



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

O USO DE FILTROS DE BLOOM EM UM PROTOCOLO BASEADO NA MÉTRICA DE  
OU-EXCLUSIVO EM REDES VEICULARES URBANAS

Ederval Pablo Ferreira da Cruz

**Orientador**

Prof. Dr. Carlos Alberto Vieira Campos

**Co-Orientador**

Prof. Dr. Rafael Pasquini

RIO DE JANEIRO, RJ – BRASIL  
JULHO DE 2012

O USO DE FILTROS DE BLOOM EM UM PROTOCOLO BASEADO NA MÉTRICA DE  
OU-EXCLUSIVO EM REDES VEICULARES URBANAS

Ederval Pablo Ferreira da Cruz

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO  
DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-GRADUAÇÃO EM  
INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO  
(UNIRIO), APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

Aprovada por:

---

Carlos Alberto Vieira Campos, D. Sc – UNIRIO

---

Rafael Pasquini, D. Sc – UFU

---

Morganna Diniz, D. Sc – UNIRIO

---

Ronaldo Moreira Salles, Ph.D. – IME

RIO DE JANEIRO, RJ – BRASIL  
JULHO DE 2012

C957 Cruz, Ederval Pablo Ferreira da.  
O uso de filtros de Bloom em protocolo baseado na métrica de ou-exclusivo em redes veiculares urbanas / Ederval Pablo Ferreira da Cruz, 2012. 96f.

Orientador: Carlos Alberto Vieira Campos.

Coorientador: Rafael Pasquini.

Dissertação (Mestrado em Informática) – Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2012.

1. Roteamento (Administração de redes de computadores). 2. VANET (Veicular Ad Hoc Networks). 3. Filtros de Bloom. 4. Xor. 5. Algoritmos de computador. I. Campos, Carlos Alberto Vieira. II. Pasquini, Rafael. III. Universidade Federal do Estado do Rio de Janeiro. Centro de Ciências Exatas e Tecnologia. Curso de Mestrado em em Informática. IV. Título.

CDD – 004.62

Ao meu pai Ivan e minha mãe Larrúbia,  
que são as pessoas mais importantes na minha vida  
e que sempre me incentivaram a seguir  
em frente em meus projetos de vida.

## Agradecimentos

ÀQUELE que é responsável por tudo que existe neste mundo e que me deu forças para conseguir finalizar este trabalho e concluir mais uma etapa da minha vida.

À Thássia, minha amada, companheira e amiga, pela paciência e compreensão dos momentos em que não pude estar presente e dar todo o carinho que era merecido.

Ao meu irmão Adam que apesar da dor de cabeça que às vezes me dá, as suas ligações nos momentos de saudade foram altamente reflitantes.

Ao meu orientador Prof. Dr. Carlos Alberto Vieira Campos, que na verdade se tornou nesses dois anos de convivência, um amigo. Acho que orientador não é a melhor palavra para definir sua função no mestrado, mas sim, conselheiro, que soube compreender as minhas limitações e me ajudou a superá-las, além de várias outras dicas valiosas que espero compartilhar com os meus alunos no meu local de trabalho e também segui-las mais à risca!!!! Assim como o grande parceiro, meu co-orientador, Prof. Dr. Rafael Pasquini, da Universidade Federal de Uberlândia, pela oportunidade e ajuda na realização deste trabalho e que nunca faltou com o apoio quando precisado e com dicas preciosas.

Agradeço também ao Prof. Rodolfo Oliveira, da Universidade Nova de Lisboa que contribuiu, e muito, com dicas e sugestões para a realização deste trabalho.

Aos professores Sidney, Vânia, Flávia e Morganna que fizeram o seu trabalho com maestria, e que nos momentos difíceis também deram seus “pitacos” para não desistirmos no caminhar dessa jornada.

Aos meus colegas de mestrado, em especial aos amigos Carlos Nilton, Elenilson e Daniel que compartilharam da minha angústia e foram companheiros em ouvir minhas lamentações e desabafos nos momentos difíceis no decorrer do mestrado. Não posso esquecer, em hipótese alguma, do colega Bruno Silva que na época dessa dissertação era graduando, mas que já tinha cabeça de mestrando e que foi um parceiro imensurável na realização deste trabalho. Valeu Bruno pela força e parceria na implementação do código e todo o sucesso do mundo na sua vida pessoal/profissional e, de repente, inicia aqui uma parceria para o seu mestrado. Conte sempre comigo meu amigo!!!!

A minha Tia Noêmia, meus primos Wallace e Thainá que me suportaram durante 2 anos em seu lar, e que de fato, sem suas companhias e momentos de

alegria, talvez o trabalho seria menos lúdico. Assim digo também dos amigos Fábio, Priscila e Estela que foram companheiros nestes quase 2 anos de vivência no Rio.

Não posso esquecer da minha prima Mariana que segurou uma barra e tanto enquanto estive no Rio de Janeiro na realização deste trabalho. Obrigado de coração!!!

Aos colegas do IFES (Instituto Federal do Espírito Santo) que tanto me incentivaram a fazer o mestrado, para me capacitar e me tornar um melhor profissional beneficiando os meus futuros alunos com o meu conhecimento adquirido.

CRUZ, Ederval Pablo Ferreira da. O Uso de filtros de Bloom em um Protocolo Baseado na Métrica XOR em Redes Veiculares Urbanas. UNIRIO, 2012. Xxx páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

## RESUMO

Em redes *ad hoc* veiculares, mais conhecidas como VANETs, a alta mobilidade dos veículos faz com que os protocolos de roteamento existentes para as redes *ad hoc* móveis tenham a necessidade de características que lidem com este ambiente dinâmico, justificando assim, a necessidade de projetar novos protocolos.

Nesta dissertação é realizado um estudo de como se comporta o protocolo baseado na métrica OU-Exclusivo (XOR), utilizando filtros de Bloom na sua estrutura de envio de mensagens, em um cenário urbano. Primeiramente é proposta uma alteração no protocolo baseado na métrica XOR, com alterações incorporando os filtros de Bloom, surgindo assim o protocolo XORBF, de forma a lidar melhor com a dinamicidade da rede, melhorando o desempenho. Em seguida são realizadas comparações de desempenho do protocolo com outros protocolos de roteamento baseados na topologia e baseados na localização, analisando a taxa de caminhos criados, o atraso na entrega dos pacotes, o comprimento médio dos caminhos (em número de saltos) e a duração dos caminhos. Os resultados mostram que a utilização dos filtros de Bloom impactou positivamente na redução do atraso quando comparado com outro protocolo da família XOR, assim como impactou positivamente no comprimento dos caminhos, superando protocolos baseados em topologia e em localização. Sendo assim, este trabalho mostra que a introdução dos filtros de Bloom na estrutura de sinalização do protocolo de roteamento baseado na métrica XOR contribui para a melhoria no desempenho das VANETs que utilizam a solução proposta. Em resumo, o uso de filtros de Bloom evita que informações de vizinhança desnecessárias sejam trocadas entre os nós da rede, contribuindo para a convergência das tabelas de rotas.

**Palavras-chave:** Redes *ad hoc* veiculares, VANET, protocolos de roteamento baseados na métrica XOR, VANET urbana, filtros de Bloom



## ABSTRACT

In vehicular *ad hoc* networks (VANETs), the high mobility of vehicles challenges the existing routing protocols for the mobile *ad hoc* networks, requiring that they have new characteristics in order to deal with this dynamic environment, so, it is necessary to design new routing protocols capable of dealing with such scenario.

This dissertation presents a study where the behavior of a routing protocol, based on the XOR metric is verified, extending its signaling mechanism to support the use of *Bloom filters*, considering an urban scenario. First of all, it is proposed an extension in the routing protocol based on the XOR metric, including *Bloom filters* in its structure, originating in this way the XORBF protocol. Basically, the objective is to offer a mechanism capable of better dealing with the dynamicity of the network, improving the performance of the overall system. In the sequence, experiments are performed to compare the performance of the proposed XORBF protocol with other topology-based and geographic-based protocols, in terms of path availability, packet delivery delay, path length (in number of hops) and duration of the paths. The results show that using the *Bloom filters* technique, there is an important contribution, reducing the values of the delay when the XORBF protocol is compared with another XOR protocol, at the same time, there is a gain regarding the path length metric, overcoming others topology and geographic based protocols of the literature. This work shows that the proposed routing mechanism based on the XOR metric and *Bloom filters*, provides an interesting alternative for VANETs, contributing for the convergence of the routing tables. In essence, the use of *Bloom filters* avoids unnecessary neighborhood signaling exchange, improving the VANET performance.

**Keywords:** Vehicular *ad hoc* networks, VANET, routing protocols based on the XOR metric, urban VANET, *Bloom filter*

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação	3
1.2	Objetivos e contribuições	5
1.3	Estrutura da dissertação	6
<b>2</b>	<b>Redes <i>ad hoc</i> Veiculares</b>	<b>8</b>
2.1	Arquitetura das VANETs	9
2.2	Características das VANETs	11
2.3	Cenários de Mobilidade em VANETs	12
2.3.1	Mobilidade em Rodovias	13
2.3.2	Mobilidade em Ambientes Urbanos	14
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>17</b>
3.1	Roteamento baseado na Topologia	18
3.1.1	AODV	20
3.1.2	DSR	21
3.1.3	OLSR	24
3.2	Roteamento Geográfico	26
3.2.1	GPSR	27
3.2.2	GSR	31
3.2.3	A-STAR	32
3.2.4	GYTAR	33
<b>4</b>	<b>Roteamento baseado em OU-Exclusivo</b>	<b>35</b>
4.1	Princípio de Roteamento XOR	36
4.2	Processo de Descoberta	39
4.3	Processo de Roteamento	40
4.4	Processo de Aprendizagem	42
4.5	Conceito de Visibilidade Local	43
4.6	Considerações Finais	45

<b>5</b>	<b>Protocolo Proposto</b> .....	<b>47</b>
5.1	Filtros de Bloom .....	48
5.2	Cálculo da taxa de Falsos Positivos .....	50
5.3	Utilizando filtros de Bloom no protocolo baseado em XOR.....	51
5.4	A mensagem <i>QUERY</i> .....	53
5.5	A mensagem <i>RESPONSE</i> .....	55
<b>6</b>	<b>Análise de Desempenho</b> .....	<b>58</b>
6.1	Cenário de Simulação .....	59
6.1.1	Métricas de Avaliação .....	63
6.1.2	Resultados das Simulações .....	64
6.1.3	Discussão dos Resultados .....	90
<b>7</b>	<b>Conclusões</b> .....	<b>93</b>
7.1	Discussões Finais .....	93
7.2	Trabalhos Futuros .....	95
	<b>Referências Bibliográficas</b> .....	<b>96</b>

# Lista de Tabelas

Tabela 4.1 – Tabela de roteamento do nó 11 .....	37
Tabela 6.1 – Características da simulação .....	61

# Lista de Figuras

Figura 2.1 – Arquitetura das VANETs .....	10
Figura 2.2 – Rodovia onde os carros são representados pela cor azul e os caminhões pela cor vermelha. (a) Rodovia unidirecional com 3 faixas; (b) rodovia bidirecional com 4 faixas em cada direção .....	13
Figura 2.3 – Mapa de uma região da cidade de Nova York representando um cenário urbano .....	15
Figura 2.4 – Aproximação do mapa da Figura 2.3, mostrando diversas características existentes nos cenários urbanos .....	15
Figura 3.1 – Taxonomia de vários protocolos de roteamento utilizados em VANETs.....	18
Figura 3.2 – Exemplo de descoberta de rotas do protocolos DSR .....	22
Figura 3.3 – Tipos de inundação: inundação pura e inundação por MPRs....	25
Figura 3.4 – Situação de máximo local do protocolo GPSR .....	28
Figura 3.5 – Exemplo de um grafo de vizinhança relativa .....	30
Figura 4.1 – Processo de envio de mensagens QUERY no protocolo XOR .....	39
Figura 4.2 – Exemplo de encaminhamento do protocolo XOR.....	42
Figura 4.3 – Exemplo do processo de aprendizagem .....	43
Figura 4.4 – Mensagem <i>QUERY</i> limitada a 3 saltos .....	45
Figura 5.1 – Exemplo de adição e consulta de elementos usando filtro de Bloom .....	50
Figura 5.2 – Exemplo da utilização do filtro de Bloom na mensagem <i>QUERY</i> .....	52
Figura 5.3 – Processo de envio de mensagem de <i>QUERY</i> com o filtro de Bloom .....	55

