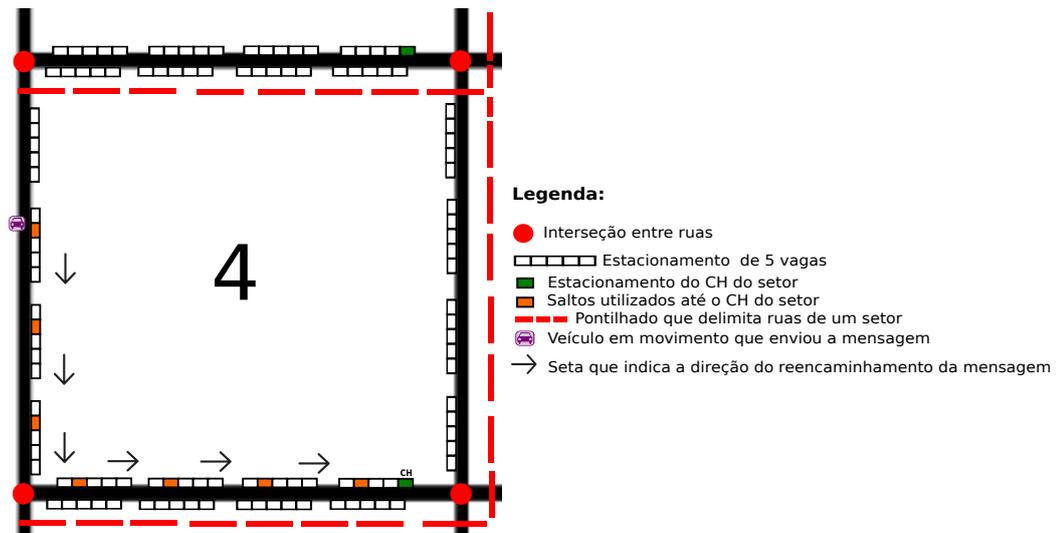
Quando a mensagem *hello* é recebida pelo nó de encaminhamento descoberto anteriormente, ele verifica se é o CH do setor o qual destina-se a mensagem (FIGURA 4.5). Se for, posteriormente ele verifica se o nó já está registrado em sua tabela `membersTable`, se estiver e a posição atual deste veículo é diferente da posição anterior armazenada, ele atualizará na tabela, o tempo de envio da mensagem, a posição atual e o status que indica se está estacionado ou em movimento. Além disso, o id do veículo, a posição atual e o id do setor, são inseridos na tabela local `updateTable` a qual posteriormente será enviada para o CH2. Porém, caso não exista registro do nó na tabela `membersTable`, o tempo de envio da mensagem, a posição atual e o status que indica se está estacionado ou em movimento são inseridos na tabela e da mesma forma que o que ocorre na atualização do registro, o id do veículo, a posição atual e o id do setor, são inseridos na tabela local `updateTable`. Se o nó não for o CH do setor, mas se o conhece, reenvia a mensagem para o nó estacionado mais próximo dele ao utilizar roteamento geográfico guloso.

Figura 4.5 – Etapas do Encaminhamento da Mensagem *Hello*



Fonte: autor (2020)

Na Figura 4.6, utilizando o setor 4 como exemplo é mostrado como o corre o reencaminhamento de uma mensagem *hello* até o CH do setor. Os nós estacionados que possuem a vaga de estacionamento na cor laranja reencaminham a mensagem até que ela chegue na vaga de estacionamento na cor verde onde está o CH.

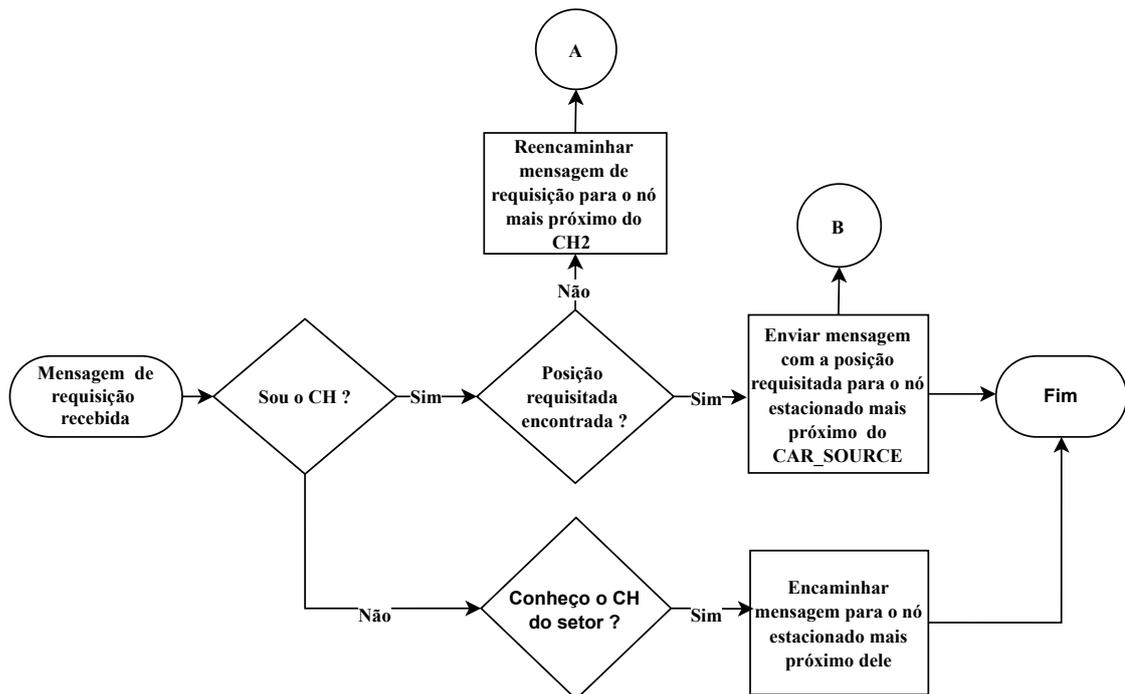
Figura 4.6 – Envio de uma Mensagem *Hello* para o CH do Setor

Fonte: autor (2020)

#### 4.2.5 Descoberta da Posição de um Nó de Destino

Quando a mensagem de requisição de posição de nó de destino é recebida pelo nó de encaminhamento descoberto anteriormente, ele verifica se é o CH do setor o qual destina-se a mensagem (FIGURA 4.7). Se este veículo for o CH, ele procurará em sua tabela `membersTable` a posição do nó requisitada pelo veículo móvel de origem. Se a posição requisitada do nó for encontrada, o CH envia uma mensagem de resposta para o nó mais próximo do veículo móvel de origem com a posição encontrada. Caso o nó seja o CH e a posição requisitada não seja encontrada, a mensagem de requisição de posição é enviada para o nó estacionado mais próximo do CH<sup>2</sup>. Porém, caso o nó que recebeu a mensagem de requisição de posição, não seja o CH do setor, ele enviará a mensagem para o nó mais próximo dele.

Figura 4.7 – Etapas de Encaminhamento da Requisição até um CH

**Legendas:**

Encaminhamento da mensagem de requisição por veículos estacionados até o CH2.

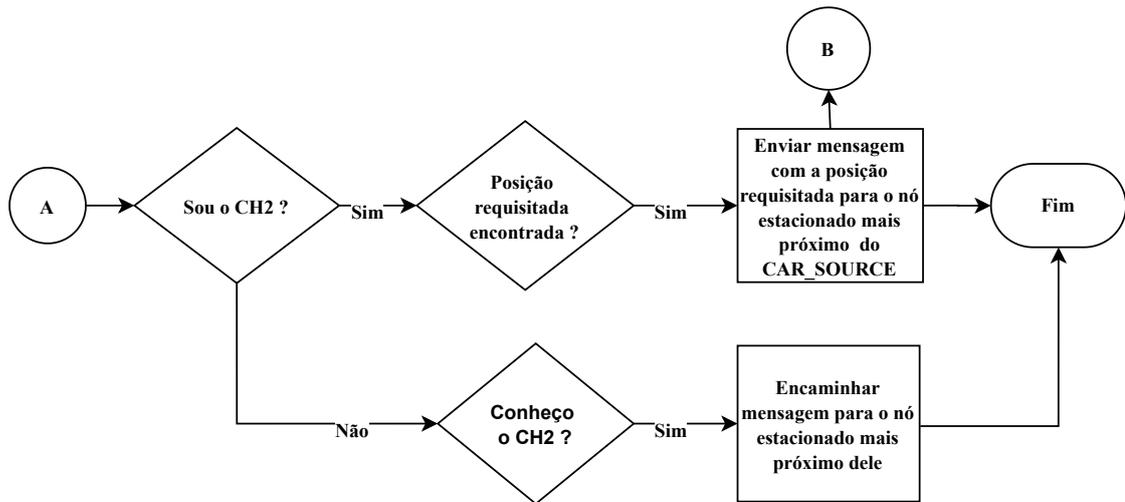


Encaminhamento da mensagem com a posição requisitada por veículos estacionados até o CAR\_SOURCE.

Fonte: autor (2020)

Ao ser recebida a mensagem de requisição de posição encaminha geograficamente de um CH com destino para o CH2, o nó verifica se é o CH2 (FIGURA 4.8). Se este veículo for o CH2, ele procurará em sua tabela l2Table a posição do nó requisitada pelo veículo móvel de origem. Se o nó requisitado for encontrado, o CH2 envia uma mensagem de resposta para o nó mais próximo do veículo móvel de origem com a posição encontrada do nó requisitado. Caso a posição não seja encontrada, nenhuma resposta é enviada. Porém, caso o nó que recebeu a mensagem de requisição de posição, não seja o CH2, ele reenviará a mensagem para o nó mais próximo dele.

Figura 4.8 – Etapas de Encaminhamento da Requisição até um CH2

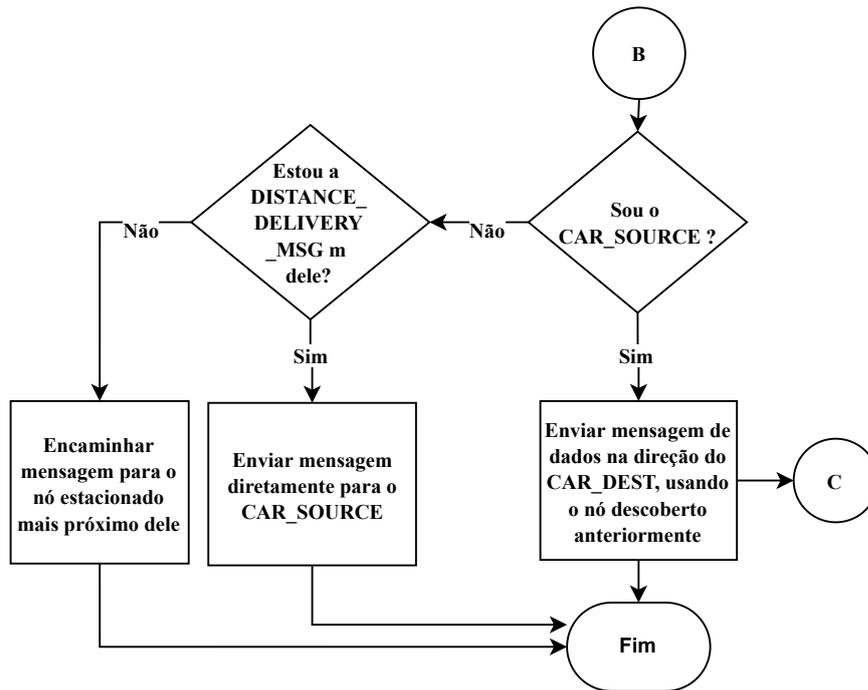
**Legendas:**

- A** Encaminhamento da mensagem de requisição por veículos estacionados até o CH2.
- B** Encaminhamento da mensagem com a posição requisitada por veículos estacionados até o CAR\_SOURCE.

Fonte: autor (2020)

Um nó ao receber a mensagem de resposta do CH ou CH2, verifica se ele é o veículo móvel de origem (FIGURA 4.9). Se for, ele envia uma mensagem de dados em direção do endereço geográfico do veículo móvel de destino recebido, usando como primeiro nó de salto, o mesmo veículo descoberto para o envio da mensagem de requisição de posição. Porém, caso a mensagem seja recebida por um nó estacionado que esteja a *DISTANCE\_DELIVERY\_MSG* metros do veículo móvel de origem, ele enviará a mensagem diretamente para este nó. Caso a mensagem seja recebida por um nó estacionado que não esteja a *DISTANCE\_DELIVERY\_MSG* metros do veículo móvel de origem, ele reencaminha a mensagem para o nó estacionado mais próximo dele.

Figura 4.9 – Etapas do Encaminhamento da Mensagem de Resposta

**Legendas:**

Encaminhamento da mensagem com a posição requisitada por veículos estacionados até o CAR\_SOURCE.

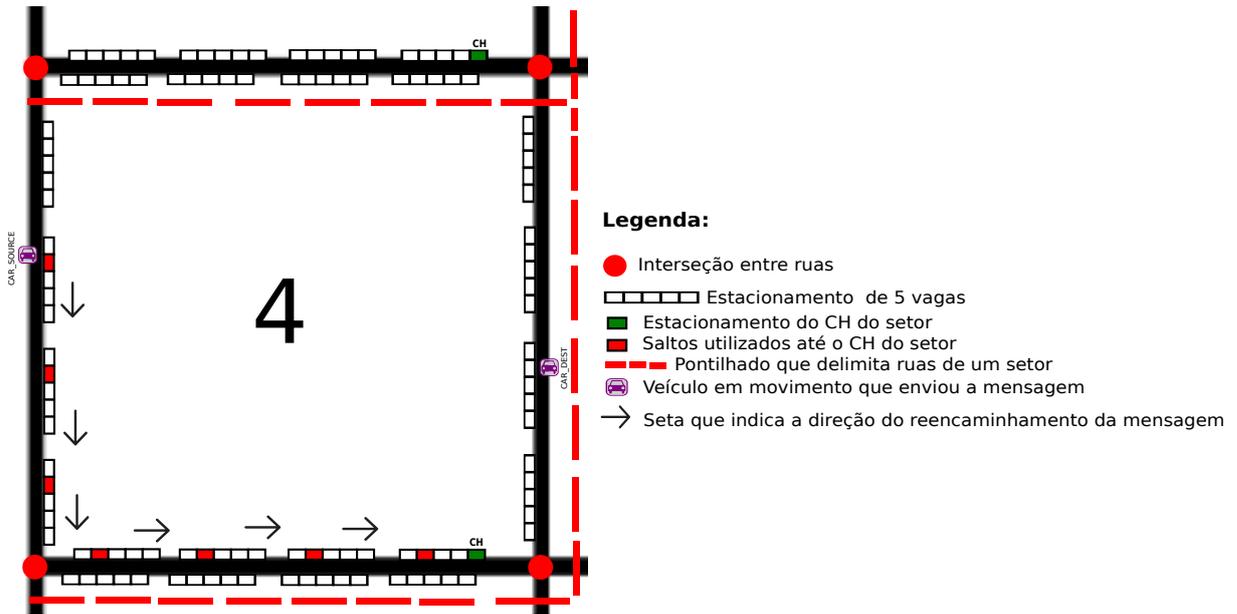


Encaminhamento da mensagem de dados por veículos estacionados até o CAR\_DEST.

Fonte: autor (2020)

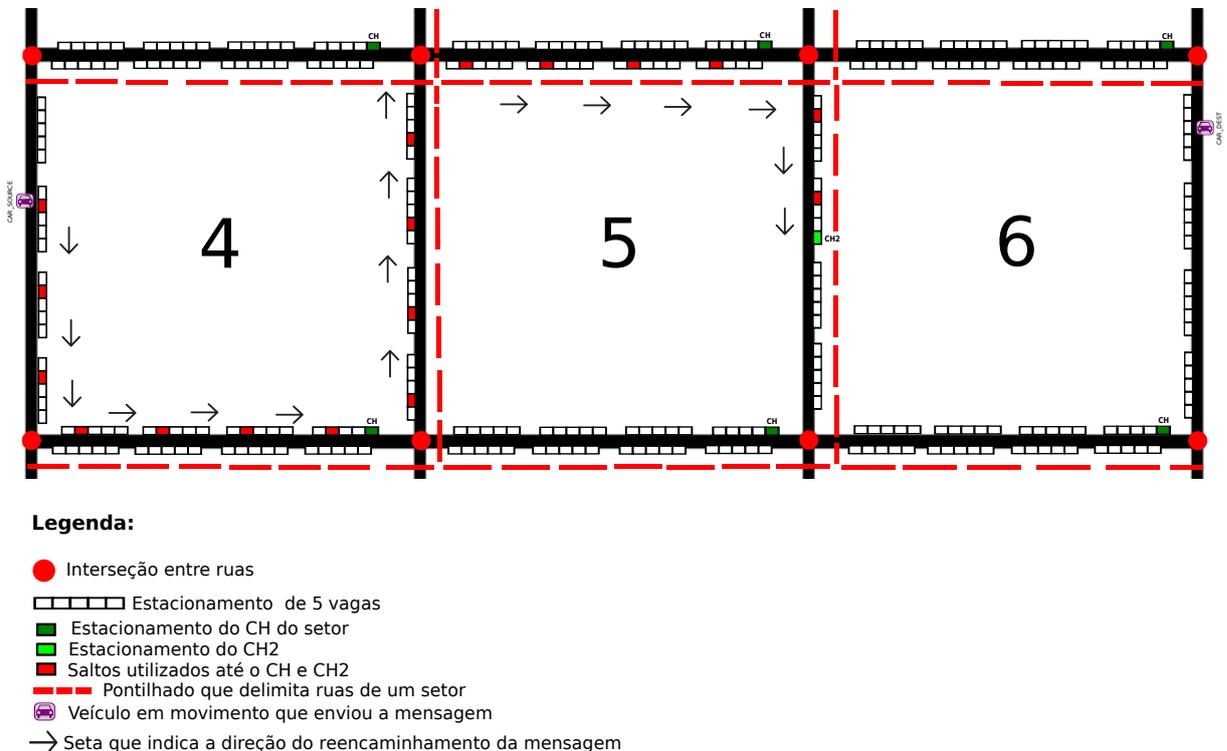
Na Figura 4.10, utilizando o setor 4 como exemplo é mostrado como o corre o reencaminhamento de uma mensagem de requisição de posição do veículo móvel de destino até o CH do setor. Os nós estacionados que possuem a vaga de estacionamento na cor vermelha reencaminham a mensagem até que ela chegue na vaga de estacionamento na cor verde onde está o CH. Na Figura 4.11, como o CH do setor 4 não sabe a posição do veículo móvel de destino pelo fato deste não fazer parte do seu setor, ele reencaminha a solicitação para o CH2. Os nós estacionados que possuem a vaga de estacionamento na cor vermelha reencaminham a mensagem até que ela chegue na vaga de estacionamento na cor verde-claro onde está o CH2.

Figura 4.10 – Envio de uma Mensagem Solicitando a Posição do Veículo Móvel de Destino



Fonte: autor (2020)

Figura 4.11 – Envio de uma Mensagem Solicitando a Posição do Veículo Móvel de Destino para o CH2

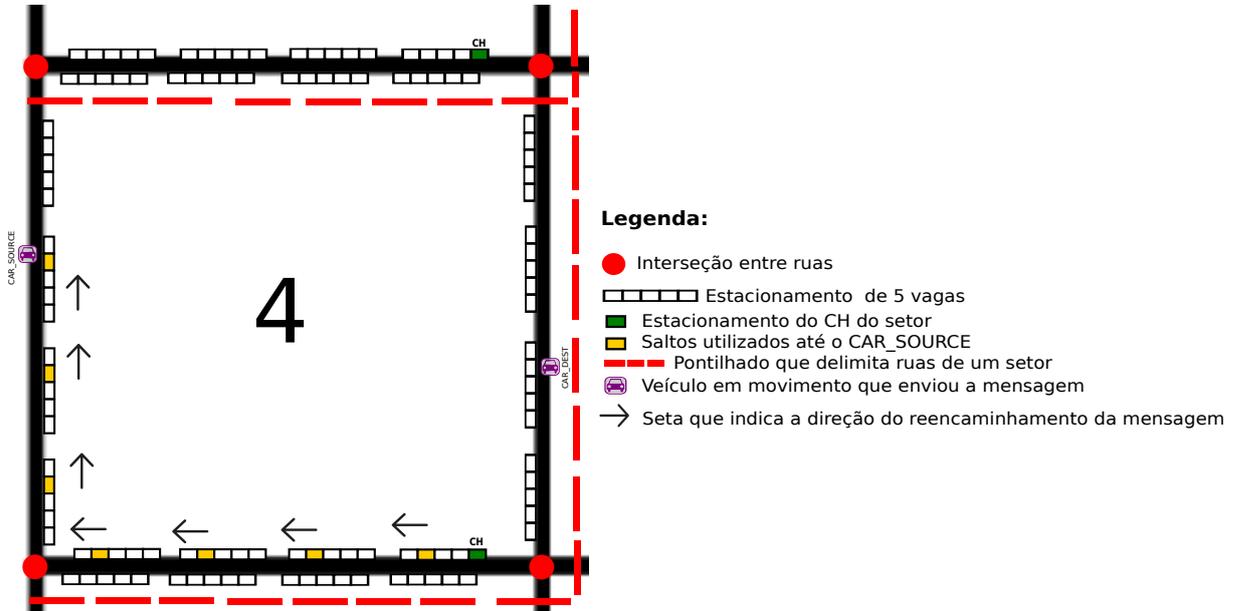


Fonte: autor (2020)

A Figura 4.12 mostra o reencaminhamento da resposta do CH para o veículo móvel de origem quando a posição do veículo móvel de destino solicitada é encontrada pelo fato dos dois veículos estarem em um mesmo setor/cluster. A Figura 4.13 mostra o reencaminhamento da

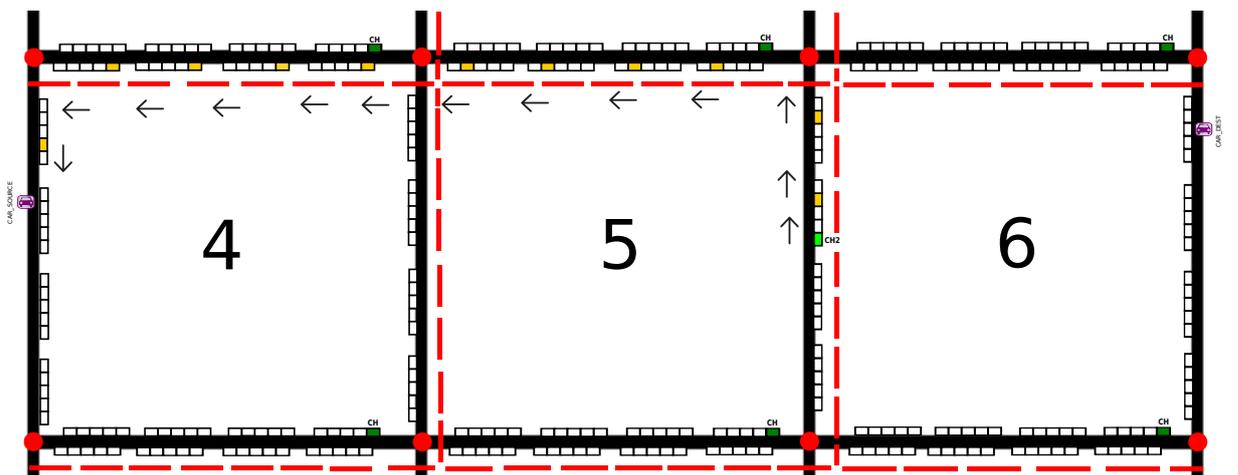
resposta do CH2 para o veículo móvel de origem, quando o veículo de origem e destino estão em setores/clusters diferentes. Em ambos os casos, os nós estacionados que possuem a vaga de estacionamento na cor amarela reencaminham a mensagem até que ela chegue ao veículo.