Figura 6.1 – Cenário urbano utilizado na simulação .....	59
Figura 6.2 – Parte do mapa urbano utilizado na simulação mostrando características realísticas.....	60
Figura 6.3 – Taxa de caminhos disponíveis/criados pelos protocolos XOR (H=2), XORBF (H=2), GPSR, AODV e OLSR .....	65
Figura 6.4 – Taxa de caminhos disponíveis/criados pelos protocolos XOR (H=3), XORBF (H=3), GPSR, AODV e OLSR .....	65
Figura 6.5 – Taxa de caminhos disponíveis/criados pelos protocolos XOR (H=4), XORBF (H=4), GPSR, AODV e OLSR .....	66
Figura 6.6 – Taxa de caminhos disponíveis/criados pelos protocolos XOR (H=5), XORBF (H=5), GPSR, AODV e OLSR .....	66
Figura 6.7 – Atraso na entrega dos pacotes (H=2).....	69
Figura 6.8 – Atraso na entrega dos pacotes (H=2 com zoom em 10 ms.).....	70
Figura 6.9 – Atraso na entrega dos pacotes (H=3).....	70
Figura 6.10 – Atraso na entrega dos pacotes (H=3 com zoom em 20 ms.)...	71
Figura 6.11 – Atraso na entrega dos pacotes (H=4).....	71
Figura 6.12 – Atraso na entrega dos pacotes (H=4 com zoom em 10 ms.)...	72
Figura 6.13 – Atraso na entrega dos pacotes (H=5).....	72
Figura 6.14 – Atraso na entrega dos pacotes (H=5 com zoom em 10 ms.)...	73
Figura 6.15 – Comprimento dos caminhos criados pelos protocolos XOR (H=2), XORBF (H=2), AODV, OLSR e GPSR .....	74
Figura 6.16 – Comprimento dos caminhos criados pelos protocolos XOR (H=3), XORBF (H=3), AODV, OLSR e GPSR .....	75
Figura 6.17 – Comprimento dos caminhos criados pelos protocolos XOR (H=4), XORBF (H=4), AODV, OLSR e GPSR .....	75
Figura 6.18 – Comprimento dos caminhos criados pelos protocolos XOR (H=5), XORBF (H=5), AODV, OLSR e GPSR .....	76
Figura 6.19 – Tempo médio de duração dos caminhos criados pelos	

protocolos XOR (H=2), XORBF (H=2), AODV, OLSR e GPSR .....	77
Figura 6.20 – Tempo médio de duração dos caminhos criados pelos protocolos XOR (H=3), XORBF (H=3), AODV, OLSR e GPSR .....	77
Figura 6.21 – Tempo médio de duração dos caminhos criados pelos protocolos XOR (H=4), XORBF (H=4), AODV, OLSR e GPSR .....	78
Figura 6.22 – Tempo médio de duração dos caminhos criados pelos protocolos XOR (H=5), XORBF (H=5), AODV, OLSR e GPSR .....	78
Figura 6.23 – FDAE da duração dos caminhos criados pelos protocolos XOR e XORBF com H=2 a H=5 no cenário com 40 veículos .....	80
Figura 6.24 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 40 veículos .....	80
Figura 6.25 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com H=2 a H=5 no cenário com 80 veículos .....	81
Figura 6.26 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 80 veículos .....	82
Figura 6.27 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com H=2 a H=5 no cenário com 120 veículos .....	83
Figura 6.28 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 120 veículos .....	83
Figura 6.29 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com H=2 a H=5 no cenário com 160 veículos .....	84
Figura 6.30 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 160 veículos .....	85
Figura 6.31 – FDAE da duração dos caminhos dos protocolos	

XOR e XORBF com H=2 a H=5 no cenário com 200 veículos .....	86
Figura 6.32 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 200 veículos .....	
	86
Figura 6.33 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com H=2 a H=5 no cenário com 240 veículos .....	
	87
Figura 6.34 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 240 veículos .....	
	88
Figura 6.35 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com H=2 a H=5 no cenário com 280 veículos .....	
	89
Figura 6.36 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 280 veículos .....	
	89



# Glossário

AODV – Ad hoc On-Demand Distance Vector

A-STAR - Anchor-Based Street and Traffic Aware Routing

DSDV – Destination Sequence Distance Vector

DSR – Dynamic Source Routing

FDAE – Função Densidade Acumulada Empírica

GLONASS – Global Navigation Satellite System

GPS – Global Positioning System

GPSR – Greedy Perimeter Stateless Routing

GSR – Geographic Source Routing

GyTAR - Improved Greedy Traffic Aware Routing

Hz - Hertz

ITS – Intelligent Transportation System

OLSR – Optimized Link State Routing

MAC – Medium Access Control

MANET – Mobile Ad hoc Network

MID – Multiple Interface Declaration

MPR – Multipoint Relay

P2P – Peer-to-Peer

RREQ – Route Request

RREP – Route Reply

RERR – Route Error

RLS – Reactive Location Service

RVC – Roadside to Vehicle Communication

TC – Topology Control

V2I – Vehicular to Infrastructure

V2V – Vehicle to Vehicle

VANET – Vehicular Ad Hoc Network

XOR – OU-Exclusivo

# 1 Introdução

Nas últimas décadas, os veículos cada vez mais fazem parte da vida das pessoas, seja em rodovias ou nas ruas das cidades. Diversas novas ideias referentes ao aumento da segurança e aplicações que forneçam conforto foram propostas e, assim, o conceito tradicional de viajar de carro tende a ser modificado. Uma das tecnologias que têm atraído muito a atenção de pesquisadores ao redor do mundo e de empresas automobilísticas é a *Vehicular Ad hoc Networks* (VANETs), tecnologia esta que introduz o conceito de comunicação de dados entre veículos. Porém, existem trabalhos que também utilizam os termos IVC (*Inter-Vehicle Communications*) e V2V (*Vehicle-to-vehicle Communications*), mas como o termo VANET é amplamente adotado nos trabalhos da área, apesar de ainda não estar claro entre a comunidade científica qual termo deve ser adotado de forma definitiva, o termo VANET será utilizado no presente trabalho.

VANET é uma rede sem fios que é formada puramente entre os veículos, de forma a não precisar da presença de qualquer infraestrutura de comunicação como, por exemplo, a presença de pontos de acesso ou estações base. Neste contexto, para

que esses veículos possam trocar informações, é necessário que os mesmos estejam equipados com dispositivos que permitam tal comunicação, tais como emissores e receptores sem fios e módulos informatizados, de forma a conseguir a troca de informações entre os veículos.

As VANETs podem ser consideradas uma instanciação de *Mobile Ad Hoc Networks* (MANETs), porém com características especiais deste tipo de rede. As MANETs possuem várias características tais como [1]:

- **Topologia dinâmica** – Mudanças constantes na sua topologia devido à liberdade de mobilidade dos nós;
- **Banda limitada** – Equipamentos pequenos que possuem capacidade de transmissão limitada;
- **Energia limitada** – Os equipamentos em geral são alimentados por baterias que possuem capacidade de fornecimento de energia limitada.

Nas VANETs os nós também são móveis, porém por serem veículos, tendem a ter a sua mobilidade limitada às estradas ou vias urbanas pelas quais trafegam. Diferentes de outras redes móveis, nas VANETs os nós não possuem limitação quanto à energia devido ao fato de que os veículos possuem baterias de grande capacidade e longa duração.

No cenário das rodovias, em relação à comunicação entre os nós, existe com menor ocorrência a presença de obstáculos tais como prédios e árvores o que não impacta significativamente na propagação do sinal de rádio. Porém, as áreas urbanas possuem características como limitações de velocidade, semáforos, variações no número de faixas existentes, entre outros fatores que influenciam no desenvolvimento de soluções para VANETs e que tornam este tipo de cenário altamente desafiador e difícil de ser modelado.

Portanto, é possível vislumbrar que diversas aplicações poderão ser desenvolvidas para as VANETs, visando suprir as expectativas de consumidores,

empresas, governos, entre outros setores de segurança do trânsito e serviços de emergência.

Muitas das aplicações visam à melhoria da segurança no trânsito, como por exemplo, evitar acidentes. Um exemplo de aplicação deste tipo seria a rápida disseminação de mensagens na ocorrência de um acidente, fazendo com que os condutores reduzissem a velocidade dos veículos e, além disso, informando sobre outras rotas possíveis a serem utilizadas de forma a evitar a região onde o acidente ocorreu. Neste caso, benefícios incluem a melhora no fluxo dos veículos e, também, ganhos de agilidade no serviço de emergência, atendendo mais rapidamente os possíveis feridos. Outros tipos de aplicações podem ser desenvolvidos, com apelos diferentes, como por exemplo, jogos cooperativos, Internet, multimídia, etc. Contudo, a alta mobilidade dos nós existentes neste tipo de rede faz das VANETs uma tecnologia altamente desafiadora.

## **1.1 Motivação**

A mobilidade não é o único desafio existente nas VANETs. Outro fator que deve também ser levado em consideração nos protocolos de encaminhamento é a escalabilidade. Os protocolos de roteamento existentes podem ser classificados em: baseado na topologia e baseado na posição dos nós.

Os protocolos baseados na topologia, tais como o *Ad hoc On-Demand Distance Vector* (AODV) [2], *Dynamic Source Routing* (DSR) [3], *Optimized Link State Routing* (OLSR) [4] e outros protocolos da mesma classe, apresentam resultados frustrantes no ambiente das VANETs devido à alta mobilidade dos nós, porque tendem a ter pouca convergência de rotas e baixa taxa de comunicação [5,6]. Devido à necessidade de manter as rotas sempre atualizadas, tais protocolos comportam-se de forma com que haja um consumo excessivo por parte do tráfego de sinalização, o que se torna ainda mais preocupante com o aumento da densidade dos nós.

Os protocolos baseados na posição dos nós, também conhecidos como protocolos de roteamento geográfico, necessitam, também, ter conhecimento da posição dos nós na rede, logo, o tráfego de sinalização ocupa boa parte da banda passante, o que, novamente, torna-se preocupante com o aumento da densidade dos nós. Em geral, boa parte dos protocolos desta categoria utiliza de algum sistema de posicionamento, como, por exemplo, o GPS (*Global Positioning System*), mas, além disso, podem existir outros recursos que auxiliam no roteamento geográfico, tais como a utilização de mapas digitais.

Inicialmente, os trabalhos de avaliação de desempenho de protocolos nas VANETs eram realizados nos cenários de rodovias [7,8,9]. Porém, nos últimos anos, os cenários urbanos têm sido mais explorados pelos pesquisadores da área [10,11], por causa das características existentes nestes cenários que os tornam mais complexos, tais como: algumas regiões da cidade, em determinados horários, possuem características diversas, como diferentes densidades de nós, além das diferentes características das ruas, tais como: números de faixas, direção das vias, limitação de velocidade, presença de semáforos, presença de prédios e árvores que dificultam a propagação do sinal de rádio, entre outras. Desta forma, podemos dizer que os cenários urbanos são altamente desafiadores para os protocolos de roteamento nas VANETs.