Figura 4.12 – Resposta do CH para o Veículo Móvel de Origem



Fonte: autor (2020)

Figura 4.13 – Resposta do CH2 para o Veículo Móvel de Origem



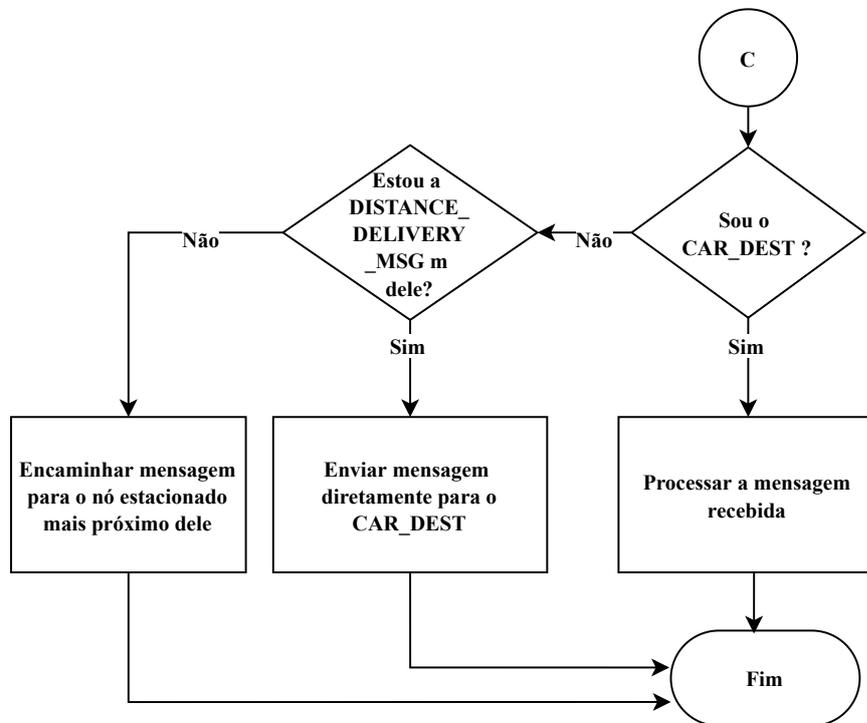
- Legenda:**
- Interseção entre ruas
  - ▭ Estacionamento de 5 vagas
  - Estacionamento do CH do setor
  - Saltos utilizados até o CAR\_SOURCE
  - - - Pontilhado que delimita ruas de um setor
  - 🚗 Veículo em movimento que enviou a mensagem
  - Seta que indica a direção do reencaminhamento da mensagem

Fonte: autor (2020)

#### 4.2.6 Recebimento de uma Mensagem de Dados

Ao ser recebida a mensagem de dados, enviada pelo veículo móvel de origem, o nó verifica se é o veículo móvel de destino (FIGURA 4.14). Se for, a mensagem chegou ao destino. Caso a mensagem seja recebida por um nó estacionado que esteja a *DISTANCE\_DELIVERY\_MSG* metros do veículo móvel de destino, este nó enviará a mensagem diretamente para ele. Caso a mensagem seja recebida por um nó estacionado que não esteja a *DISTANCE\_DELIVERY\_MSG* metros do veículo móvel de destino, ele reencaminhará a mensagem para o nó estacionado mais próximo dele.

Figura 4.14 – Etapas do Encaminhamento da Mensagem de Dados



#### Legenda:



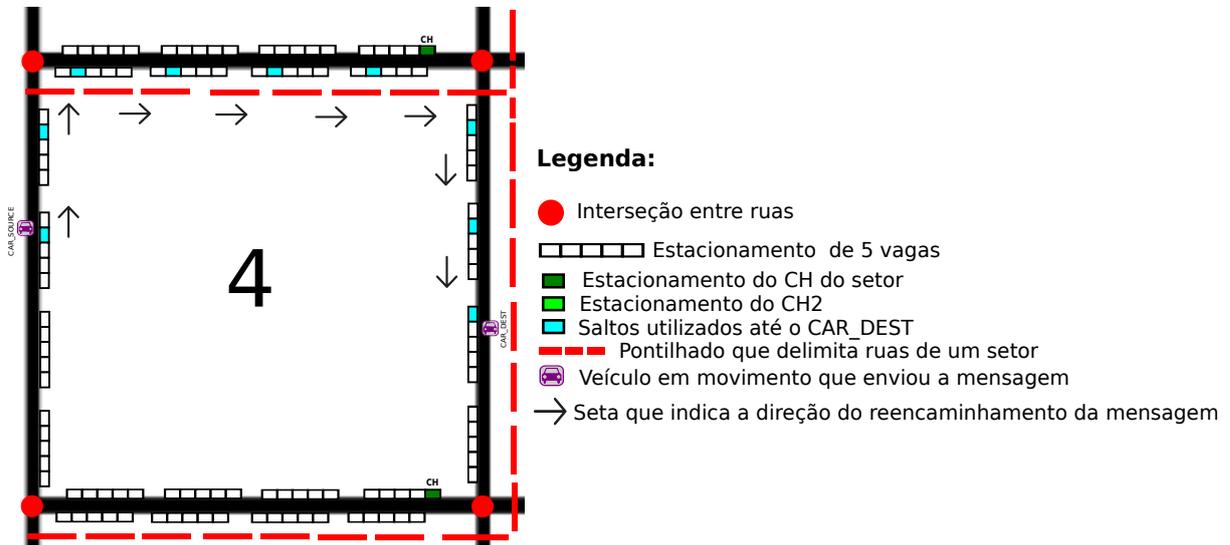
Encaminhamento da mensagem de dados por veículos estacionados até o *CAR\_DEST*.

Fonte: autor (2020)

A Figura 4.15 mostra o reencaminhamento da mensagem do veículo móvel de origem para o veículo móvel de destino em um mesmo setor/cluster. A Figura 4.16 mostra o reencaminhamento da mensagem do veículo móvel de origem para o veículo móvel de destino, quando os dois veículos estão em setores/clusters diferentes. Em ambos os casos, os nós estacionados

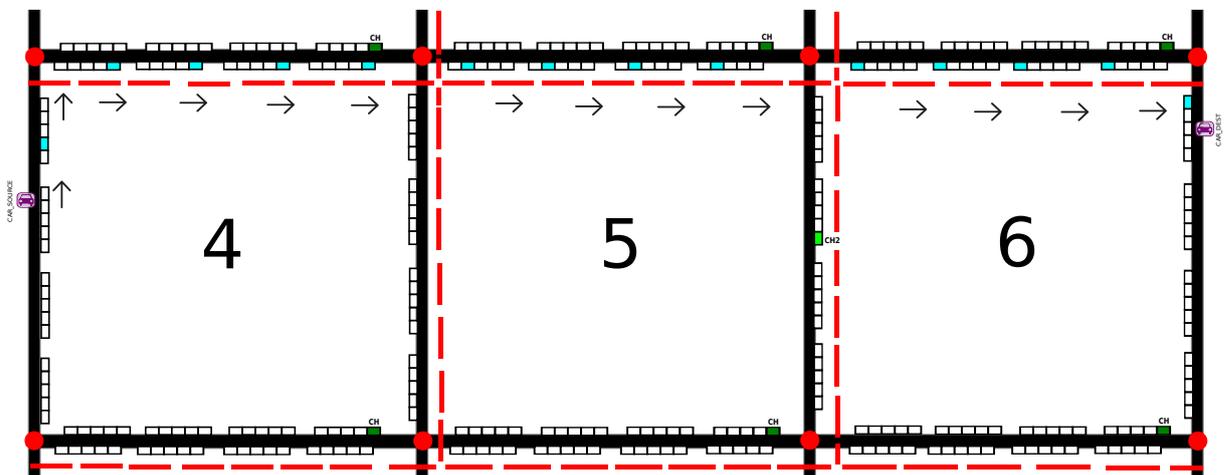
que possuem a vaga de estacionamento na cor azul ciano reencaminham a mensagem até que ela chegue ao veículo móvel de destino.

Figura 4.15 – Envio de Mensagem do Veículo Móvel de Origem para o Veículo Móvel de Destino em um Mesmo Setor



Fonte: autor (2020)

Figura 4.16 – Envio de Mensagem do Veículo Móvel de Origem para o Veículo Móvel de Destino em Setores Diferentes



**Legenda:**

- Interseção entre ruas
- Estacionamento de 5 vagas
- Estacionamento do CH do setor
- Estacionamento do CH2
- Saltos utilizados até o CAR\_DEST
- Pontilhado que delimita ruas de um setor
- 🚗 Veículo em movimento que enviou a mensagem
- Seta que indica a direção do reencaminhamento da mensagem

Fonte: autor (2020)

#### 4.2.7 Envio de Posição de Veículos em Movimento para o CH2

Periodicamente a cada tempo de `SEND_TIME`, o CH verifica o tempo passado desde o último recebimento de uma mensagem *hello* enviada por um nó móvel. Se o tempo passado for maior que `MAINTENANCE_THRESHOLD`, o registro do veículo é removido da tabela `membersTable` e o seu Id e o id do seu setor, são inseridos na tabela `updateTable` que depois será enviada para o CH2 para que ele remova o registro de sua tabela `l2Table`. A tabela `updateTable` a ser enviada, além de possuir registros a serem removidos, pode possuir também registros a serem inseridos ou atualizados como visto no tópico 4.2.4.

Posteriormente, após ser feita a verificação de remoção de nós, o CH verifica se a tabela `updateTable` possui registros, se sim, descobre um nó vizinho estacionado mais próximo do CH2, ao realizar um cálculo que utiliza a posição dos nós vizinhos estacionados armazenados na tabela `neighborsTable` e a posição do CH2. Posteriormente, uma mensagem contendo a tabela `updateTable` é enviada usando o nó descoberto como salto. Após a tabela `updateTable` ser enviada para o CH2, o CH limpa os registros da tabela local que possui para que mais tarde ela seja atualizada e reenviada novamente.

Quando a mensagem é recebida, o nó verifica se é o CH2, se for, os registros da tabela `updateTable` recebidos por meio de mensagem devem ser inseridos, atualizados ou removidos da tabela `l2Table` do CH2. No caso da remoção, o CH2 verifica se o id do setor de um veículo recebido via mensagem é igual ao id do setor deste mesmo veículo armazenado anteriormente na sua tabela `l2Table`, se for o nó é removido desta tabela. Esta verificação evita que o pedido de remoção de um nó por um CH de um setor antigo no qual o veículo trafegava antes, remova da tabela `l2Table` do CH2, o nó quando este esteja trafegando em um novo setor. Caso esta verificação não fosse realizada, o nó poderia ser removido e readicionado na tabela mesmo quando o seu registro já estivesse atualizado para o setor atual, após o CH2 receber por mensagem, atualizações de veículos enviadas pelo CH responsável. O nó só é removido da tabela `l2Table` quando este em seu trajeto sai da área do grid. Porém, se o veículo que recebeu a mensagem não for o CH2, a mensagem é reencaminhada para o nó estacionado mais próximo dele.

### 4.3 Roteamento Geográfico Detalhado

#### 4.3.1 Modos do Roteamento Geográfico

No algoritmo proposto, quando um nó que recebe a mensagem não é o nó de destino, ela é sempre reencaminhada para um outro nó até que a mensagem chegue ao destino. Para isto é usado o Greedy Geographic Routing, que pode ser traduzido como roteamento geográfico guloso, roteamento geográfico ganancioso ou ainda roteamento geográfico ávido. No RHBS, são definidos dois tipos de roteamento guloso. O primeiro tipo, G1 (Greedy tipo 1), da mesma forma que outros algoritmos gulosos, reencaminha a mensagem sempre para um nó mais próximo do destino. O segundo tipo, G2 (Greedy tipo 2), reencaminha a mensagem sempre para um nó mais próximo da interseção de uma rua e é usado para se sair de uma situação de local máximo.

Inicialmente no RHBS, conforme é mostrado no fluxograma da Figura 4.18, os nós reencaminham a mensagem usando o roteamento G1. Para encontrar o nó de salto mais próximo do destino é feito um cálculo simples utilizando a posição dos nós vizinhos estacionados armazenados na tabela `neighborsTable` e a posição do destino. Se o nó encontrado for diferente do sender de quem se recebeu a mensagem, ela é encaminhada para ele e continua-se a transmissão no roteamento G1. Porém, se o nó encontrado for o próprio sender, um máximo local é identificado e a mensagem não é reencaminhada pra ele.

Para resolver este problema, o nó ao ter o conhecimento de sua distância até o início da via obtida por meio do mapa de ruas e da distância dos nós vizinhos até o início da via obtida por meio do anúncio de posição, pode encontrar o nó mais próximo de uma de suas interseções. Para isto de acordo com o algoritmo 2 da Figura 4.17, o veículo verifica se sua distância em relação ao início da via é menor ou igual à metade do tamanho dela, caso seja ele procurará na tabela `neighborsTable`, o nó mais próximo da interseção do início da via. Caso sua distância seja maior, ele procurará em sua tabela o vizinho mais longe da interseção do início da via, porém mais próximo da interseção do fim dela. Ao ser encontrado um nó, o roteamento muda para o roteamento G2 e a mensagem é reenviada para ele. Ao estar nesse segundo tipo de roteamento guloso, a mensagem continuará sendo reencaminhada até chegar em um nó de borda de uma interseção. O nó da borda ao receber a mensagem retorna para o roteamento G1 e calcula o nó mais próximo do destino. Posteriormente a mensagem é reencaminhada para o nó encontrado que será um veículo estacionado em uma rua vizinha ligada a intercessão.