Existem também os protocolos baseados na métrica de OU-Exclusivo (XOR), que em sua maioria foram propostos para redes infraestruturadas, tal como o Kademia [12]. Estes protocolos têm como principal diferencial o fato de não dependerem nem da localização dos nós, nem da topologia da rede, propiciando um cenário de investigação inovador em busca de soluções para os problemas de escalabilidade das VANETs. Nesta classe de protocolos, o encaminhamento dos pacotes é baseado na operação lógica de OU-Exclusivo (XOR), aplicada entre os identificadores dos nós, que permite assim, a criação de tabelas de roteamento compostas por apenas uma fração de todos os identificadores disponíveis na rede.

Assim, não é necessário ter tanta informação como nos protocolos baseados em topologia e em posição, portanto, utilizando uma menor quantidade de banda passante nas ações de convergência do sistema de roteamento.

Desta forma, considerando os cenários urbanos, onde a densidade de veículos tende a ser maior do que nas rodovias, por existirem poucos trabalhos disponíveis na literatura que analisam ou propõem novos protocolos de roteamento em cenários urbanos e que os cenários urbanos são essenciais no desenvolvimento da tecnologia VANET, decidiu-se efetuar as pesquisas apresentadas neste trabalho em tais cenário. Em resumo, todos os desafios relacionados às VANETs nos cenários urbanos motivam o trabalho apresentado nesta dissertação, onde extensões ao protocolo baseado na métrica XOR são propostas, de tal forma a melhorar sua estrutura de sinalização, reduzindo o número de informações necessárias na convergência das tabelas de rotas.

## **1.2 Objetivos e contribuições**

Esta dissertação busca, como o seu objetivo principal, estender o protocolo baseado em XOR, inserindo os filtros de Bloom na estrutura de sinalização, evitando que informações repetidas/desnecessárias sejam injetadas na rede. Desta forma, a convergência é acelerada, melhorando o comprimento dos caminhos, conseqüentemente, aumentando a duração dos caminhos, ao mesmo tempo reduzindo o atraso fim-a-fim e conseguindo melhor taxa de caminhos criados. Tempos maiores de duração significam mais tempo para a troca efetiva de dados, que no ambiente das VANETs é muito importante, dada as suas características abordadas na Seção 1.1.

Em primeiro lugar é proposta uma modificação no protocolo baseado na métrica XOR que visa lidar melhor com a mobilidade dos veículos, incorporando adaptações de forma a reduzir a troca da quantidade de informações de vizinhança

desnecessárias entre os nós da rede, buscando preencher de forma mais otimizada as tabelas de roteamento dos nós, lidando assim, com a dinamicidade dos cenários das VANETs. Em seguida é realizada uma comparação, com diversas densidades de nós, entre o protocolo baseado na métrica XOR proposto e alguns protocolos baseados na topologia e baseado na posição dos nós, caracterizando o desempenho desses protocolos em relação à taxa de disponibilidade dos caminhos, atraso na entrega dos pacotes, o número médio de saltos e o tempo médio de duração dos caminhos gerados pelos protocolos. Os resultados obtidos mostram que o protocolo baseado na métrica XOR proposto obteve resultados melhores que os outros protocolos avaliados, mostrando ser uma abordagem interessante a ser utilizada nas VANETs urbanas. Conforme pesquisas realizadas até o momento, este é o primeiro trabalho que analisa o desempenho de protocolos baseado na métrica XOR em cenários VANET urbanos.

### **1.3 Estrutura da dissertação**

Esta dissertação encontra-se dividida em 7 capítulos. No Capítulo 2 é abordada, de forma resumida, a arquitetura e as características que fazem das VANETs um tipo de rede especial e única. Além disso, a mobilidade existente nas VANETs, seja em ambientes de rodovias ou em ambientes urbanos, também é discutido de forma resumida.

No Capítulo 3 é realizada uma revisão sobre os protocolos de roteamento existentes para cenários de redes *ad hoc* móveis. De forma geral, são apresentadas as características de cada protocolo abordado.

No Capítulo 4 é apresentado o protocolo XOR voltado para redes *ad hoc* móveis. São apresentadas todas as características que envolvem o funcionamento deste protocolo: o processo de descoberta dos nós, o processo de roteamento, processo de aprendizagem e o conceito de visibilidade local.



No Capítulo 5 é apresentado o protocolo de roteamento baseado em OU-Exclusivo que integra os filtros de Bloom nas mensagens de *QUERY* do protocolo XOR, surgindo assim o protocolo XORBF.

No Capítulo 6 o protocolo XORBF é avaliado em um cenário urbano, comparando seu desempenho em relação ao protocolo baseado em OU-Exclusivo (sem a implementação dos filtros de Bloom), os protocolos baseados na topologia OLSR e AODV e o protocolo baseado na localização GPSR. As métricas avaliadas são: taxa de caminhos criados/disponíveis, comprimento do caminho (em número de saltos), tempo de duração dos caminhos e o atraso médio fim-a-fim.

Para finalizar, no Capítulo 7 é realizada uma análise sobre o trabalho realizado e os resultados obtidos. Em seguida são dadas ideias de possíveis trabalhos futuros.

## 2 Redes Ad Hoc Veiculares

Neste capítulo o objetivo é dar ao leitor uma visão inicial sobre as VANETs, detalhando suas características e arquiteturas.

O capítulo está disposto em 4 partes. Na primeira parte é dada uma introdução sobre o conceito das VANETs. Na segunda parte é explanado sobre as possíveis arquiteturas a serem exploradas nas VANETs. Na terceira parte são descritas as características das VANETs que as tornam um tipo de rede móvel desafiadora. Na quarta e última parte são abordadas as principais características da mobilidade dos veículos, tanto em ambientes de rodovias quanto em ambientes urbanos.

As redes veiculares surgiram a partir da ideia de que com o aumento da segurança dos veículos, seria possível diminuir o número de acidentes, reduzindo a quantidade de mortos e feridos, seja em vias urbanas ou em rodovias. Além disso, existem outros tipos de aplicações que objetivam melhorar a fluidez do trânsito, diminuindo engarrafamentos ou informando aos motoristas vias de circulação com melhores condições de tráfego, fornecendo conectividade à Internet, entre várias outras aplicações [13,14]. A tendência é de que as redes de comunicações que envolvam veículos utilizem uma abordagem híbrida (infraestruturada + *ad hoc*), bem definida e implementada, e em constante melhora de forma a serem eficientes no

oferecimento dos diversos serviços para os usuários móveis, especialmente nos cenários urbanos [15].

A cada lançamento de um novo veículo, o mesmo tem sido colocado no mercado com diversos recursos tecnológicos embutidos. Pode ser citados como exemplos dessas tecnologias o *Global Positioning System* (GPS) [16], alarmes de excesso de velocidade, sensores de estacionamento, câmeras, entre outros. Esses recursos, em geral, possuem como objetivo auxiliar o motorista na condução do carro seja em vias urbanas ou em rodovias.

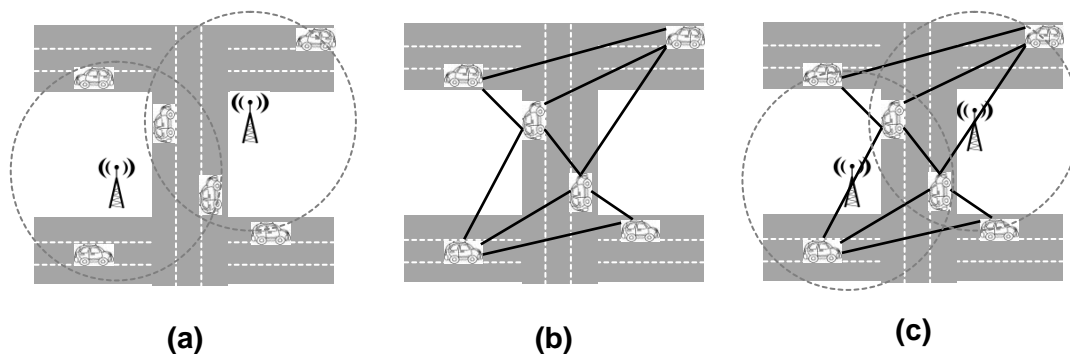
O advento dos recursos citados anteriormente tem ajudado bastante na busca da criação de um Sistema Inteligente de Transporte (*Intelligent Transportation System* – ITS)[13,14,15,17], sistema este que visa permitir a interação entre veículos, possibilitando comunicações móveis e o uso das mais diversas aplicações, que podem ser aplicações específicas para o trânsito, tais como prevenção de colisões, controle do fluxo de tráfego, monitoramento cooperativo do tráfego, além de aplicações que permitam o acesso à Internet. [13].

## **2.1 Arquitetura das VANETs**

Uma importante parte das VANETs que deve ser entendida é acerca da sua arquitetura. As MANETs não possuem uma infraestrutura fixa, centralizadora, que seja responsável pela realização e gerenciamento da comunicação entre os nós [14]. Portanto, por herdarem algumas características das MANETs, as VANETs poderiam ser consideradas como uma instanciação de uma MANET. As VANETs podem ser implementadas nas seguintes formas: infraestruturada (a), *ad hoc* (b) e híbrida (c) [13], conforme pode ser visto na Figura 2.1.

Na Figura 2.1(a) é mostrada uma arquitetura de comunicação infraestruturada em que normalmente são utilizados equipamentos como pontos de acesso ou estações base, que podem funcionar como *gateways* fornecendo conexão à Internet,

localizados à beira da estrada. Mas esta é uma arquitetura muito complexa de ser colocada em prática, considerando os custos envolvidos neste tipo de arquitetura para alcançar uma conectividade satisfatória, devido à necessidade de instalação de vários pontos de acesso ao longo das estradas. Alguns trabalhos [18] abordam este tipo de arquitetura como *Vehicle-To-Infrastructure (V2I)*, assim como o trabalho [19] utiliza o termo *Roadside-to-Vehicle Communication Systems (RVC systems)*.



**Figura 2.1 - Arquitetura das VANETs.**

Na Figura 2.1(b) é mostrada uma arquitetura do tipo *ad hoc*, em que nenhuma estrutura fixa é utilizada, logo, fácil perceber que a comunicação é realizada diretamente entre os veículos. Alguns trabalhos [18,19,20] também chamam esse tipo de comunicação como *Vehicle-to-Vehicle (V2V)*. Esta arquitetura reduz os custos com a instalação de estações base ou pontos de acesso na beira das estradas/ruas, facilitando sua implementação, porém por se tratar de uma comunicação direta veículo-a-veículo, a conectividade da rede dependerá da densidade e do padrão de mobilidade dos veículos, o que vem a ser uma desvantagem dessa arquitetura.

Na Figura 2.1(c) é mostrada uma arquitetura onde são combinadas as arquiteturas infraestruturada e *ad hoc*. Nesta arquitetura, é possível instalar uma infraestrutura mínima de forma a aumentar a conectividade da rede e fornecer serviços de interconexão com um menor número de saltos. Ao mesmo tempo é possível utilizar a comunicação em múltiplos saltos, numa situação em que a infraestrutura não

estivesse disponível, como por exemplo, um desastre natural que viesse a danificar as estações base presentes na beira da estrada.

## 2.2 Características das VANETs

As VANETs possuem características únicas que as diferenciam de quaisquer outros tipos de rede. Nós com alta mobilidade e sem limitações de energia e armazenamento de dados, a presença de obstáculos, tecnologias *on-board*, entre outras características, devem ser levadas em conta ao projetar qualquer tipo de solução voltada para este tipo de rede. A seguir segue uma descrição mais detalhada sobre essas diversas características que fazem das VANETs um tipo de rede única.