Figura 4.17 – Algoritmo 2: Descoberta de um Nó Próximo a uma Interseção

---

```

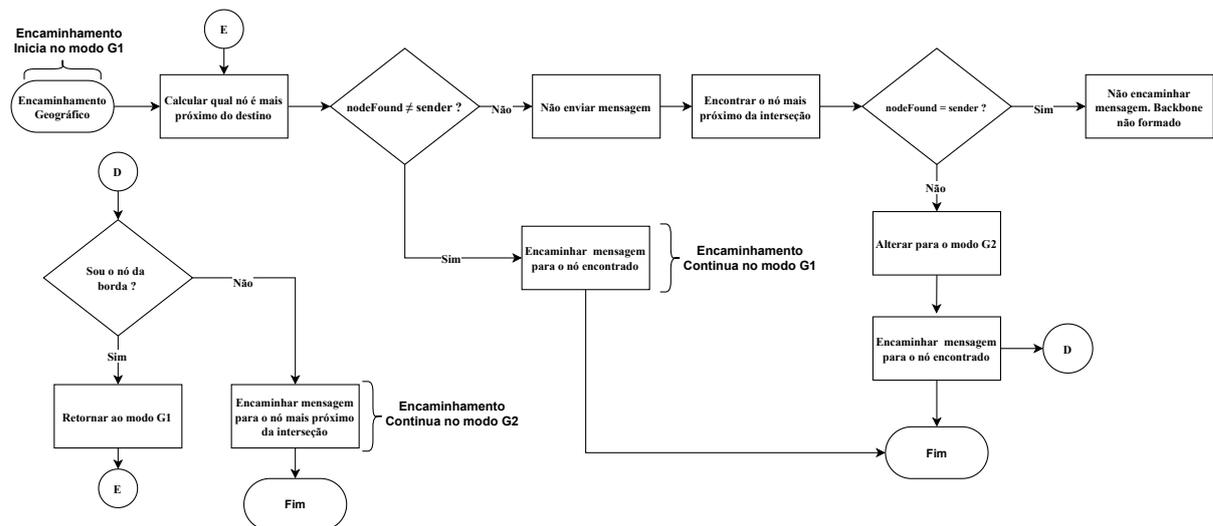
1 Function findNeighborCloseToJunction():
2   if myCurrentDistance <= LANE_LENGTH/2 then
3     |   getNeighborInitJunction();
4   else
5     |   getNeighborEndJunction();
6   end
7 End Function

```

---

Fonte: autor (2020)

Figura 4.18 – Roteamento Geográfico com Alteração de Modos



Fonte: autor (2020)

No trabalho é admitido um backbone formado por carros estacionados, no qual todas as vagas de estacionamento estão ocupadas. Em um cenário no qual o backbone de veículos está em formação, os tipos de roteamento G1 e G2, não conseguem entregar todas as mensagens, isso ocorre devido ao número limitado de veículos estacionados. Em muitos casos, o único nó mais próximo que pode ser utilizado é o próprio sender devido as lacunas existentes nos estacionamentos.

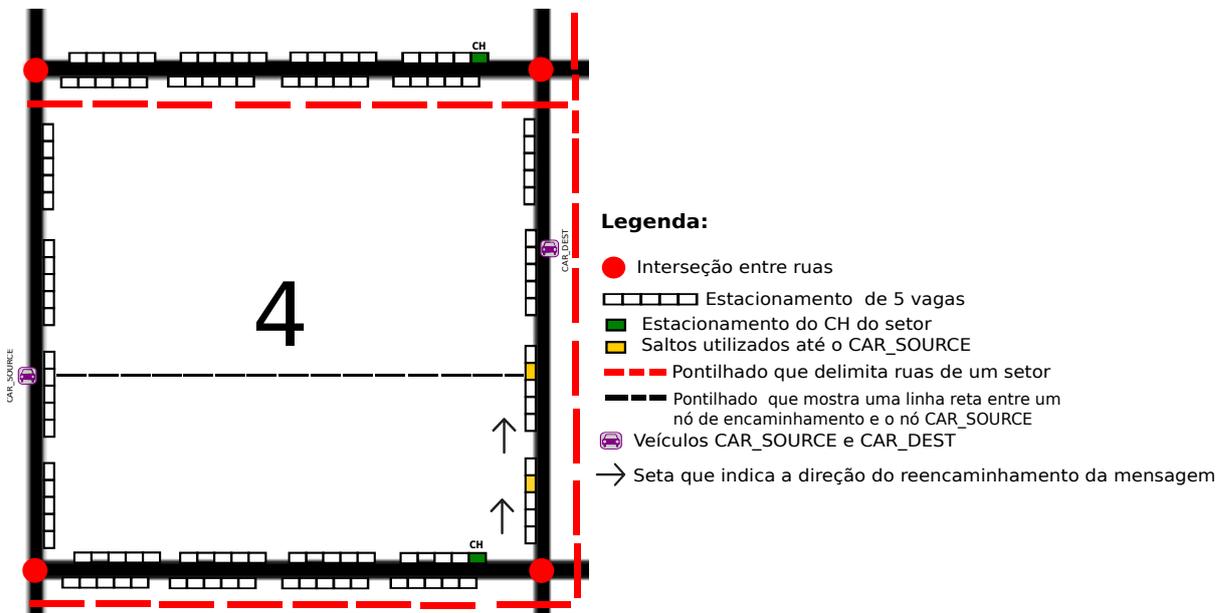
#### 4.3.2 Situações de Ocorrência de Máximo Local

Quando uma mensagem está sendo encaminhada no roteamento geográfico guloso, pode ser que o nó de encaminhamento fique em paralelo com o nó de destino móvel, tendo-se o centro

do setor/grid como um buraco entre eles. Neste caso, o nó de encaminhamento é o nó mais próximo do destino, mas como ele não consegue enviar a mensagem diretamente pelo alcance do seu sinal, ele deseja devolver a mensagem para o nó de quem recebeu a mensagem (sender) por achar que este é o nó mais próximo do destino além dele, este problema é o que chamamos de máximo local. Porém, caso não houvesse nenhum mecanismo de controle e a mensagem fosse devolvida para o nó sender, ele reenviaria a mensagem para o mesmo nó novamente. O reencaminhamento da mensagem entre os dois nós continuaria até a mensagem ser perdida.

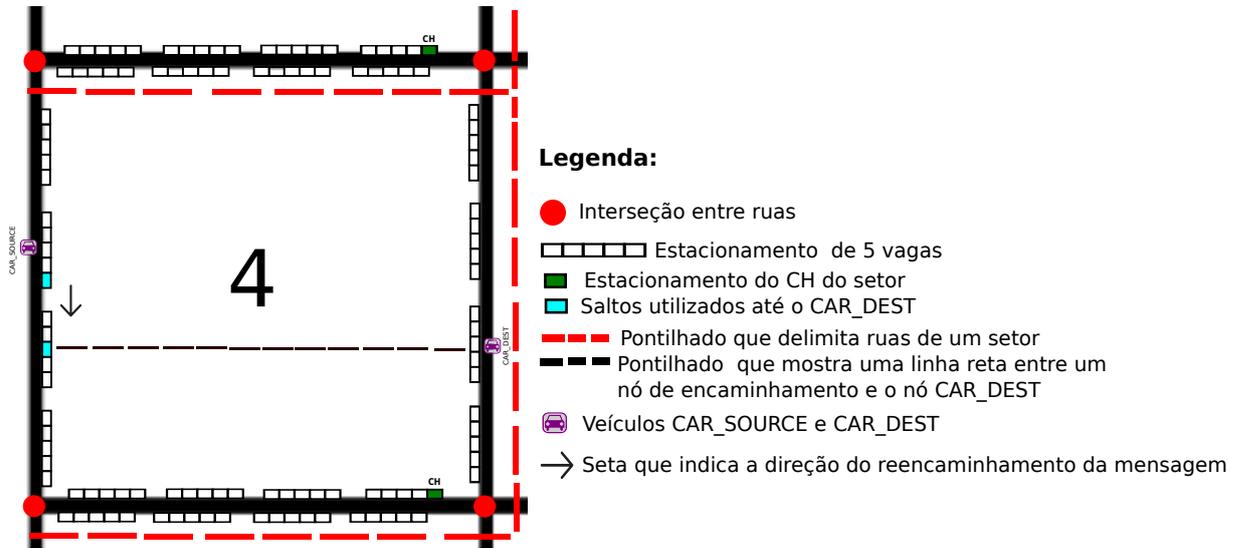
No algoritmo proposto detectamos duas situações em que isso ocorre. A primeira situação ilustrada pela Figura 4.19, no momento de encaminhamento da mensagem com a posição do veículo móvel de destino pelo CH para o veículo móvel de origem. A segunda situação ilustrada pela Figura 4.20, no momento de encaminhamento da mensagem de dados do veículo móvel de origem para o veículo móvel de destino. Em ambos os casos, um nó de encaminhamento fica em paralelo com o nó de destino da mensagem que pode ser o veículo móvel de origem ou veículo móvel de destino. Como descrito anteriormente, a troca entre os modos de roteamento geográfico G1 e G2, permite escapar da situação de máximo local para que a mensagem possa ser entregue para um nó de encaminhamento que possa reencaminhar a mensagem.

Figura 4.19 – Máximo Local no Envio de Mensagem do CH para o Veículo Móvel de Origem



Fonte: autor (2020)

Figura 4.20 – Máximo Local no Envio de Mensagem do Veículo Móvel de Origem para o Veículo Móvel de Destino



Fonte: autor (2020)

## 5 METODOLOGIA

### 5.1 Simuladores Utilizados

Para o desenvolvimento do protocolo RHBS foram utilizados em conjunto, o simulador de rede *Objective Modular Network Testbed in C++* (OMNeT++) (VARGA, 2001), o simulador de tráfego rodoviário *Simulation of Urban Mobility* (SUMO) (KRAJZEWICZ et al., 2012) e o *framework Vehicular in Network Simulation* (Veins) que possibilita o desenvolvimento de protocolos de roteamento ao interligar os simuladores OMNeT++ e SUMO (SOMMER; GERMAN; DRESSLER, 2011). O protocolo AODV, comparado com o RHBS, foi desenvolvido pelo *Grupo de Redes Ubíquas* (GRUBi), utilizado o simulador GrubiX e o traço de mobilidade de veículos exportado do SUMO. O traço de mobilidade possibilitou a atualização de posições de veículos durante seus trajetos e permitiu também que o cenário de *grid* utilizado no RHBS fosse utilizado no AODV.

#### 5.1.1 OMNeT++

OMNeT++ é uma biblioteca e estrutura de simulação C++ extensível, modular e baseada em componentes que utiliza eventos discretos e orientação a objetos para o desenvolvimento. Ao ser utilizado é possível simular redes com e sem fio. Ele oferece um IDE baseado em Eclipse e um ambiente de tempo de execução gráfico. Seu uso é livre para o ambiente acadêmico e para fins não lucrativos (VARGA, 2001), (VARGA, 2016) e (OMNET++, 2020).

#### 5.1.2 SUMO

O SUMO é um conjunto de aplicações que são utilizadas para preparar e executar a simulação de um cenário de tráfego veicular. Ao ser utilizado, as redes viárias podem ser geradas usando a aplicação *netgenerate* ou importadas de um mapa digital como o OpenStreetMap (OpenStreetMap contributors, 2019) e convertidas para um formato compatível usando a aplicação *netconvert*. As redes viárias ao serem geradas ou importadas são representadas como grafos, nos quais os vértices são as interseções e as vias de trânsito as arestas. No que se refere aos veículos, o SUMO possibilita a simulação microscópica de tráfego, portanto o comportamento de cada veículo pode ser modelado individualmente (KRAJZEWICZ et al., 2012) e (Lopez et al., 2018).

### 5.1.3 Veins

Veins é um *framework open source* usado em simulações de redes veiculares. Ele permite o acoplamento bidirecional do simulador de rede OMNeT++ e do simulador de tráfego rodoviário SUMO, estendendo cada simulador com um módulo de comunicação dedicado. Durante a simulação, esses módulos de comunicação trocam comandos e traços de mobilidade por meio de conexões TCP. Desta forma, o movimento de veículos no simulador SUMO é refletido como o movimento de nós em uma simulação do OMNeT++ (SOMMER; GERMAN; DRESSLER, 2011) e (VEINS, 2020).

### 5.1.4 GrubiX

O simulador GrubiX Wireless Sensor Networks é uma extensão do simulador de redes ad hoc ShoX (Lessmann; Heimfarth; Janacik, 2008). Este simulador utiliza a linguagem de programação Java para a implementação e a de marcação XML para a configuração e exportação de resultados. Ele é um simulador orientado a eventos discretos que possibilita a utilização do paradigma de orientação a objetos para o desenvolvimento de aplicações. Outra característica presente é o suporte a GUI (Graphical User Interface) que possibilita a configuração de cenários e visualização de rede.

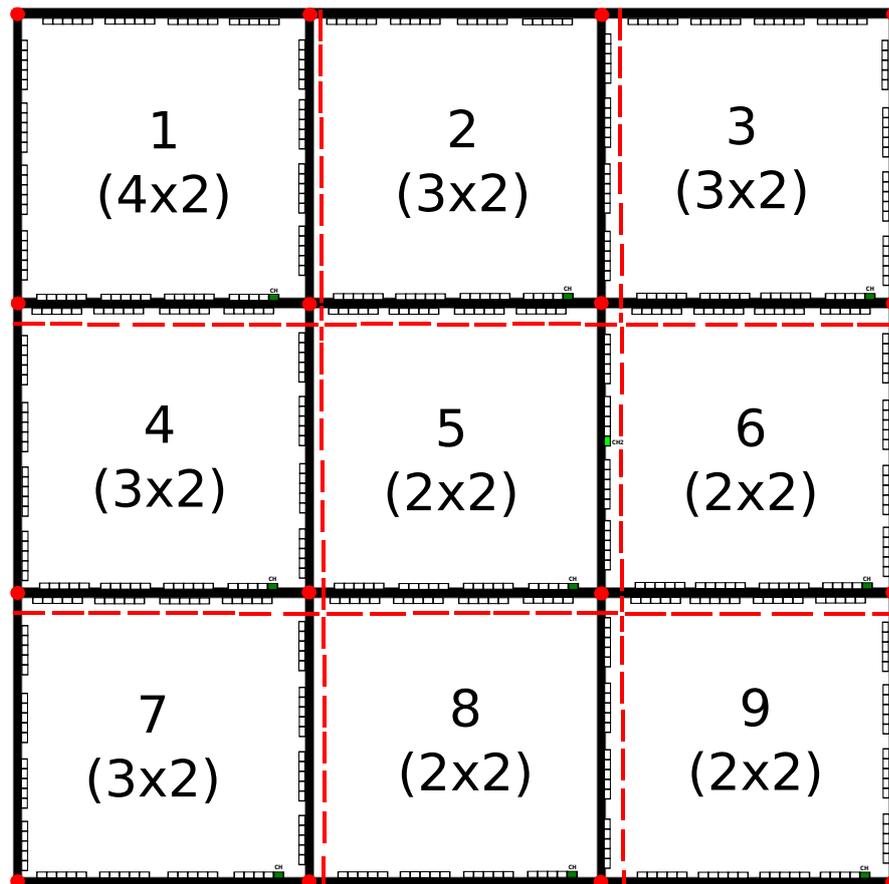
## 5.2 Descrição do Cenário

O cenário utilizado de tráfego de veículos é baseado no modelo de mobilidade Manhattan gerado pela ferramenta *netgenerate* do SUMO. Este cenário é um *grid* composto por nove setores. Cada setor é composto por um conjunto de ruas de duplo sentido com áreas de estacionamento. Cada rua possui o tamanho de 200 metros e cada lateral do grid é composto por 3 ruas o que resulta em uma área com o tamanho de 600m x 600m. Cada área de estacionamento possui 5 vagas e 32 metros de largura. A distância entre estas áreas em uma rua é de 10 metros.

No protocolo RHBS, cada setor é um cluster. Os CHs de cada cluster são posicionados próximos a interseção mais interna de cada setor e o CH2 é posicionado em uma rua lateral no quinto setor. Cada interseção faz parte apenas de um setor para facilitar o mapeamento de ruas no modelo utilizado. No total a simulação do RHBS possui 600 carros estacionados no grid que são utilizados como backbone por 11 carros em movimento que trocam mensagens de dados entre si usando roteamento geográfico guloso (FIGURA 5.1).

No AODV a quantidade total de veículos é a mesma do RHBS e embora estes nós estejam estacionados ou em movimento, os nós estacionados não assumem funções de CH ou CH2 pelo fato deste protocolo usar *flooding* de mensagens para encontrar uma rota para o nó de destino da mensagem.

Figura 5.1 – Cenário com CHs Utilizado pelo RHBS



**Legenda:**

- Interseção entre ruas
- Estacionamento de 5 vagas
- Estacionamentos dos CHs dos setores
- Estacionamento do CH2
- Pontilhado que delimita ruas de um setor
- (Rx F) Quantidade de ruas x Quantidade de faixas de trânsito

Fonte: autor (2020)

### 5.3 Configurações Utilizadas

#### 5.3.1 RHBS

- Envio de mensagens de descoberta de nós móveis de destino e o envio de mensagens de dados inicia-se após o tempo de startup de 112 segundos da simulação, quando o backbone de carros estacionados está formado e os 11 carros testados em movimento estão trafegando pelo grid;
- Veículos enviam mensagem *hello* a partir do tempo de 100 segundos da simulação;
- Um veículo envia uma mensagem *hello* quando entra na área do grid e posteriormente a cada intervalo de SECTOR\_CHECK;
- Nós estacionados permanecem neste estado até o fim da simulação;
- Mensagens de anúncio de posição não são avaliadas estatisticamente por serem disseminadas somente durante a formação do backbone de carros estacionados;
- CHs após estacionarem agendam para enviar mensagens para o CH2, após o tempo de SEND\_TIME\_INITIAL;
- Parâmetros definidos:

Quadro 5.1 – Parâmetros Utilizados no RHBS

Parametro	Valor
ANNPOS_CHECK	1 segundo
SEND_TIME	1 segundo
SEND_TIME_INITIAL	90 segundos
SECTOR_CHECK	2 segundos
MAINTENANCE_THRESHOLD	6 segundos
DISTANCE_DELIVERY_MSG	35 metros
LANE_LENGTH	200 metros

Fonte: autor (2020)

#### 5.3.2 Geral

- Alcance de sinal de 70 metros;

- Tempo de simulação avaliado de 10 segundos para coleta de dados para análise. No RHBS este tempo inicia a partir de 112 segundos. Para o AODV ele inicia com a simulação na qual o backbone de carros estacionados já está formado;
- Os veículos móveis de origem e destino utilizados para a troca de dados são sorteados aleatoriamente;
- Tamanho de mensagens de dados testadas: 230, 1152 e 2304 Bytes;
- Total de simulações por tamanho de mensagem de dados: 45 RHBS e 300 AODV;
- Uma comunicação entre um veículo móvel de origem e um veículo móvel de destino por simulação;
- *Bitrate* definido de acordo com cada padrão de comunicação implementado nos simuladores utilizados. 9 Mbps para o RHBS que utiliza o padrão de comunicação 802.11p e 11 Mbps para AODV que utiliza o padrão de comunicação 802.11b;

## 6 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados das simulações realizadas. Inicialmente são comparados os resultados do RHBS com o AODV. Posteriormente, no RHBS de acordo com a utilização dos CHs e CH2, são comparados os resultados obtidos nas etapas de requisição de posição e de entrega da mensagem de dados, além de uma avaliação geral de todo o tempo da simulação ser realizada.

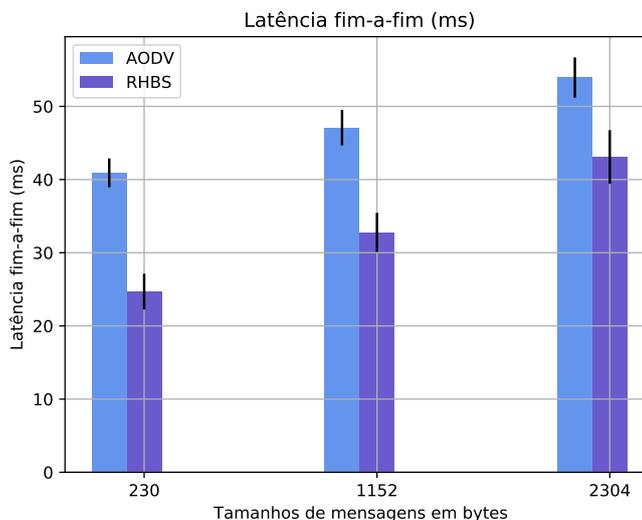
### 6.1 AODV X RHBS

#### Latência Fim-a-fim

Na Figura 6.1 é avaliada a latência fim-a-fim (latência média geral). No RHBS, esta latência corresponde ao período que compreende a descoberta do nó de encaminhamento e o recebimento da mensagem de dados pelo nó de destino. No AODV a latência avaliada, corresponde ao período que compreende o início do *flooding* de mensagens RREQ e o recebimento da mensagem de dados pelo nó de destino.

De forma resumida, no AODV após ser realizado o *flooding* de mensagens RREQ via *broadcast*, uma destas mensagens chega ao nó de destino ou a um nó que sabe uma rota até o nó de destino. Quando isto ocorre, uma mensagem de RREP é enviada de forma *unicast* pelo caminho reverso até o nó de origem. Após receber o RREP, o nó de origem envia a mensagem de dados para o nó de destino. Ao se avaliar o protocolo AODV notou-se que ele apresentou um maior tempo de latência devido ao congestionamento de mensagens RREQ no canal de comunicação. Portanto mesmo que uma rota tenha sido encontrada, no momento que a mensagem RREP estava sendo encaminhada, os nós tinham que aguardar a disponibilidade do canal para encaminhar a mensagem. Outro fator que aumentou o tempo de latência do AODV nos testes realizados é o tamanho da mensagem. Este fator também afeta o RHBS. Em ambos os protocolos, a mensagem de tamanho de 2304 bytes teve um tempo de latência maior.

Figura 6.1 – Latência Fim-a-fim



Fonte: autor (2020)

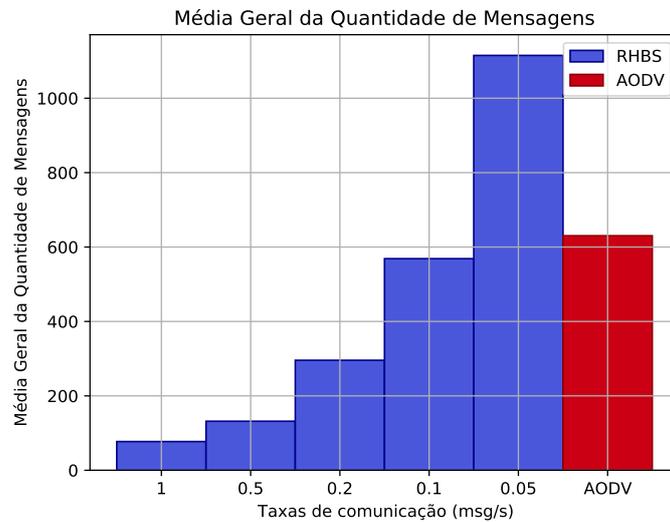
### Média Geral da Quantidade de Mensagens

Ao ser avaliado a quantidade média geral de mensagens, cada protocolo apresentou diferenças mínimas relacionadas aos tamanhos de 230, 1152 e 2304 bytes da mensagem de dados. A Figura 6.2, exibe os resultados para a mensagem de tamanho de 2304 bytes. Nela é comparado a quantidade média geral de mensagens gastas para cada comunicação.

Nesta comparação, uma comunicação do AODV é comparada com cinco diferentes comunicações do RHBS. Cada teste do RHBS utilizou uma taxa de comunicação diferente. As taxas de comunicação do RHBS são: 1 (uma comunicação por segundo), 0.5 (uma comunicação a cada dois segundos), 0.2 (Uma comunicação a cada cinco segundos), 0.1 (uma comunicação a cada 10 segundos) e 0.05 (Uma comunicação a cada 20 segundos). Para cada comunicação com as taxas 1, 0.5, 0.2 e 0.1, o RHBS apresentou uma quantidade média geral de mensagens menor que o AODV. Isto deve-se, ao mecanismo de *flooding* de mensagens utilizado pelo AODV para a descoberta de rota para o nó de destino. Como somente uma mensagem é enviada e os nós são móveis é necessário realizar sempre este mecanismo de *flooding* para encontrar o destino.

No entanto, ao se avaliar a taxa de 0.05 no RHBS, a quantidade média geral de mensagens é maior que a do AODV. Tal situação ocorre, porque no RHBS o número de mensagens periódicas acumulam-se ao decorrer do tempo.