- **Topologia extremamente dinâmica:** devido aos nós existentes nas VANETs serem veículos, e que estes possuem alta velocidade em sua movimentação, a topologia das VANETs frequentemente é alterada. Já existem trabalhos [21, 22] que tentam encontrar o tempo de vida aproximado de uma conexão entre veículos. Devido à dinamicidade existente neste tipo de rede, a conectividade das VANETs também muda com frequência, principalmente em ambientes que possuem baixa densidade de veículos. Em ambientes onde possui uma baixa densidade veicular, o uso da arquitetura infraestruturada talvez seria uma solução para fornecer conectividade por exemplo à Internet;
- **Sem limitações de energia e armazenamento:** o porte dos nós existentes, tais como carros, ônibus, caminhões, entre outros, dá a possibilidade de possuírem energia e capacidade de armazenar e processar informações;
- **Mobilidade:** Modelos de mobilidade deveriam ser considerados nas VANETs devido às suas características únicas de mobilidade. Ruas urbanas, rodovias ou BRs, prédios, árvores, condições de temperatura e outros efeitos devem ser levados em conta nas VANETs, mostrando que a movimentação e a comunicação dos nós são limitadas a esses tipos de ambientes além da necessidade de se levar em conta também o comportamento do motorista ao

volante, que pode reagir de diferentes formas, dependendo do tipo de mensagem que venha a receber ao volante;

- **Modelos de propagação:** A presença de prédios, árvores, outros veículos e condições de temperaturas deveriam ser considerados ao realizar a comunicação entre os nós;
- **Uso de tecnologias *on-board*:** Podem existir nos veículos equipamentos tais como sensores e GPS. Este último é de grande auxílio ao considerar o seu uso nas soluções de roteamento, por fornecer informações de localização e que o seu uso está sendo cada vez mais crescente nos veículos;

No geral, as diversas características existentes nas VANETs, fazendo-a ser única, mesmo sendo uma instanciação das MANETs, fazem com que os protocolos desenvolvidos para as MANETs não alcancem uma vazão de dados satisfatória [23] ou em alguns casos chegue até mesmo a ser inviável a utilização desses protocolos nas VANETs [24]. Por isso que, ao projetar protocolos para as VANETs, as características citadas e explicadas anteriormente devem ser levadas em consideração, de forma a conseguir resultados realistas.

### 2.3 Cenários de Mobilidade em VANETs

Um aspecto muito importante a ser considerado nas VANETs é a mobilidade dos veículos. Dependendo das características de mobilidade de cada veículo, o impacto no desempenho de um protocolo pode ser muito significativo.

Os modelos de mobilidade para VANETs buscam representar, de forma mais próxima possível, o comportamento real do tráfego veicular [24]. Por isso, é importante utilizar modelos de mobilidade que reflitam o comportamento deste tráfego de maneira mais realística possível.

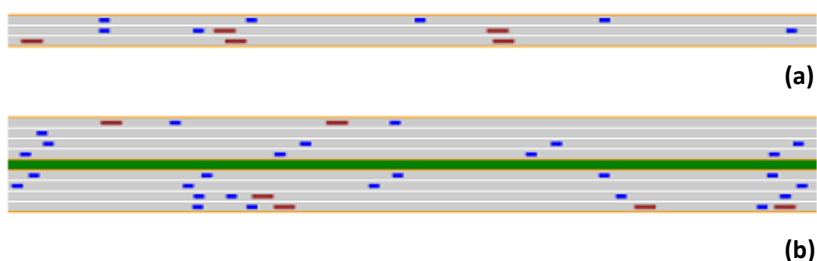
Levando em consideração que o método utilizado neste trabalho é a simulação, devido às diversas dificuldades existentes em adotar experimentos práticos em VANETs, como o alto custo financeiro, necessidade de grande número de

peças, podendo ser necessário também ter ambientes e condições climáticas favoráveis. Assim, o objetivo das próximas seções é dar ao leitor uma visão mais ampla acerca das características que formam os cenários de rodovias e os cenários urbanos.

### 2.3.1 Mobilidade em Rodovias

A maioria dos trabalhos, pesquisados até o momento, utilizam as rodovias como cenário de investigação [9,25,26,27,28]. As rodovias são vias importantes que, em geral, fazem a ligação entre cidades vizinhas e que, dependendo da região geográfica onde estão inseridas e do horário, podem ter uma densidade maior ou menor de carros.

Nas rodovias, porém, os desafios impostos por este tipo de cenário não são tão desafiadores, se comparados ao cenário urbano, que será visto na próxima seção, pois as rodovias consistem, em sua maioria, de vias de ida e volta (bidirecional) que podem variar de uma faixa até várias faixas. Os veículos neste tipo de cenário tendem a se mover em alta velocidade, exceto em situações como engarrafamentos ocorridos por blitz policial, manutenção da via ou a ocorrência de um acidente. Nas rodovias quando comparado ao ambiente urbano, também não existem tantos obstáculos tais como prédios, árvores, casas, entre outros, o que dificultaria a propagação do sinal de rádio. Na Figura 2.2 é mostrado um exemplo de rodovias.



**Figura 2.2 – Rodovia onde os carros são representados pela cor azul e os caminhões pela cor vermelha. (a) Rodovia unidirecional com 3 faixas; (b) rodovia bidirecional com 4 faixas em cada direção [28].**

Devido ao fato de ter menor complexidade ao modelar tais cenários, inicialmente o cenário de rodovia foi mais investigado pelos pesquisadores da área. Porém, por serem mais desafiadores, devido a fato de existirem ferramentas que facilitem a modelagem e de serem essenciais para a adoção da tecnologia VANET pelas empresas automobilísticas, os cenários urbanos começaram a ganhar a atenção por parte dos pesquisadores.

### **2.3.2 Mobilidade em Ambientes Urbanos**

A pesquisa das VANETs em cenários urbanos tem atraído cada vez mais a atenção de governos, pesquisadores e empresas automobilísticas. Um dos motivos é que este tipo de cenário é muito desafiador devido a várias características, tais como:

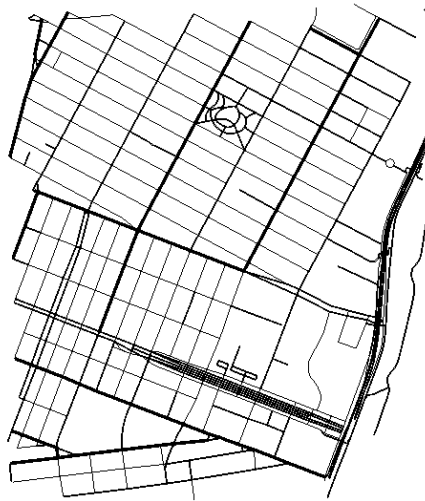
- Presença de obstáculos, tais como prédios e árvores, o que dificulta a propagação do sinal de rádio;
- Densidade veicular variável dependendo da região e do horário;
- Presença de semáforos, redutores de velocidade, placas de pare em cruzamentos, mudanças de faixas e limitadores de velocidade, o que impacta diretamente na aceleração e desaceleração dos veículos;
- Layouts das ruas que determina a quantidade de cruzamentos existentes em uma determinada área da cidade e que em geral possuem obstáculos de forma a interromper ou dificultar a propagação do sinal de rádio entre os veículos;
- Aceleração, desaceleração e velocidade variadas nos diversos ambientes.

Além das características acima serem motivadoras para o aumento de pesquisa em VANETs neste tipo de cenário, outro motivador é o crescente aumento no número de carros e pelo fato de que as pessoas estão passando cada vez mais uma parte significativa do seu tempo diário no trânsito, estas condições fazem com que, mercadologicamente falando, haja um grande nicho a ser explorado, seja para

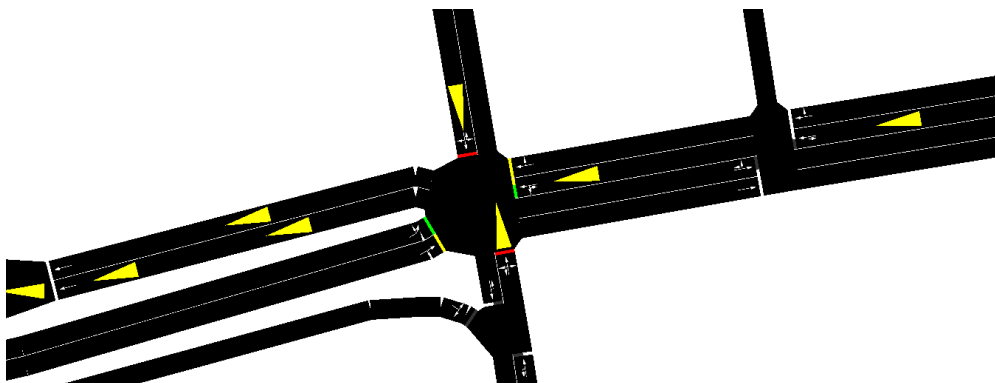


fornecer entretenimento e/ou aumentar a segurança dos passageiros, entre diversos outros.

Na Figura 2.3 é mostrado um exemplo de cenário urbano, com dimensões de 1.7Km x 2.0Km, onde é possível perceber um mapa bem mais complexo do que o cenário de rodovia mostrado na Figura 2.2, devido a presença de muitos cruzamentos, ruas sem saída, a existência de vias com semáforos e limitações de velocidade. Além disso, na Figura 2.4 uma região do mapa da Figura 2.3 é visualizada, onde existem também várias faixas de rolamento, sentidos permitidos de circulação, mostrando a complexidade dos cenários urbanos.



**Figura 2.3 - Mapa de uma região da cidade de Nova York representando de um cenário urbano.**



**Figura 2.4 – Aproximação do mapa da Figura 2.3, mostrando diversas características existentes nos cenários urbanos**

Portanto, é possível perceber que os cenários urbanos são mais complexos devido às características citadas anteriormente, sendo este um dos motivos que atrai a atenção da comunidade científica. Pode ser dito que a mobilidade dos veículos nos cenários urbanos deve ser um aspecto muito importante a ser considerado nas simulações, pois pode influenciar diretamente nos resultados dos protocolos de roteamento.

O objetivo deste capítulo foi abordar alguns detalhes das VANETs que as tornam um tipo de rede única, assim como suas arquiteturas, a mobilidade especial existente neste tipo de rede, de forma a poder visualizar melhor como as redes VANETs podem ser formadas. No próximo capítulo é realizada uma revisão sobre alguns protocolos de roteamento voltadas para redes *ad hoc* móveis, apresentando suas características e suas diferenças.

### **3 Trabalhos Relacionados**

As VANETs têm recebido cada vez mais atenção por parte das empresas automobilísticas, governo, pesquisadores, devido à grande possibilidade das aplicações conseguirem dar suporte quanto ao aumento da segurança no trânsito e a possibilidade de fornecer entretenimento a bordo de veículos. Neste capítulo, alguns protocolos de roteamento baseado em topologia e na posição geográfica serão apresentados, dando assim, base para o trabalho apresentado nos próximos capítulos.

Este capítulo está dividido em 2 partes. Na primeira parte são apresentados os protocolos baseados na topologia e suas características. Na segunda parte são apresentados os protocolos baseados na posição geográfica dos nós assim como suas características.