Figura 6.2 – Média Geral da Quantidade de Mensagens para cada Comunicação

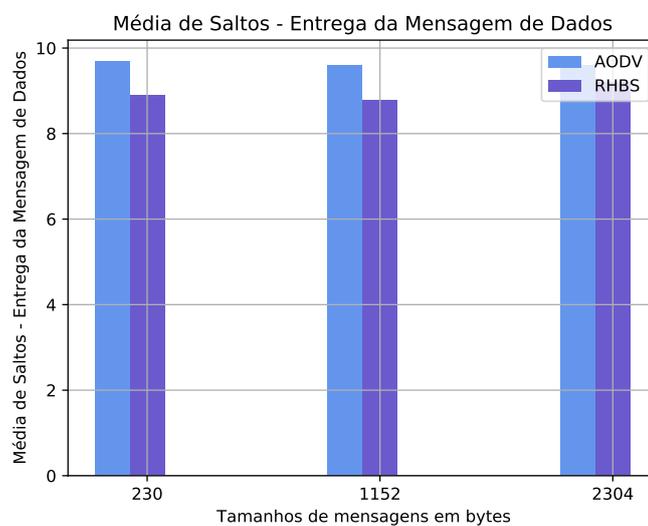


Fonte: autor (2020)

### Média de Saltos Utilizados na Entrega da Mensagem de Dados

Na Figura 6.3 é avaliado a quantidade média de saltos utilizados somente para a entrega da mensagem de dados. Nesta figura, pode ser notado que o RHBS e o AODV, ao serem comparados, não apresentaram grande diferenças para os três tamanhos de mensagem testados. Além disso, cada um dos protocolos manteve também um número médio de saltos muito próximo para cada um dos três tamanhos de mensagem testados.

Figura 6.3 – Média de Saltos - Entrega da Mensagem de Dados



Fonte: autor (2020)

## 6.2 RHBS

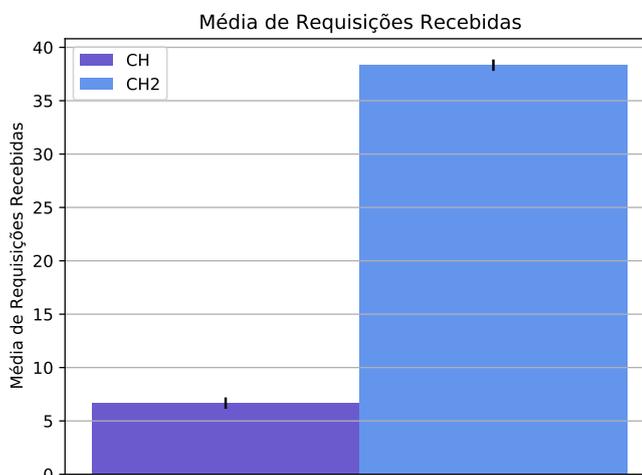
### 6.2.1 Etapa de Requisição de Posição

Na etapa de requisição de posição é avaliado o período que compreende a descoberta do nó de encaminhamento e o recebimento da mensagem de resposta do CH ou CH2. Como o número de requisições foram próximos nos testes para os três tamanhos de mensagens testados, neste tópico as médias dos resultados apresentadas são a média dos resultados da requisição de posição dos três tamanhos de mensagem.

#### Média de Requisições Recebidas

A Figura 6.4, demonstra a média de requisições recebidas pelos CHs e CH2. Durante as simulações dos três tamanhos de mensagens, a maioria dos veículos móveis de destino se encontravam em um setor diferente do setor do veículo móvel de origem, o que resultou em um maior número de requisições de posição reencaminhadas dos CHs para o CH2.

Figura 6.4 – Média de Requisições Recebidas



Fonte: autor (2020)

#### Latência Média, Média de Saltos e Quantidade Média de Mensagens

Como visto no capítulo 4, quando um veículo móvel de origem deseja enviar uma mensagem de dados para um veículo móvel de destino, ele deve primeiro descobrir a posição deste nó. Para isto, uma mensagem de requisição de posição é enviada para o CH do seu setor. Caso

este CH saiba ele enviará para o veículo de origem uma mensagem que possui a posição requisitada. Para uma melhor organização podemos chamar de Requisição Nível 1, o período gasto desde a requisição até o recebimento da posição ao utilizar somente o CH. Porém, caso o CH não saiba a posição ele reencaminha a requisição para o CH2 que então enviará uma mensagem para o veículo de origem contendo a posição requisitada. O período gasto desde a requisição até o recebimento da posição ao utilizar o CH2 é a Requisição Nível 2.

Na Figura 6.5, a latência média das requisições que utilizaram o CH2 é maior porque na Requisição Nível 2 um maior número de saltos (FIGURA 6.6) foi gasto. Isto se deve ao reencaminhamento da mensagem de requisição do CH para o CH2 e do envio da mensagem com a posição requisitada do CH2 para o veículo móvel de origem. Devido a disposição dos nós, quanto mais distante está um CH de um CH2 e quanto mais distante está o CH2 do veículo móvel de origem, mais saltos serão gastos e por consequência se terá um maior tempo de latência.

Ao ser observado também na Figura 6.5, a latência média das requisições que utilizaram somente o CH é menor porque na Requisição Nível 1, o próprio CH que recebeu a requisição responde para o veículo móvel de origem, o que diminui o número de saltos gastos e com isto a latência.

Na Figura 6.7, é avaliado o número de mensagens não periódicas gastas durante o período de requisição de posição. Pode-se perceber que o número de mensagens gastas igual ao número de saltos (FIGURA 6.6) devido a ser utilizado o roteamento geográfico guloso.

Figura 6.5 – Latência Média no Envio de Mensagens, com o Uso do CH2 e Sem o Uso do CH2

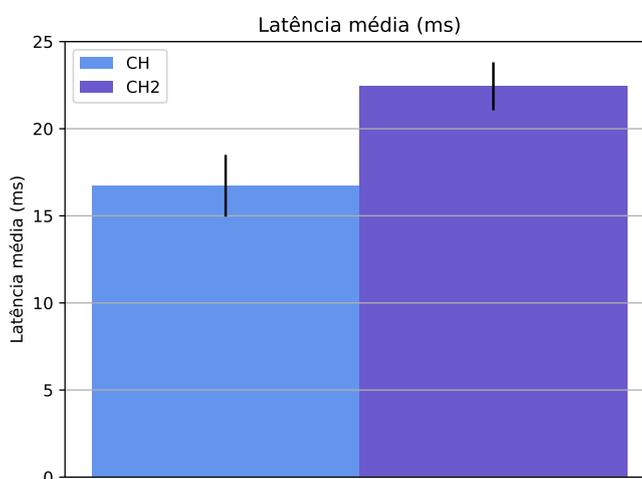
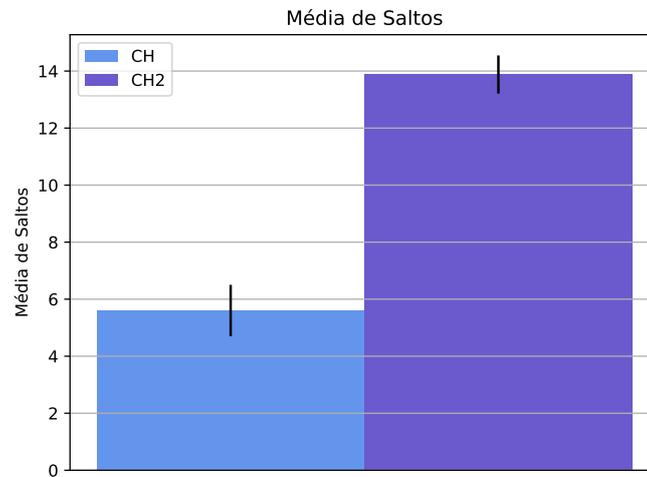
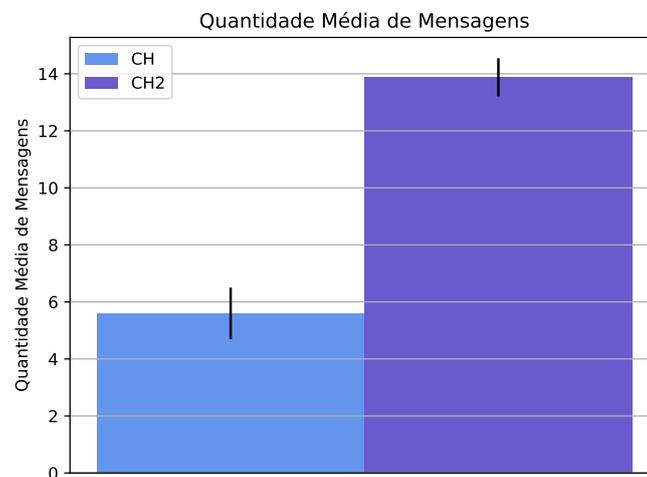


Figura 6.6 – Média de Saltos



Fonte: autor (2020)

Figura 6.7 – Quantidade Média de Mensagens



Fonte: autor (2020)

### 6.2.2 Etapa de Entrega da Mensagem de Dados

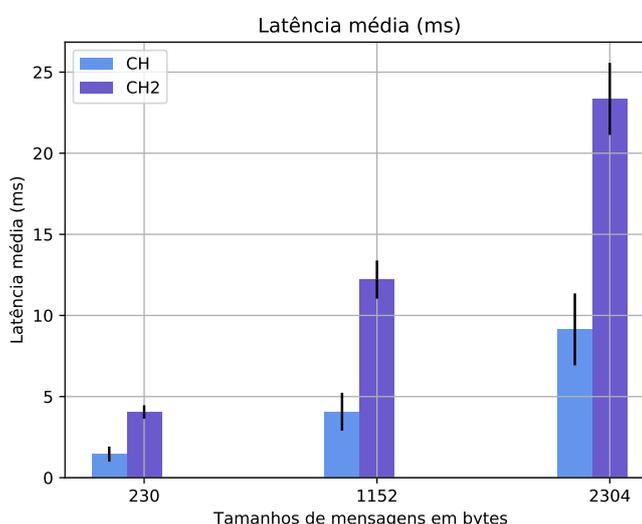
Nesta etapa é avaliado o período entre o recebimento da posição do veículo móvel de destino e o recebimento da mensagem de dados.

### Latência Média, Média de Saltos e Quantidade Média de Mensagens

Quando um veículo móvel de origem recebe a posição requisitada por meio de um CH ou CH2, ele logo em seguida deve enviar uma mensagem de dados para o veículo móvel de destino. Caso o veículo de origem e destino estejam no mesmo setor/cluster um número menor de saltos (FIGURA 6.9) será gasto, porém caso estejam em setores diferentes um número maior de saltos poderá ser gasto devido a uma maior distância entre eles. Porém, além do número de saltos na etapa de entrega de mensagens de dados, o tamanho da mensagem influencia no tempo de latência, pois quanto maior ela for, mais tempo será gasto para que ela seja transmitida pelo canal. Portanto na Figura 6.8, pode-se ver que o maior tempo de latência está relacionado ao tamanho de mensagem de 2304 bytes, tanto quando a posição requisitada é recebida do CH ou CH2.

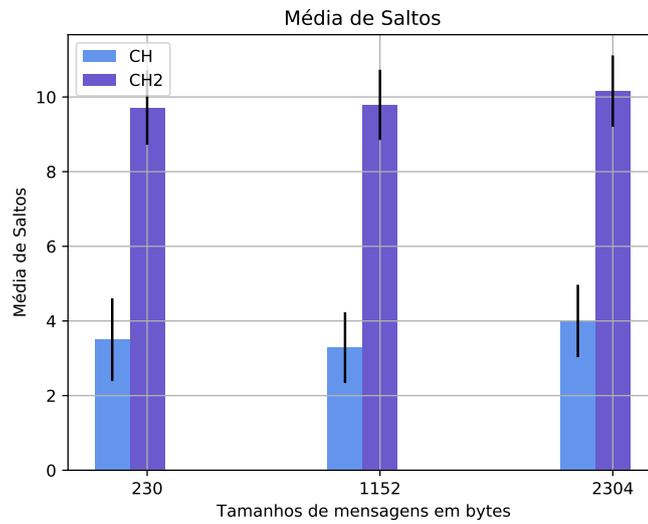
Nesta etapa também pode se observar que assim como na etapa de requisição de posição, o número de mensagens não periódicas (FIGURA 6.10) é proporcional ao número de saltos utilizados devido ao uso do roteamento geográfico guloso.