Nas VANETs, o roteamento que também é chamado de encaminhamento de pacotes é uma questão crítica, tendo em vista a alta mobilidade dos nós neste tipo de rede, pois os atuais protocolos de encaminhamento não foram projetados para lidar com tal característica. Duas tarefas são importantíssimas quando falamos em protocolos de roteamento: a primeira é escolher as rotas entre os diversos pares origem-destino de tal forma a garantir a entrega correta dos pacotes entre elas; a segunda consiste na utilização de vários algoritmos e estruturas de dados,

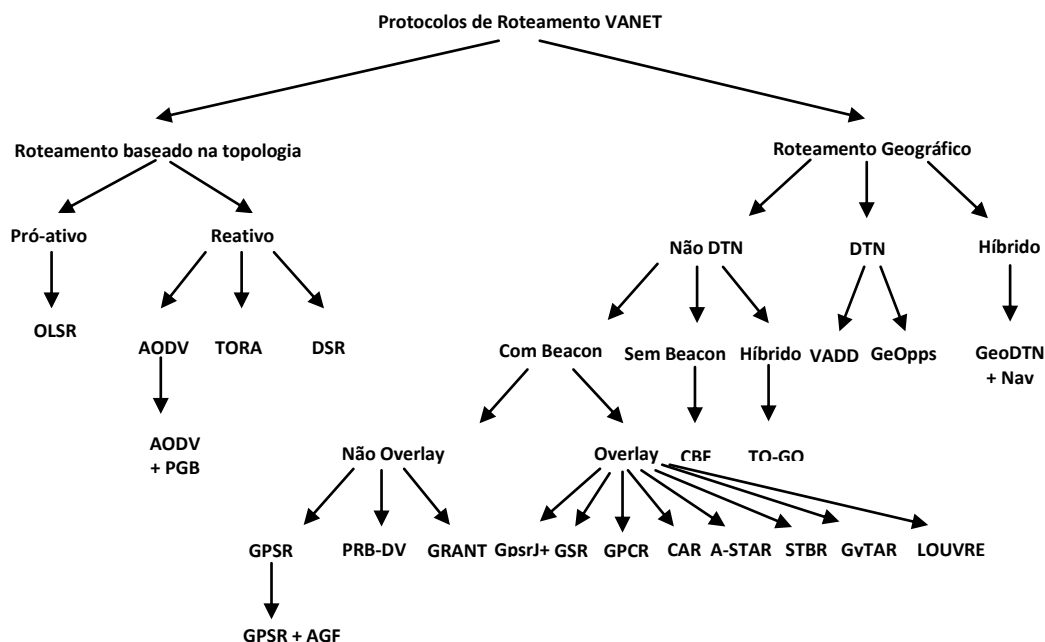
responsáveis pela criação e manutenção das tabelas de roteamento para que o roteamento dos pacotes seja possível de ser feito.

### 3.1 Roteamento baseado na Topologia

No início da década de 70, os protocolos de roteamento para redes *ad hoc* surgiram com o objetivo de suprir as deficiências que existiam nos protocolos voltados para redes infraestruturadas, devido ao baixo desempenho alcançado por estes protocolos em cenários móveis. Dentre os mais diversos protocolos que têm surgido, é possível destacar as seguintes características principais:

- Redução do consumo de energia;
- Reduzir ao máximo o uso da largura de banda com mensagens de controle;

Os protocolos que se encontram nesta categoria buscam selecionar os nós que irão encaminhar os pacotes baseados nas informações dos links existentes na rede. A partir da informação desses links é possível descobrir qual é o melhor caminho existente entre o par origem-destino. No trabalho de [29] é realizada uma classificação dos protocolos de roteamento utilizados nas VANETs, conforme Figura 3.1, onde é visto que o roteamento baseado em topologia é dividido em proativo e reativo.



**Figura 3.1 - Taxonomia de vários protocolos de roteamento utilizados em VANETs [29].**

Com base nesta troca de mensagens, é possível lidar com as mudanças da topologia da rede. Dessa forma, cada nó possui, previamente, uma tabela que contém qual ou quais são os seus próximos saltos na direção de todos os destinos disponíveis na rede. Um problema dos protocolos desta categoria é que há um custo na utilização da banda em manter as informações na tabela de rotas de cada nó, e isso é mais crítico ainda nos ambientes das VANETs, onde a mobilidade dos nós é alta. Os protocolos pró-ativos mais conhecidos são o *Destination Sequence Distance Vector* (DSDV) [30] e o *Optimized Link State Routing* (OLSR) [4].

Os protocolos de roteamento reativos possuem como característica principal a criação de rotas somente quando há necessidade de encaminhar um pacote e a não necessidade de conhecer rotas para todos os destinos disponíveis na rede. Quando um nó qualquer precisar enviar um pacote para outro nó de destino, o processo de descoberta de rotas é iniciado e só é finalizado quando tal rota é encontrada ou quando tal rota não é possível de ser encontrada. Esse processo de descoberta é realizado através da inundação (*flooding*) de pacotes dentro da rede objetivando conseguir um caminho em direção ao nó de destino. A manutenção das rotas é realizada enquanto o nó de destino for alcançável ou enquanto for necessário utilizar essa rota. Os protocolos reativos mais conhecidos são o *Dynamic Source Routing* (DSR) [3] e o *Ad hoc On-Demand Distance Vector Routing* (AODV) [2].

No ambiente das VANETs, e em especial para as aplicações que são sensíveis ao atraso como no caso das aplicações de segurança, o protocolo AODV, que é classificado como reativo, não seria uma solução viável, considerando o atraso gasto para a descoberta do caminho origem-destino [6]. Sendo assim, o uso dos protocolos proativos, tal como o OLSR, seria uma melhor solução, pois os nós neste caso já possuem uma tabela com as rotas disponíveis a distância de um salto. Logo não há atraso na busca de rotas e é disseminada a mensagem mais rapidamente, mesmo considerando que os protocolos pró-ativos consomem mais banda para manter as informações das rotas existentes [31].

### 3.1.1 – AODV

Desenvolvido por Perkins e Royer em 1999, o protocolo *Ad hoc On-Demand Distance Vector* (AODV) [2] é baseado no algoritmo de vetor de distância, assim como o protocolo DSDV [30].

Um detalhe importante a ser considerado neste protocolo é sobre as quebras de enlaces. Quando essas quebras ocorrem, notificações são enviadas imediatamente para os nós afetados. Assim como o protocolo DSR [3], este protocolo possui duas fases: a descoberta de novas rotas e a manutenção das mesmas.

Na fase de descoberta de rotas, quando um nó que deseja enviar um pacote para um determinado destino e que não se encontra na tabela de roteamento do nó de origem (desconhecimento da rota até o nó de destino ou a uma rota anteriormente disponível está expirada), é realizado um *broadcast* de um tipo de mensagem chamada de *Route Request* (RREQ) para todos os vizinhos do nó de origem. Depois da transmissão deste pacote, o nó de origem fica a espera de receber de volta a mensagem *Route Reply* (RREP). Caso o nó não a receba durante certo tempo, o nó pode retransmitir o RREQ ou concluir que, de fato, não existe rota para aquele destino.

O nó que recebe a RREQ cria uma rota temporária inversa para o nó de origem em sua tabela de roteamento, com o objetivo de que exista um controle da rota e é realizado para que o nó que gerou a mensagem RREP encontre o caminho de volta ao nó de origem, isto é, o nó solicitante da rota através do pacote RREQ. Tais rotas são temporárias, tendo um tempo de vida bem limitado. Assim que o nó de destino recebe o pacote RREQ, uma mensagem de RREP é gerada em modo *unicast*<sup>1</sup> de volta ao nó de origem. Dessa forma, a rota vai sendo estabelecida enquanto o pacote RREP vai passando pelos nós intermediários até o momento em que o pacote RREP chega ao nó de origem, estabelecendo assim, a rota entre origem-destino.

Outra fase importante relativa ao protocolo AODV é a manutenção das rotas. Tal manutenção é realizada através da transmissão de mensagens de *HELLO* para os

---

<sup>1</sup> Também chamado de salto a salto

vizinhos a um salto de distância. Tais mensagens são necessárias para que o nó avise aos seus vizinhos que está presente e que, desta forma, as rotas que utilizem este nó na sequência de roteamento continuem a ser confirmadas como válidas. Caso tais mensagens de *HELLO* não sejam recebidas, o nó pode entender que o nó vizinho não está mais alcançável, seja por motivo de ter saído do seu raio de alcance de rádio ou por não estar mais funcional e, portanto, irá marcar tal enlace como quebrado.

Quando um nó detecta que um nó vizinho não é mais alcançável, o mesmo é removido da tabela de roteamento e uma mensagem do tipo *Route Error* (RERR) é enviada, e assim, a quebra de enlace é informada para todos os nós que utilizavam rotas que passavam pelo nó inalcançável ou que tinham este nó como um nó de destino. Um nó que receber esta mensagem, encaminha tal mensagem para todos os nós que foram afetados, de forma que todos sejam notificados e que as suas tabelas de roteamento sejam atualizadas.

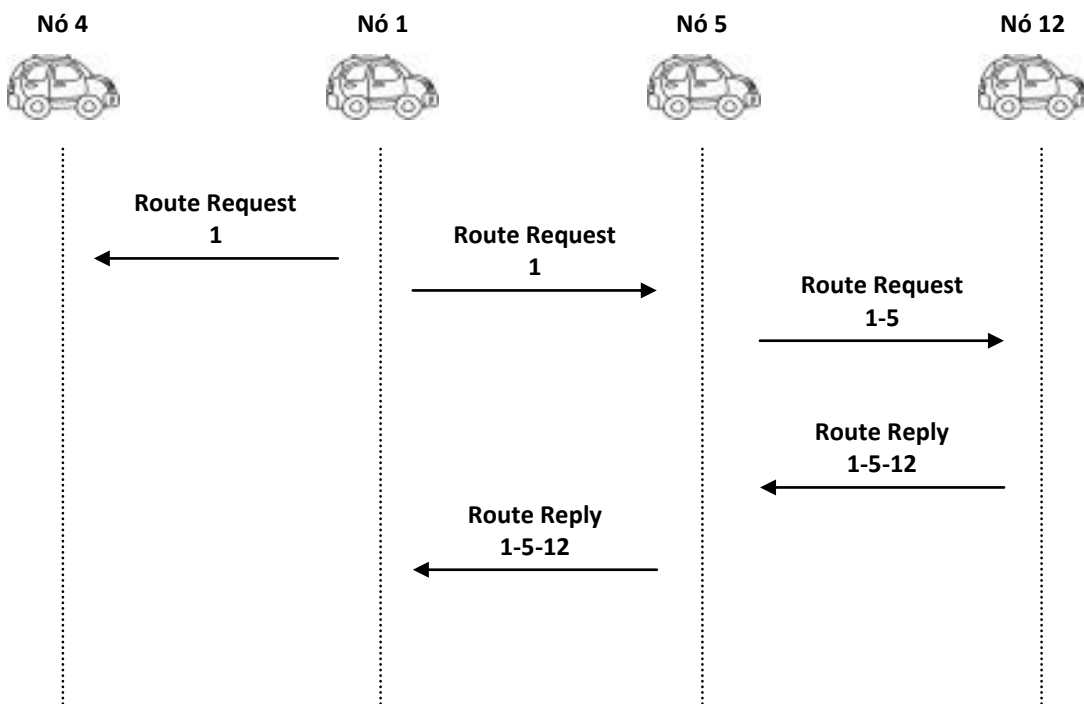
Uma vantagem do AODV em relação aos protocolos baseados em topologia pró-ativos, tais como o DSDV e o OLSR, é o número reduzido de mensagens de sinalização [31]. Entretanto, como já visto, quando ocorre uma quebra de enlace, a mensagem RRRR percorre todos os nós afetados até o nó de origem, porém, pode acontecer da distância entre o nó de origem e o nó de destino ser muito grande, em número de saltos, o que pode fazer com que um mesmo nó receba a mesma mensagem RRRR, porém de diferentes emissores. Este tipo de solução possui um elevado custo de mensagens de sinalização, em especial, torna-se mais grave ainda conforme aumenta a densidade dos nós, o que tende a aumentar a probabilidade de quebras de enlaces.

### **3.1.2 – DSR**

O protocolo *Dynamic Source Routing* (DSR) [3] é um protocolo que permite que as rotas para qualquer destino sejam descobertas de forma dinâmica. Uma das características importante deste protocolo consiste no fato de que cada nó na rede

possui em sua tabela de roteamento todos os saltos a partir da origem até o destino, ou seja, armazena todo o caminho que o pacote deve percorrer. Este protocolo pode ser dividido em duas partes: a descoberta de rotas e a sua manutenção.

O processo de descoberta de rotas é iniciado quando um determinado nó de origem deseja enviar um pacote para um determinado nó de destino, e claro, precisa de uma rota para conseguir tal objetivo. Na Figura 3.2 é ilustrado como é realizado a descoberta de rotas pelo protocolo DSR.



**Figura 3.2 - Exemplo da descoberta de rotas do protocolo DSR.**

Na Figura 3.2 o nó 1 deseja enviar um pacote para o nó 12 e começa a transmitir para todos os seus nós vizinhos localizados a um salto um pacote do tipo *Route Request* (RREQ). Todo nó que venha a receber o pacote RREQ, irá procurar em suas respectivas tabelas de roteamento alguma rota para o nó 12. Caso nenhuma rota seja encontrada, a mensagem de RREQ é retransmitida e o nó que a transmite é adicionado à sequência de nós já visitados pela mensagem. Tal mensagem irá ser propagada até que o nó de destino seja encontrado ou encontre algum nó que tenha, em sua tabela de roteamento, o caminho para chegar ao nó 12. Ao acontecer isso,



uma mensagem de *Route Reply* (RREP) é enviada de volta para o nó de origem, na forma *unicast*, contendo a sequência de nós com os quais é possível chegar ao nó de destino. Assim que encontrada, uma rota é guardada na tabela de roteamento do nó por um determinado intervalo de tempo, dando início assim ao processo de manutenção de rotas.

O processo de manutenção de rotas do protocolo DSR é semelhante ao processo existente no protocolo AODV. A manutenção de rotas existe para identificar quando há uma perda de comunicação dos vizinhos de um nó, assim não sendo possível utilizar uma determinada rota já previamente existente em sua tabela de roteamento. Situações em que um nó tenha se desligado ou um nó esteja fora do alcance do raio de comunicação dos seus vizinhos são exemplos das quais rotas não podem ser mais utilizadas. Quando um nó detecta que um vizinho não é alcançável, remove-o da tabela de roteamento e transmite uma mensagem do tipo *Route Error* (RERR) para todos os nós que utilizavam tal vizinho em suas tabelas de roteamento até o instante em que todos os nós que foram afetados sejam notificados. Pode acontecer, casualmente, de a mensagem chegar aos nós de origem que podem por decidir parar a transmissão de dados ou reiniciar um novo processo de descoberta de dados.

Uma característica interessante do protocolo DSR é que os nós aprendem as rotas através das informações existentes nos pacotes. Como é possível perceber na Figura 3.2, na rota do nó 1 (nó de origem) para o nó 12 (nó de destino), o nó aprende a rota do nó 12, mas também aprende a rota para o nó 5. Continuando, o nó 5, por sua vez, aprende a rota do nó 1 para o nó 12 e o nó 12 aprende a rota do nó 1 para o nó 5. Tal forma de conhecimento reduz, de forma significativa, a carga que o *Route Request* (RREQ) introduz na rede. Porém, há uma sobrecarga de tráfego na rede que é em função dos números de nós que o pacote RREQ visita, pois a cada endereço visitado o pacote aumenta de tamanho, devido à inclusão do endereço dos nós visitados no corpo da mensagem. Por isso, há uma tendência de que em redes com elevado

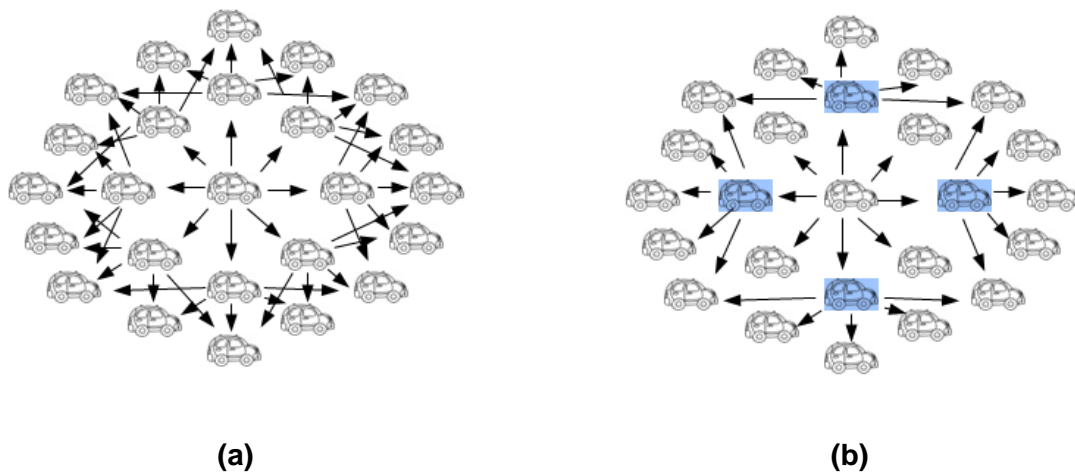
número de nós, o tamanho das mensagens seja preocupante, devido à carga imposta tornar-se muito alta por adicionar os endereços dos nós visitados.

### 3.1.3 – OLSR

O *Optimized Link State Routing* (OLSR) [4] é um protocolo pró-ativo e um dos principais protocolos de roteamento existentes para redes *ad hoc*. Como o próprio nome do protocolo diz, ele é uma otimização dos protocolos de *link state* existentes, devido ao fato de reduzir o tamanho do pacote de sinalização, assim como, o número de transmissões necessárias de tais pacotes.

A redução do tamanho do pacote de sinalização é obtida com a limitação do número de nós, localizados a um salto, que são responsáveis por disseminar a informação da topologia da rede. Tais nós são chamados de *Multipoint Relays* (MPR). Desta forma, ao invés de ser realizada a inundação (*flooding*) da informação de sinalização em toda a rede, os pacotes são enviados somente pelos MPRs escolhidos pelo nó. Na Figura 3.3 é possível percebermos a diferença do tráfego gerado entre uma inundação pura e uma inundação utilizando MPRs.

É possível perceber que na Figura 3.3(a) a troca de informações é muito mais intensa do que na Figura 3.3(b), devido ao fato de que a utilização dos MPRs objetiva a redução do número de retransmissões necessárias, fazendo com que não exista nós recebendo os pacotes mais de uma vez.



**Figura 3.3 - Tipos de inundação: inundação pura e inundação por MPRs.**

Três tipos de mensagens compõem o protocolo OLSR: *HELLO*, *Topology Control (TC)* e *Multiple Interface Declaration (MID)*. A mensagem *HELLO* é responsável por informar os nós vizinhos a um salto do nó que está transmitindo tal mensagem aos MPRs já escolhidos e, também, contém uma lista de vizinhos cujos *links* bidirecionais ainda não foram confirmados. A mensagem *HELLO* é transmitida periodicamente para todos os nós vizinhos localizados a um salto. As mensagens TC são transmitidas pelos MPRs quando ocorrem mudanças significativas na topologia da rede, sendo, em seguida, transmitidas para todos os nós da rede. Além disso, as mensagens de TC também contêm os MPRs escolhidos. E por fim, conforme o próprio nome já diz, as mensagens MID são responsáveis por informar se o nó está funcionando com o protocolo OLSR em mais de uma interface.

Como já explicado anteriormente, o OLSR utiliza os MPRs para evitar a inundação. Cada nó irá escolher alguns nós da vizinhança localizada a um salto, que são os nós MPRs, assim, fazendo com que cada nó que esteja localizado a dois saltos possa se comunicar com pelo menos um membro do grupo de MPRs [31]. Todo nó que é eleito como MPR é marcado como vizinho MPR e tal informação é transmitida na próxima mensagem *HELLO*. Este nó eleito como MPR, irá adicionar o nó que o escolheu a uma lista chamada de *MPR Selector*, sendo assim, as informações sobre a

topologia da rede só serão encaminhadas por um nó se tais informações forem provenientes de nós que estão na sua lista de *MPR Selector*. A outra otimização do protocolo OLSR é a minimização do número de nós que anunciarão as informações sobre a topologia da rede. Somente os nós escolhidos como MPR divulgarão os nós que os elegeram (contidos na lista *MPR Selector*) para toda a rede. Dessa forma, é possível criar as tabelas de roteamento, a partir das informações topológicas e dos vizinhos localizados a um e a dois saltos de todos os nós da rede.

Por ser um protocolo de roteamento de natureza pró-ativa, o OLSR gera uma carga de tráfego de sinalização muito alta. Para fazer a manutenção das tabelas de roteamento dos nós existentes na rede, cada vez que a topologia da rede é alterada, uma grande quantidade de mensagens TC é gerada. A introdução do conceito de MPRs permite que tal sobrecarga do tráfego de sinalização seja reduzida.

### **3.2 Roteamento Geográfico**

Os protocolos de roteamento podem sofrer sobrecargas de comunicações em diferentes situações e dependendo de diversos fatores. Um dos fatores que pode impactar no desempenho dos protocolos de roteamento é a mudança frequente na topologia da rede.

No caso das VANETs, onde os nós, por serem veículos, possuem um alto grau de mobilidade, resulta em freqüentes alterações na topologia da rede, havendo assim, uma sobrecarga grande de tráfego de sinalização em relação à atualização das tabelas de roteamento. Esta parte da dissertação aborda outra categoria de protocolos de roteamento que objetiva lidar com essa mobilidade e busca minimizar o número de atualizações necessárias levando em consideração a localização dos nós.

Nos protocolos de roteamento baseado em posição, ou também chamados na literatura de protocolos de roteamento geográfico, os nós devem possuir a capacidade de determinar a sua própria posição e, além disso, conhecer a posição dos outros nós.

Em geral, o conhecimento dessa posição é obtido através de um serviço de posicionamento baseado em satélite, tais como o GPS (EUA) [16], GLONASS (Rússia) [32] ou Galileo (Europa) [33], onde é possível calcular a posição absoluta dos veículos com uma precisão de poucos metros. As decisões de encaminhamento dos pacotes levam em consideração a posição do nó de destino, a posição do nó que irá realizar o encaminhamento dos pacotes e as posições dos nós vizinhos. A seguir, são apresentados alguns protocolos de roteamento que adotam estes conceitos.

### 3.2.1 – GPSR

O protocolo *Greedy Perimeter Stateless Routing* (GPSR) [34] utiliza o conceito da escolha do vizinho mais próximo ao nó de destino para encaminhar os pacotes. Neste protocolo, todo nó possui conhecimento tanto da sua posição física quanto das posições dos seus vizinhos e da posição do destinatário. Durante o funcionamento deste protocolo, um tipo de mensagem especial é transmitido para os vizinhos localizados a um salto. Esta mensagem é conhecida como *beacon*. Tal mensagem é responsável por informar a identificação do nó e sua localização. Através destes *beacons*, todos os nós adquirem informações acerca da localização dos nós vizinhos a um salto de distância. Caso um nó qualquer não receba os *beacons* de um vizinho por um determinado período de tempo, assume-se que o vizinho está fora do alcance do sinal de rádio e, logo, é excluído da lista de vizinhos. O protocolo GPSR normalmente é dividido em duas partes: encaminhamento *greedy*<sup>2</sup> e o encaminhamento sobre o perímetro.