Figura 6.8 – Latência média no envio de mensagens, com o uso do CH2 e sem o uso do CH2



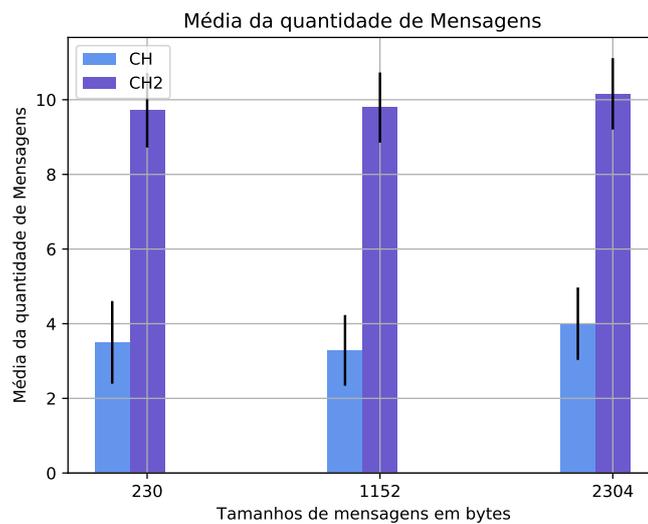
Fonte: autor (2020)

Figura 6.9 – Média de Saltos



Fonte: autor (2020)

Figura 6.10 – Média da Quantidade de Mensagens



Fonte: autor (2020)

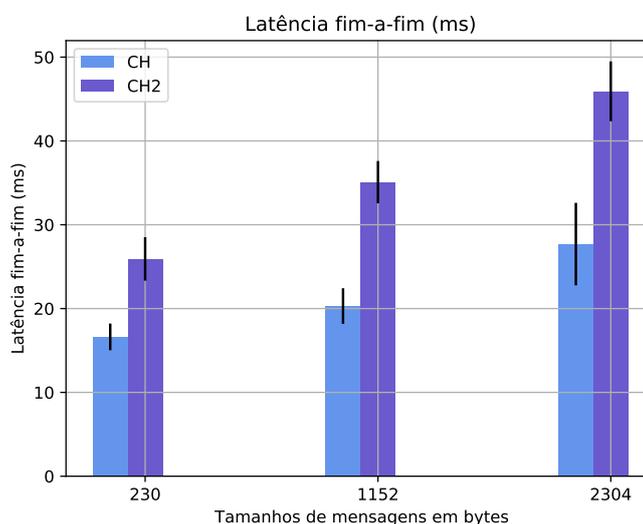
### 6.2.3 Avaliação Geral

Na avaliação geral é avaliado o período que corresponde a descoberta do nó de encaminhamento (etapa de requisição de posição) e a entrega da mensagem de dados (etapa de entrega da mensagem de dados).

## Latência Fim-a-fim

A Latência fim-a-fim (latência média geral) sofre a influência do número de saltos utilizados nas etapas de requisição de posição e envio de mensagem de dados. Além disso, como também já visto, na etapa de envio de mensagem de dados o tamanho da mensagem influencia no tempo de latência, o que não é diferente na latência média geral. Na Figura 6.11, um maior tempo de latência foi obtido quando mensagens de tamanho de 2304 bytes foram utilizadas. Este maior tempo de latência é perceptível tanto quando somente o CH é utilizado ou quando o CH2 também é utilizado.

Figura 6.11 – Latência Fim-a-fim

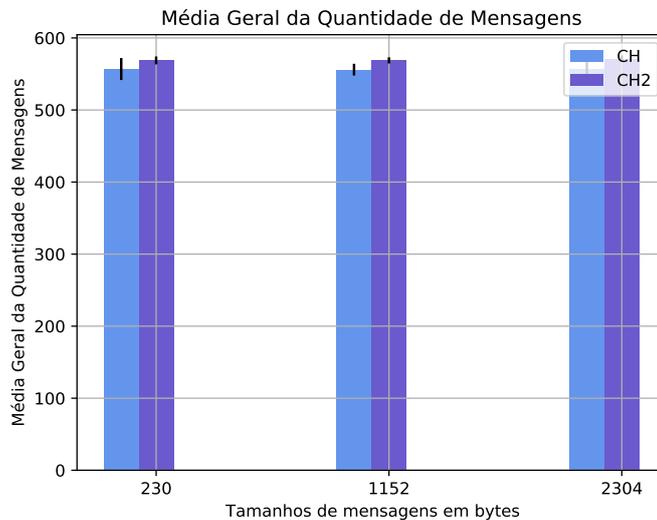


Fonte: autor (2020)

## Média Geral da Quantidade de Mensagens

A Figura 6.12, mostra a avaliação da média geral de quantidade de mensagens (mensagens periódicas e não periódicas) em 10 segundos de simulação. As médias obtidas foram parecidas, ao se utilizar somente o CH ou ao se utilizar também o CH2 na etapa de requisição de posição. Isto deve-se ao fato de que a quantidade de mensagens periódicas é a mesma nos dois casos.

Figura 6.12 – Média Geral da Quantidade de Mensagens

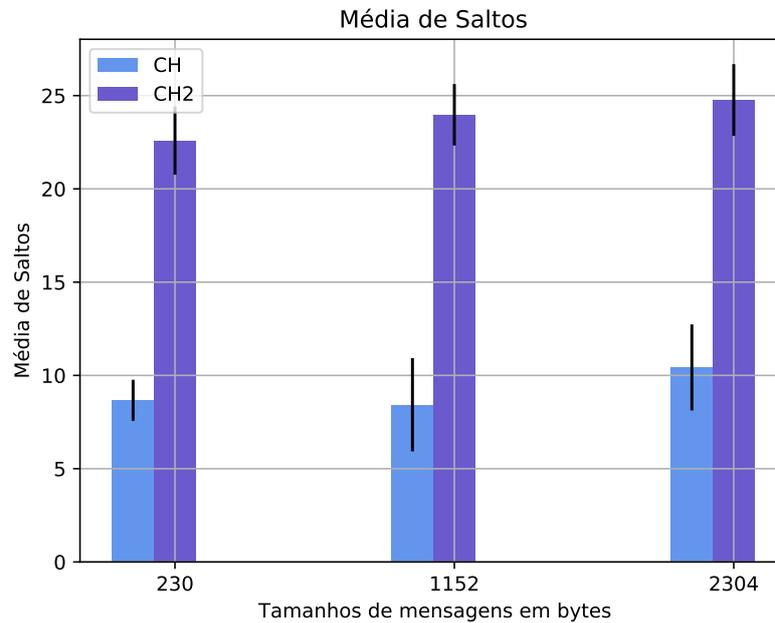


Fonte: autor (2020)

### Média de Saltos

Na Fig. 6.13, a média do total de saltos utilizados por mensagens não periódicas é avaliada. Esta média se manteve menor ao ser utilizado somente CHs. Quando somente o CH é usado, o veículo móvel de origem e o veículo móvel de destino estão no mesmo cluster. Portanto, menos saltos são gastos para requisitar a posição, para enviar a resposta da requisição e para enviar a mensagem de dados. Porém, ao ser observado a média de quantidade de saltos gastos para os três tamanhos de mensagem, percebe uma pequena diferença tanto ao utilizar o CH como ao utilizar o CH2. Esta diferença pode ocorrer por causa do sorteio aleatório de pares de veículos que se comunicaram entre si. Um veículo de origem pode estar mais distante de um CH, CH2 ou mesmo do veículo móvel de destino.

Figura 6.13 – Média de Saltos



Fonte: autor (2020)

### 6.3 Análise Final

Quando comparada a latência fim-a-fim dos protocolos AODV e RHBS, o RHBS apresentou um menor tempo de latência para os três tamanhos de mensagens testadas. Porém, após ser avaliada a média geral da quantidade de mensagens dos protocolos RHBS e AODV, o RHBS apresentou um desempenho superior quando existe um maior número de comunicações entre veículos que justifique as mensagens periódicas trocadas, já o AODV apresentou um desempenho superior quando a comunicação aconteceu de forma esporádica, na qual o número de mensagens de *flooding* não foi um problema.

Posteriormente, ao se avaliar a comunicação entre veículos somente no protocolo RHBS, percebe-se que o número de saltos utilizados por mensagens não periódicas é menor quando os veículos de origem e de destino estão em um mesmo setor. Embora, o número de saltos utilizados e o tamanho da mensagem transmitida influenciem no tempo de latência, a arquitetura hierárquica proposta se mostrou também eficiente na comunicação entre veículos de setores diferentes. O tempo de latência alcançado neste caso se mostrou também pequeno.

Ao serem realizados os testes e comparações descritos, conclui-se que o RHBS se qualifica como um protocolo de roteamento para redes veiculares. Ele possibilita uma comunicação rápida tanto entre veículos próximos como entre veículos distantes em um ambiente urbano.

## 7 CONCLUSÃO E TRABALHOS FUTUROS

Para possibilitar o roteamento entre veículos no ambiente de cidade, foi proposto o protocolo RHBS, algoritmo que ao fundir roteamento em cluster com roteamento geográfico, utiliza um backbone de carros estacionados para o encaminhamento de mensagens de controle e dados, além de uma hierarquia de CHs de dois níveis que fornecem as posições de veículos de destino que serão utilizadas durante o roteamento. O RHBS quando comparado com o protocolo AODV apresentou um menor tempo médio de latência e uma menor quantidade média de mensagens por comunicação em quatro das cinco taxas comparadas.

Em trabalhos futuros, pretende-se algumas melhorias e alguns novos testes para o RHBS. Relativo as melhorias, primeiro deseja-se alterar o algoritmo para que um veículo móvel de origem utilize um mesmo nó de encaminhamento descoberto por um período maior de tempo. Desta forma, diminui-se o número de mensagens utilizadas para descobrir nós deste tipo. Segundo, pretende-se definir um mecanismo de detecção de backbone. De acordo com esse mecanismo, um nó só envia uma mensagem quando o backbone está formado. Terceiro, criar uma estratégia em que CHs e CH2 enviem suas tabelas de membros para nós estacionados que irão substituí-los quando eles saírem do estacionamento. Relativo aos testes, primeiramente, é desejado para o trabalho que diferentes cenários que simulem construções e árvores que possam obstruir o sinal no ambiente urbano, sejam testados. Segundo, definir uma porcentagem de tempo que veículos devem permanecer estacionados e assim fazer rotacionamento de estacionamentos. Terceiro, pretende-se comparar o RHBS com protocolos de roteamento baseados em cluster e com protocolos de roteamento geográfico.

## REFERÊNCIAS

- AL-SULTAN, S. et al. A comprehensive survey on vehicular ad hoc network. **Journal of Network and Computer Applications**, v. 37, p. 380 – 392, 2014. ISSN 1084-8045. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S108480451300074X>>.
- AlMheiri, S. M.; AlQamzi, H. S. Manets and vanets clustering algorithms: A survey. In: **2015 IEEE 8th GCC Conference Exhibition**. [S.l.: s.n.], 2015. p. 1–6. ISSN null.
- ALOWISH, M. et al. Performance evaluation of a cluster based routing protocol for vanets. **Journal of Communications**, 2017. Disponível em: <[www.jocm.us/uploadfile/2017/0306/20170306060749561.pdf](http://www.jocm.us/uploadfile/2017/0306/20170306060749561.pdf)>.
- ALVES, R. S. et al. **Uma Análise Experimental da Capacidade de Redes Ad Hoc Veiculares**. 2008. XXVI SIMPÓSIO BRASILEIRO DE TELECOMUNICACÕES - SBrT'08. Disponível em: <[https://www.researchgate.net/profile/Marcelo\\_Rubinstein2/publication/267205608\\_Uma\\_Analise\\_Experimental\\_da\\_Capacidade\\_de\\_Redes\\_Ad\\_Hoc\\_Veiculares/links/55ae49c608aee079922086aa/Uma-Analise-Experimental-da-Capacidade-de-Redes-Ad-Hoc-Veiculares.pdf](https://www.researchgate.net/profile/Marcelo_Rubinstein2/publication/267205608_Uma_Analise_Experimental_da_Capacidade_de_Redes_Ad_Hoc_Veiculares/links/55ae49c608aee079922086aa/Uma-Analise-Experimental-da-Capacidade-de-Redes-Ad-Hoc-Veiculares.pdf)>.
- ALVES, R. S. et al. **Redes Veiculares: Princípios, Aplicações e Desafios**. 2009. 27º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Disponível em: <<https://www.gta.ufrj.br/ftp/gta/TechReports/ACC09.pdf>>.
- BALI, R. S.; KUMAR, N.; RODRIGUES, J. J. Clustering in vehicular ad hoc networks: Taxonomy, challenges and solutions. **Vehicular Communications**, v. 1, n. 3, p. 134 – 152, 2014. ISSN 2214-2096. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2214209614000217>>.
- Bengag, A.; El Boukhari, M. Classification and comparison of routing protocols in vanets. In: **2018 International Conference on Intelligent Systems and Computer Vision (ISCV)**. [S.l.: s.n.], 2018. p. 1–8. ISSN null.
- Bononi, L.; di Felice, M. A cross layered mac and clustering scheme for efficient broadcast in vanets. In: **2007 IEEE International Conference on Mobile Adhoc and Sensor Systems**. [S.l.: s.n.], 2007. p. 1–8. ISSN 2155-6814.
- BOUSSOUFA-LAHLAH, S.; SEMCHEDINE, F.; BOUALLOUCHE-MEDJKOUNE, L. Geographic routing protocols for vehicular ad hoc networks (vanets): A survey. **Vehicular Communications**, v. 11, p. 20 – 31, 2018. ISSN 2214-2096. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2214209616300183>>.
- Cadger, F. et al. A survey of geographical routing in wireless ad-hoc networks. **IEEE Communications Surveys Tutorials**, v. 15, n. 2, p. 621–653, 2013.
- COOPER, C. et al. A comparative survey of vanet clustering techniques. **IEEE Communications Surveys Tutorials**, v. 19, n. 1, p. 657–681, Firstquarter 2017. ISSN 1553-877X.
- Devangavi, A. D.; Gupta, R. Routing protocols in vanet — a survey. In: **2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)**. [S.l.: s.n.], 2017. p. 163–167. ISSN null.