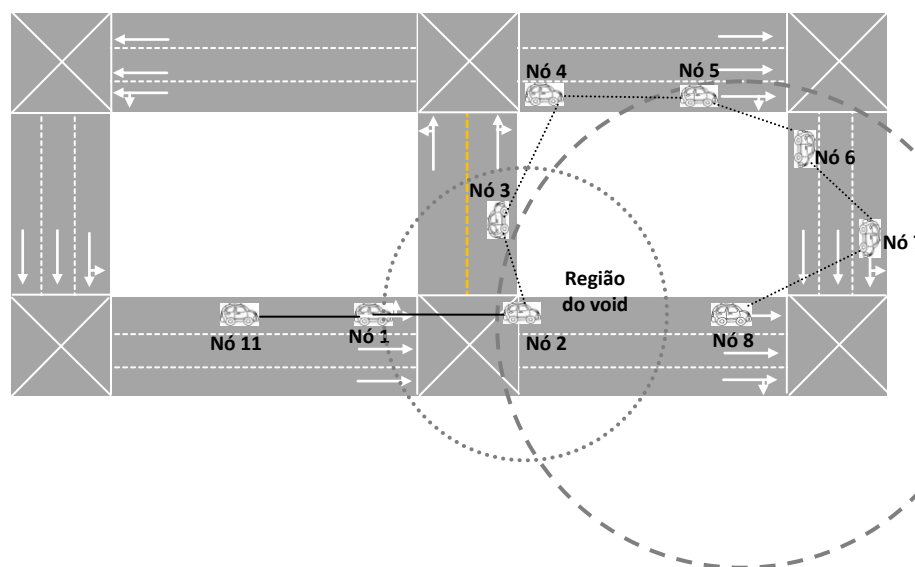
No encaminhamento *greedy*, o pacote é encaminhado para o vizinho que está mais próximo do nó de destino, partindo do princípio que o nó vizinho está dentro do raio de alcance de rádio da sua última localização. Em cenários de alta densidade,

---

<sup>2</sup> Também chamado de guloso

esta técnica tende a funcionar de maneira eficiente, porém, em cenários com baixa densidade, esta técnica pode não apresentar resultados interessantes.

Na Figura 3.4, é apresentado um exemplo de funcionamento do protocolo GPSR.



**Figura 3.4 - Situação de máximo local do protocolo GPSR.**

Na Figura 3.4, o nó 11 deseja enviar um pacote para o nó 8, o pacote é encaminhado de forma *greedy* até chegar ao nó 2. Neste momento, ocorre o que é chamado de máximo local ou ótimo local [35], isto é, não existe nenhum outro nó mais próximo ao nó de destino do que ele mesmo, no caso, o nó 2 (o círculo tracejado representa a distância entre o nó 2 e nó 8 e o círculo pontilhado representa a área de alcance do sinal de rádio o nó 2). Dessa forma, o pacote não pode ser transmitido, até mesmo no caso em que exista uma rota para o destino, que é representada pela linha tracejada que vai do nó 2 até o nó 8 originando assim o *void*<sup>3</sup>. O *void* acontece quando o nó mais próximo do nó de destino é ele mesmo, porém não consegue encaminhar a informação para o nó de destino porque o encaminhamento *greedy* falha.

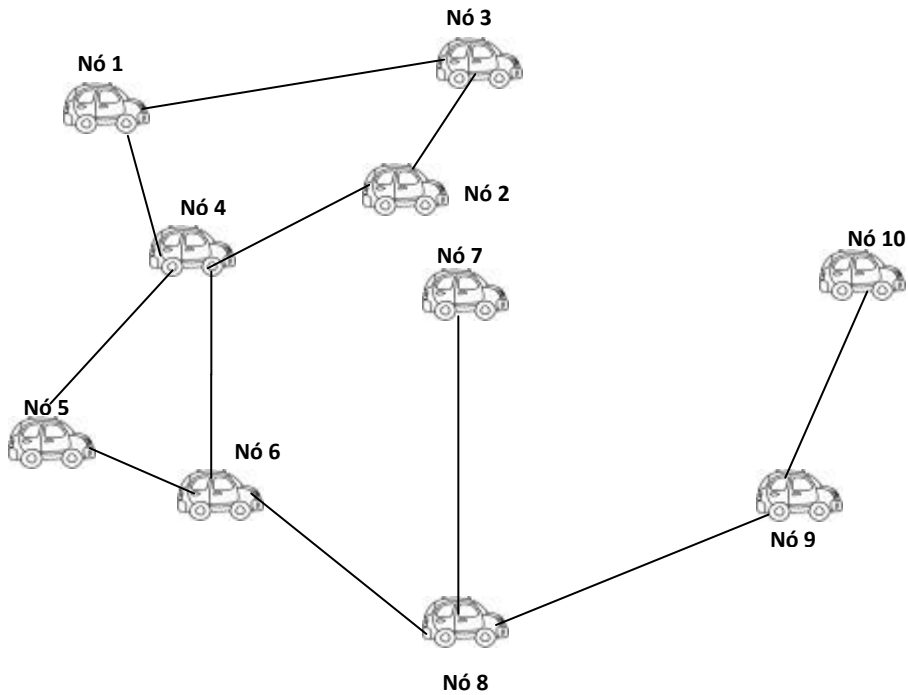
<sup>3</sup> Também conhecido como “vazio”

Como já visto no parágrafo anterior, o encaminhamento *greedy* falha quando um pacote chega a um nó que contém um *void*, (na Figura 3.4, é o nó 2). Quando o protocolo GPSR alcança um *void*, o encaminhamento em modo perímetro é utilizado de forma a conseguir encaminhar a informação evitando a região do *void*. O encaminhamento no modo perímetro cria um grafo planar utilizando os nós localizados na região onde o *void* ocorreu para conseguir encaminhar os pacotes para o nó destino, mesmo sendo esta distância maior até o nó de destino. Uma vez montado o grafo uma regra conhecida como “regra da mão direita” determina qual o próximo salto da comunicação. Tal regra define que quando um pacote chega a um nó X vindo de Y, a próxima aresta atravessada será a seguinte na sequência anti-horária do X em relação à aresta (X,Y).

De forma a evitar cruzamentos entre as arestas, levando a existência de loops, o que pode dificultar o envio de pacotes utilizando a regra da mão direita, um grafo é planarizado<sup>4</sup>. Dessa forma existem cruzamentos entre as arestas, antes de ser aplicada a regra da mão direita. Isto é conseguido com a utilização de um algoritmo que produz um grafo de vizinhança relativa [36] ou um grafo Gabriel [34]. Na Figura 3.5 é mostrado um exemplo da criação de um grafo de vizinhança relativa, onde é visto que não há cruzamento entre as arestas de forma a evitar *loops*.

---

<sup>4</sup> Um grafo é chamado de planar quando não existe cruzamento entre as arestas



**Figura 3.5 - Exemplo de um grafo de vizinhança relativa**

Devido à existência de edifícios e outros fatores que podem obstruir a propagação do sinal de rádio nos cenários urbanos, restringindo as rotas somente às ruas, o modo de perímetro é frequentemente utilizado, devido ao *void*. O grafo de conectividade planarizado, vai fazer com que as mensagens sejam enviadas aos vizinhos que estão mais distantes do nó de destino. Dessa forma, mais nós irão rotear as mensagens, o que causará um atraso maior assim como um aumento no número de saltos até alcançar o destino. Outro fator que agrava ainda mais o desempenho do protocolo GPSR em cenários urbanos é que, devido à alta mobilidade dos veículos, mais ciclos (*loops*) poderão ser originados enquanto o GPSR estiver em modo de perímetro, de forma que a persistência do uso da regra da mão direita faça com que o pacote seja transmitido na direção inversa ao nó de destino. Devido a estes problemas, outros protocolos voltados para ambientes urbanos foram propostos tais como o: *Geographic Source Routing* (GSR) [37], o *Anchor-Based Street and Traffic Aware Routing* (A-STAR) [38] e *Improved Greedy Traffic Aware Routing* (GyTAR) [39].



### 3.2.2 – GSR

Devido às características do protocolo GPSR, que em ambientes urbanos, por causa da presença de obstáculos, o fazem operar no modo perímetro, novas estratégias para o roteamento de pacotes para lidar com esses problemas foram propostas. E o protocolo GSR [37] é uma dessas propostas que, por um lado lida com a alta mobilidade dos nós existentes nas VANETs e por outro lado lida com o *layout* das estradas para a descoberta de rotas.

Os nós de origem utilizam um serviço de localização *Reactive Location Service* (RLS) [40] que é responsável por adquirir a posição dos nós de destino. Para determinar o caminho mais curto entre o nó de origem e o nó de destino é utilizado o algoritmo de *Dijkstra* [41] que busca encontrar qual o caminho mais curto entre dois pontos. Cada um dos cruzamentos do caminho origem-destino, que o pacote atravessa, é adicionado ao cabeçalho do pacote, assim como é realizado no protocolo DSR [3].

Assim como no GPSR, todo nó envia, de tempos em tempos, *beacons* que contêm a sua localização e a sua identificação de forma a possibilitar com que cada nó vizinho construa uma tabela de vizinhos localizados a um salto de distância. Dessa forma um algoritmo *greedy* é utilizado para escolher o vizinho mais próximo do cruzamento, que será o próximo nó de destino. A cada cruzamento alcançado, a identificação deste cruzamento é excluída do cabeçalho de pacote e a posição do próximo cruzamento é utilizada como o próximo destino até que o nó de destino seja alcançado.

Alguns estudos mostram que o GSR consegue ter melhor desempenho no consumo de banda, na taxa de entrega de pacotes e menor atraso em comparação com os protocolos AODV e DSR [42]. Porém, este protocolo não leva em consideração se a densidade da rede é baixa de forma a encaminhar os pacotes, o

que dificulta o processo de encontrar uma conexão fim-a-fim ao longo da rota pré-selecionada.

### 3.2.3 – A-STAR

O protocolo *Anchor-Based Street and Traffic Aware Routing* (A-STAR) [38] não utiliza a abordagem *greedy*. Na verdade utiliza uma abordagem totalmente diferente. O protocolo A-STAR se baseia nas rotas dos ônibus, pois considera que os ônibus trafegam pelas regiões mais importantes da cidade e, logo essas rotas possuem uma probabilidade maior de ter veículos para encaminhar pacotes, e entregar as mensagens.

Conforme o próprio nome do protocolo indica, o algoritmo do A-STAR utiliza um encaminhamento baseado em âncoras, que consiste na ação de inserir uma sequência de pontos geográficos no cabeçalho do pacote, pelo qual o pacote percorrerá para atingir o seu destino. Tanto o A-STAR quanto o GSR utilizam mapas estáticos das ruas a fim de encaminhar os pacotes de forma a computar o número de cruzamentos. Porém, o protocolo A-STAR usa a informação das rotas dos ônibus, pois parte-se da suposição de que ruas maiores, tendem a ter maior número de veículos e número maior de rotas de ônibus e, portanto, maior conectividade. Com esta informação do tráfego, o protocolo A-STAR associa um peso a uma determinada via, onde quanto maior o tráfego, menor o peso e quanto menor o tráfego, maior o peso. Tal informação é utilizada para definir a rota de âncoras de menor custo, pelo qual o pacote passará. O menor custo é calculado usando o algoritmo de *Dijkstra* [41].

Um pacote pode atingir o máximo local e o A-STAR calcula uma nova rota de âncoras a partir do nó em que o máximo local foi atingido e a nova rota é reescrita no cabeçalho do pacote. Contudo, é necessário evitar com que outros pacotes também alcancem o máximo local, e por isso a rua a qual o máximo local foi atingido é marcada como fora de serviço e pacotes do tipo *piggy-backing* recebem tal informação. Informação esta, que caso um nó receba um pacote *piggy-backing*, irá

atualizar o seu mapa local, evitando a utilização das rotas que estão marcadas como “fora de serviço”. Tais rotas estarão com o *status* fora de serviço durante um tempo determinado.

### 3.2.4 – GyTAR

O protocolo *Improved Greedy Traffic Aware Routing* (GyTAR) [39] é um protocolo de roteamento baseado em cruzamentos com o objetivo de encontrar rotas dentro de ambientes urbanos. Basicamente o protocolo GyTAR consiste de duas partes: (1) definir quais cruzamentos o pacote deve atravessar de forma a alcançar o destino e (2) um encaminhamento *greedy* melhorado entre dois cruzamentos.