- DUA, A.; KUMAR, N.; BAWA, S. A systematic review on routing protocols for vehicular ad hoc networks. **Vehicular Communications**, v. 1, n. 1, p. 33 – 52, 2014. ISSN 2214-2096. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2214209614000059>>.
- EZE, E. C.; ZHANG, S.; LIU, E. Vehicular ad hoc networks (vanets): Current state, challenges, potentials and way forward. In: **2014 20th International Conference on Automation and Computing**. [S.l.: s.n.], 2014. p. 176–181.
- Five G Americas. **V2X Cellular Solutions**. [S.l.], 2016. Disponível em: <[http://www.5gamericas.org/files/2914/7769/1296/5GA\\_V2X\\_Report\\_FINAL\\_for\\_upload.pdf](http://www.5gamericas.org/files/2914/7769/1296/5GA_V2X_Report_FINAL_for_upload.pdf)>.
- Five G Americas. **Cellular V2X Communications Towards 5G**. [S.l.], 2018. Disponível em: <[http://www.5gamericas.org/files/9615/2096/4441/2018\\_5G\\_Americas\\_White\\_Paper\\_Cellular\\_V2X\\_Communications\\_Towards\\_5G\\_Final\\_for\\_Distribution.pdf](http://www.5gamericas.org/files/9615/2096/4441/2018_5G_Americas_White_Paper_Cellular_V2X_Communications_Towards_5G_Final_for_Distribution.pdf)>.
- FONSECA, E.; FESTAG, A. A survey of existing approaches for secure ad hoc routing and their applicability to vanets. 01 2006.
- Goonewardene, R. T.; Ali, F. H.; Stipidis, E. Robust mobility adaptive clustering scheme with support for geographic routing for vehicular ad hoc networks. **IET Intelligent Transport Systems**, v. 3, n. 2, p. 148–158, June 2009. ISSN 1751-9578.
- HANDE, R. S.; MUDDANA, A. Comprehensive survey on clustering-based efficient data dissemination algorithms for vanet. In: **2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPE5)**. [S.l.: s.n.], 2016. p. 629–632.
- Hassan, A.; Ahmed, M. H.; Rahman, M. A. Performance evaluation for multicast transmissions in vanet. In: **2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE)**. [S.l.: s.n.], 2011. p. 001105–001108. ISSN 0840-7789.
- HOSSAIN, E. et al. Vehicular telematics over heterogeneous wireless networks: A survey. **Computer Communications**, v. 33, n. 7, p. 775 – 793, 2010. ISSN 0140-3664. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0140366410000022>>.
- IEEE Guide for Wireless Access in Vehicular Environments (WAVE) Architecture. **IEEE Std 1609.0-2019 (Revision of IEEE Std 1609.0-2013)**, p. 1–106, 2019.
- KARAGIANNIS, G. et al. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. **IEEE Communications Surveys Tutorials**, v. 13, n. 4, p. 584–616, Fourth 2011. ISSN 1553-877X.
- KARP, B.; KUNG, H. T. Gpsr: Greedy perimeter stateless routing for wireless networks. In: **Proceedings of the 6th Annual International Conference on Mobile Computing and Networking**. New York, NY, USA: ACM, 2000. (MobiCom '00), p. 243–254. ISBN 1-58113-197-6. Disponível em: <<http://doi.acm.org/10.1145/345910.345953>>.
- KRAJZEWICZ, D. et al. Recent development and applications of SUMO - Simulation of Urban MObility. **International Journal On Advances in Systems and Measurements**, v. 5, n. 3&4, p. 128–138, December 2012. Disponível em: <<http://elib.dlr.de/80483/>>.
- Lessmann, J.; Heimfarth, T.; Janacik, P. Shox: An easy to use simulation platform for wireless networks. In: **Tenth International Conference on Computer Modeling and Simulation (uksim 2008)**. [S.l.: s.n.], 2008. p. 410–415.

Li, F.; Wang, Y. Routing in vehicular ad hoc networks: A survey. **IEEE Vehicular Technology Magazine**, v. 2, n. 2, p. 12–22, June 2007. ISSN 1556-6080.

LIANG, W. et al. Vehicular ad hoc networks: Architectures, research issues, methodologies, challenges, and trends. **Int. J. Distrib. Sen. Netw.**, Taylor & Francis, Inc., Bristol, PA, USA, v. 2015, p. 17:17–17:17, jan. 2015. ISSN 1550-1329. Disponível em: <<https://doi.org/10.1155/2015/745303>>.

Lopez, P. A. et al. Microscopic traffic simulation using sumo. In: **2018 21st International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2018. p. 2575–2582.

Luo, Y.; Zhang, W.; Hu, Y. A new cluster based routing protocol for vanet. In: **2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing**. [S.l.: s.n.], 2010. v. 1, p. 176–180. ISSN null.

MEJRI, M. N.; BEN-OTHMAN, J.; HAMDI, M. Survey on vanet security challenges and possible cryptographic solutions. **Vehicular Communications**, v. 1, n. 2, p. 53 – 66, 2014. ISSN 2214-2096. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2214209614000187>>.

NI, S.-Y. et al. The broadcast storm problem in a mobile ad hoc network. In: **Fifth Annual International Conference on Mobile Computing and Networking (MobiCom'99)**. [s.n.], 1999. Disponível em: <<https://people.eecs.berkeley.edu/~culler/cs294-f03/papers/bcast-storm.pdf>>.

OMNET++. **What is OMNeT++?** 2020. Disponível em: <<https://omnetpp.org/intro/>>.

OpenStreetMap contributors. **Planet dump retrieved from <https://planet.osm.org>** . 2019. <<https://www.openstreetmap.org>>.

PAPADIMITRATOS, P. et al. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. **IEEE Communications Magazine**, v. 47, n. 11, p. 84–95, November 2009. ISSN 0163-6804.

Perkins, C. E.; Royer, E. M. Ad-hoc on-demand distance vector routing. In: **Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications**. [S.l.: s.n.], 1999. p. 90–100.

RASOOL, I. U.; ZIKRIA, Y. B.; KIM, S. W. A review of wireless access vehicular environment multichannel operational medium access control protocols: Quality-of-service analysis and other related issues. **International Journal of Distributed Sensor Networks**, 2017. Disponível em: <<https://doi.org/10.1177/1550147717710174>>.

RAWASHDEH, Z. Y.; MAHMUD, S. M. A novel algorithm to form stable clusters in vehicular ad hoc networks on highways. **EURASIP Journal on Wireless Communications and Networking**, v. 2012, 12 2012.

Santos, R. A. et al. A location-based routing algorithm for vehicle to vehicle communication. In: **Proceedings. 13th International Conference on Computer Communications and Networks (IEEE Cat. No.04EX969)**. [S.l.: s.n.], 2004. p. 221–226. ISSN 1095-2055.

SANTOS, R. A.; EDWARDS, R. M.; SEED, N. L. Using the cluster-based location routing (cblr) algorithm for exchanging information on a motorway. In: **4th International Workshop on Mobile and Wireless Communications Network**. [S.l.: s.n.], 2002. p. 212–216.

SENOUCI, O.; ZIBOUDA, A.; HAROUS, S. Survey: Routing protocols in vehicular ad hoc networks. In: **Proceedings of the Second International Conference on Advanced Wireless Information, Data, and Communication Technologies**. New York, NY, USA: Association for Computing Machinery, 2017. (AWICT 2017). ISBN 9781450353106. Disponível em: <<https://doi.org/10.1145/3231830.3231838>>.

SHAREF, B. T.; ALSAQOUR, R. A.; ISMAIL, M. Vehicular communication ad hoc routing protocols: A survey. **Journal of Network and Computer Applications**, v. 40, p. 363 – 396, 2014. ISSN 1084-8045. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1084804513001963>>.

SOMMER, C.; GERMAN, R.; DRESSLER, F. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. **IEEE Transactions on Mobile Computing**, IEEE, v. 10, n. 1, p. 3–15, January 2011.

SUCASAS, V. et al. A survey on clustering techniques for cooperative wireless networks. **Ad Hoc Networks**, v. 47, p. 53 – 81, 2016. ISSN 1570-8705. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1570870516301123>>.

Tao Song et al. A cluster-based directional routing protocol in vanet. In: **2010 IEEE 12th International Conference on Communication Technology**. [S.l.: s.n.], 2010. p. 1172–1175. ISSN null.

TIAN, D. et al. A vanets routing algorithm based on euclidean distance clustering. In: **2010 2nd International Conference on Future Computer and Communication**. [S.l.: s.n.], 2010. v. 1, p. V1–183–V1–187.

Tiecheng Wang; Gang Wang. Tibcrph: Traffic infrastructure based cluster routing protocol with handoff in vanet. In: **The 19th Annual Wireless and Optical Communications Conference (WOCC 2010)**. [S.l.: s.n.], 2010. p. 1–5. ISSN 2379-1276.

Ting-Chao Hou; Tzu-Jane Tsai. A access-based clustering protocol for multihop wireless ad hoc networks. **IEEE Journal on Selected Areas in Communications**, v. 19, n. 7, p. 1201–1210, July 2001. ISSN 1558-0008.

UZCÁTEGUI, R. A.; ACOSTA-MARUM, G. Wave: A tutorial. **IEEE Communications Magazine**, IEEE Press, Piscataway, NJ, USA, v. 47, n. 5, p. 126–133, maio 2009. ISSN 0163-6804. Disponível em: <<https://doi.org/10.1109/MCOM.2009.4939288>>.

VARGA, A. The omnet++ discrete event simulation system. In: **In ESM'01**. [S.l.: s.n.], 2001.

VARGA, A. **OMNeT++ Simulation Manual**. [S.l.], 2016. Disponível em: <<https://doc.omnetpp.org/omnetpp/manual/>>.

VEGNI, A. M.; LITTLE, T. Hybrid vehicular communications based on v2v-v2i protocol switching. v. 2, p. 213 – 231, 12 2011. Disponível em: <<https://www.inderscienceonline.com/doi/abs/10.1504/IJVICS.2011.044263>>.

VEINS. **Documentation**. 2020. Disponível em: <<https://veins.car2x.org/documentation/>>.

VODOPIVEC, S.; BEŠTER, J.; KOS, A. A survey on clustering algorithms for vehicular ad-hoc networks. In: **2012 35th International Conference on Telecommunications and Signal Processing (TSP)**. [S.l.: s.n.], 2012. p. 52–56.

WANGHAM, M. S. et al. Segurança em redes veiculares: Inovações e direções futuras. In: . [s.n.], 2014. Minicursos do XIV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais — SBSeg 2014. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbseg/2014/0037.pdf>>.

WISCHHOF, L.; EBNER, A.; ROHLING, H. Information dissemination in self-organizing intervehicle networks. **IEEE Transactions on Intelligent Transportation Systems**, v. 6, n. 1, p. 90–101, March 2005. ISSN 1524-9050.

Xia, Y.; Yeo, C. K.; Lee, B. S. Hierarchical cluster based routing for highly mobile heterogeneous manet. In: **2009 International Conference on Network and Service Security**. [S.l.: s.n.], 2009. p. 1–6. ISSN null.

YOUSEFI, S.; MOUSAVI, M. S.; FATHY, M. Vehicular ad hoc networks (vanets): Challenges and perspectives. In: **2006 6th International Conference on ITS Telecommunications**. [S.l.: s.n.], 2006. p. 761–766.

ZEADALLY, S. et al. Vehicular ad hoc networks (vanets): status, results, and challenges. **Telecommunication Systems**, v. 50, n. 4, p. 217–241, Aug 2012. ISSN 1572-9451. Disponível em: <<https://doi.org/10.1007/s11235-010-9400-5>>.