O protocolo GyTAR considera que cada veículo na rede tem conhecimento de sua própria posição através da utilização de um GPS. Além disso, devido à existência de um serviço de localização tal como o GLS (*Grid Location Service*) [43] é possível que o nó de origem saiba onde o nó de destino está localizado.

A primeira parte do protocolo GyTAR, que consiste na escolha dos cruzamento pelos quais o pacote irá atravessar, leva em conta o nível de tráfego veicular e a distância para o destino. Os cruzamentos são pontuados de acordo com a densidade do tráfego e a distância do cruzamento até o nó de destino. O cruzamento escolhido é aquele que está geograficamente mais próximo do destino além de ter o tráfego veicular mais intenso, pois haverá veículos suficientes para prover conectividade e, portanto, encaminhar os pacotes.

A segunda parte do protocolo consiste no roteamento dos pacotes entre dois cruzamentos e, neste caso, o encaminhamento *greedy* é utilizado. Todos os nós neste protocolo possuem uma tabela de roteamento que é atualizada através de mensagens *HELLO* enviadas em tempos determinados. A tabela de roteamento dos nós possuem informações tais como posição, velocidade e direção de cada nó vizinho. Com essas informações o protocolo é capaz de prever a velocidade, direção e a última posição

do nó e assim escolher qual salto está mais próximo ao cruzamento em direção ao nó destino.

Entretanto, há o risco de um pacote alcançar um máximo local quando utiliza o *greedy*, requerendo então alguma estratégia para resolver estes casos. No caso do GyTAR, a técnica do *carry and forward* é colocada em prática em situações de máximo local. Tal técnica consiste em fazer com que o nó carregue o pacote até que ele alcance o próximo cruzamento ou que outro nó que esteja mais próximo ao cruzamento de destino esteja no seu raio de transmissão para que o pacote possa ser encaminhado.

Neste capítulo foi possível ver os tipos de protocolos baseados na topologia e também baseado na localização. Como foi visto, tanto a primeira quanto a segunda categoria de protocolos não são eficazes em lidar com a alta mobilidade e a escalabilidade existente nas VANETs. São problemas detectados: a grande quantidade de mensagens de controle geradas na rede em busca de manter as tabelas de roteamento sempre atualizadas e devido ao máximo local que contribui para um baixo desempenho. No Capítulo 4 é abordado sobre o protocolo baseado na métrica OU-Exclusivo (XOR) para redes móveis que utiliza uma abordagem totalmente diferente das anteriormente apresentadas, onde será explicado o seu funcionamento.

## 4 Roteamento baseado em OU-Exclusivo

O capítulo presente visa abordar o roteamento baseado na métrica de OU-Exclusivo (XOR) voltado para redes *ad hoc* móveis. Serão detalhadas as suas características, o seu funcionamento, e assim, fornecer informações ao leitor de forma que consiga compreender este tipo de roteamento e o protocolo proposto no Capítulo 5.

Para facilitar o entendimento do conteúdo, este capítulo está dividido em 6 partes. Na primeira é dada uma introdução sobre o protocolo XOR para redes móveis. Na segunda parte é explicado sobre o mecanismo de processo de descoberta existente no protocolo XOR. Na terceira parte é mostrado como funciona o processo de roteamento do protocolo XOR. Na quarta parte é mostrado o processo de aprendizagem utilizado pelo protocolo XOR. Na quinta parte é explanado sobre o conceito de visibilidade local. Na última parte são apresentadas as considerações finais, mostrando que o processo de geração de tabelas de roteamento do protocolo XOR em redes veiculares pode ser melhorado com a introdução dos filtros de Bloom, mostrado no Capítulo 5.

O protocolo baseado na métrica de OU-Exclusivo objetiva resolver problemas de escalabilidade apresentado pelas outras classes de protocolos de roteamento. As outras classes de protocolos de roteamento possuem uma limitação: a necessidade de terem conhecimento de toda a rede ou dos enlaces existentes entre os diversos nós da rede.

O aumento da densidade da rede faz com que o tamanho das tabelas de roteamento torne-se maior, aumentando o tempo de pesquisa em tais tabelas, e também, limitando a escalabilidade da rede. Grande parte dos protocolos utilizam mensagens de *broadcast* para conhecimento sobre todos os nós existentes em uma rede. Em um caminho inverso, os protocolos baseados na métrica XOR necessitam ter um conhecimento limitado da rede, conceito este conhecido como visibilidade local [44], que tem como objetivo a redução de mensagens de sinalização, que são necessárias para a criação das tabelas de roteamento, conseguindo assim a redução do atraso na entrega dos pacotes e um aumento na duração dos caminhos.

#### 4.1 Princípio de Roteamento XOR

O protocolo de roteamento baseado na métrica XOR existente para redes *ad hoc* móveis [27] utiliza identificadores de comprimento de  $n$ -bits. Tais identificadores podem ser gerados, por exemplo, aplicando-se uma função *hash* ao nome do nó, aos endereços IPs, aos endereços MACs ou uma chave privada. Esses identificadores são utilizados no encaminhamento dos pacotes e na organização das tabelas de roteamento dos nós, que são organizadas em  $\beta$  colunas.

Seu princípio de roteamento utiliza uma abordagem inovadora que é através da realização da operação lógica, conhecida como Ou-Exclusivo (XOR), entre dois identificadores de nós. Considere um cenário com os nós  $a$ ,  $b$  e  $c$ , a distância entre os nós  $a$  e  $b$  é representada pela equação  $d(a,b) = a \oplus b$ , onde  $d(a,a) = 0$  e  $d(a,b) > 0$ ,  $\forall a,b$ . Ao considerarmos o mesmo cenário e que um pacote é gerado pelo nó  $a$  e que o destino de tal pacote seja o nó  $c$ , tomando em consideração que o conjunto de identificadores contidos na tabela de roteamento do nó  $a$  é representado por  $Y$ , o nó  $a$  irá escolher o nó  $b \in Y$  de forma que minimize a distância lógica ao nó  $c$ , que é representada pela seguinte condição:

$$R = \underset{b \in Y}{\operatorname{argmin}} \{d(b,c)\} \quad (1)$$

A equação acima faz com que cada pacote seja encaminhado para o nó que, de fato, minimiza a distância lógica para o nó de destino. Considere um caso onde o identificador de cada nó é composto por 4 *bits*, assim tendo no máximo 32 nós na rede. Na Tabela 4.1 é mostrado um exemplo de tabela de roteamento do nó 11 (1011), que deseja encaminhar um pacote para o nó 5 (0101). Portanto, pela equação 1 o nó 11 irá encaminhar o pacote para o nó 4 (0100) como próximo salto.

**Tabela 4.1: Tabela de roteamento do nó 11 (1011)**

$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$
0000	1100	1000	1010
0100	1111		
0111			
0010			

Todo conhecimento acerca dos nós vizinhos é adicionado nas  $n$  colunas da tabela, que também são chamados de *buckets* e são representados pela condição  $\beta_i = 0 \leq i \leq n - 1$ . A cada momento que um nó  $a$  obtém conhecimento acerca de um nó  $b$ , aquele armazena a informação deste no *bucket*  $\beta_{n-1-i}$  correspondente, considerando que a variável  $i$  é o valor mais elevado que atende a condição abaixo:

$$d(a,b) \operatorname{div}^5 2^i = 1, a \neq b, 0 \leq i \leq n - 1 \quad (2)$$

Para compreender melhor como é utilizada a equação acima, vejamos um cenário que possua identificadores de 4 *bits* ( $n=4$ ). Podemos considerar o nó  $a$  com o identificador 1011 e o nó  $b$  com o identificador 0010. A distância entre eles ( $d(a,b)$ ) é obtida através da operação XOR entre os identificadores dos nós  $a$  (1011) e  $b$  (0010) cujo resultado é 1001. Em seguida tal resultado é convertido para numeração decimal de onde obtemos o valor 9 e resolvemos a equação 2, onde o maior valor de  $i$  que atende a equação acima é  $i = 3$ , assim dessa forma, chegando a conclusão de que o

<sup>5</sup> Operador aritmético que realiza uma divisão inteira, desconsiderando o resto

identificador do nó  $b$  deve ser armazenado no *bucket*  $\beta_{n-1-i} = 0$ . É possível perceber que, basicamente, o número de bits iguais da esquerda para a direita, tanto do nó  $a$  (1011) quanto do nó  $b$  (0010) define em que *bucket* o identificador do nó  $b$  será armazenado na tabela de roteamento do nó  $a$ . No caso o número de bits iguais entre os nós  $a$  e  $b$  é 0 (zero), portanto o *bucket* escolhido é o  $\beta_0$ . É verificada essa relação entre os identificadores da Tabela 4.1, onde é possível ver a quantidade de bits semelhantes entre o nó 11 (1011) e os outros nós armazenados nos *buckets* de sua tabela de roteamento. Tais formas de construção das tabelas e regras de roteamento apresentadas pelo protocolo XOR, fazem com que ele seja menos complexo, tenha conhecimento de somente  $O(n)$  nós dos  $2^n$  nós presentes na rede para que o pacote seja roteado, o que é uma grande vantagem em relação aos outros protocolos de roteamento já vistos anteriormente.

Além disso, o mecanismo de roteamento do protocolo XOR leva em consideração um fator  $K$ , que define a quantidade de informação necessária para preenchimento dos *buckets* das tabelas de roteamento dos nós. Considerando que um *bucket*  $\beta_i$  pode possuir no máximo  $2^{n-1-i}$  nós, se  $K > 2^{n-1-i}$ , limitamos  $K$  para tal *bucket* segundo a equação  $K = 2^{n-1-i}$ ,  $0 \leq i \leq n - 1$ . Então, é possível perceber que, ao ser alterado o valor de  $K$ , podemos ter um conhecimento mais amplo da rede ou um menor conhecimento da rede. Então, um bom número de vizinhos em cada tabela de roteamento deveria atender a seguinte equação:

$$N = \sum_{i=0}^{n-1} \min(K, 2^{n-1-i}) \quad (3)$$

Existem duas maneiras distintas para que um nó descubra outro nó na rede: o primeiro processo é chamado de processo de descoberta. O processo de descoberta tem como objetivo descobrir os nós que estão a um salto de distância, nós estes que também são chamados de vizinhos físicos. Estes nós são armazenados nos *buckets* segundo o cálculo da equação 2. A outra maneira de descobrimento de nós é chamada de processo de aprendizagem no qual, utilizando-se de mensagens *QUERY*



e *RESPONSE*, os nós capturam informações contidas nestes pacotes que passam por eles, de forma a adicionar novos nós que sejam necessários em seus *buckets* para que atendam o valor K definido. Os nós que não estão a um salto de distância, são chamados de vizinhos virtuais.

#### 4.2 Processo de Descoberta

No processo de descoberta, os nós da rede procuram, de forma ativa, outros nós para preencher os seus *buckets* da sua tabela de roteamento. Os nós da rede, para descobrirem seus vizinhos por este processo, transmitem mensagens *HELLO* em modo *broadcast* com uma frequência de  $f$  Hz para aqueles nós que estão no seu raio de comunicação (nós localizados a um salto). Dessa forma, todo nó terá conhecimento dos vizinhos que o circundam e irá preencher os seus *buckets* na sua tabela de roteamento. Caso um nó tenha ainda *buckets* a serem preenchidos, o nó irá procurar ativamente por vizinhos que preencham tais *buckets*. Considerando que tais vizinhos já estão, no mínimo, em dois saltos de distância, portanto, tais vizinhos são chamados de vizinhos virtuais. Na Figura 4.1 é mostrado um fluxograma de como é realizado o processo de envio das mensagens para a descoberta de nós do protocolo XOR.

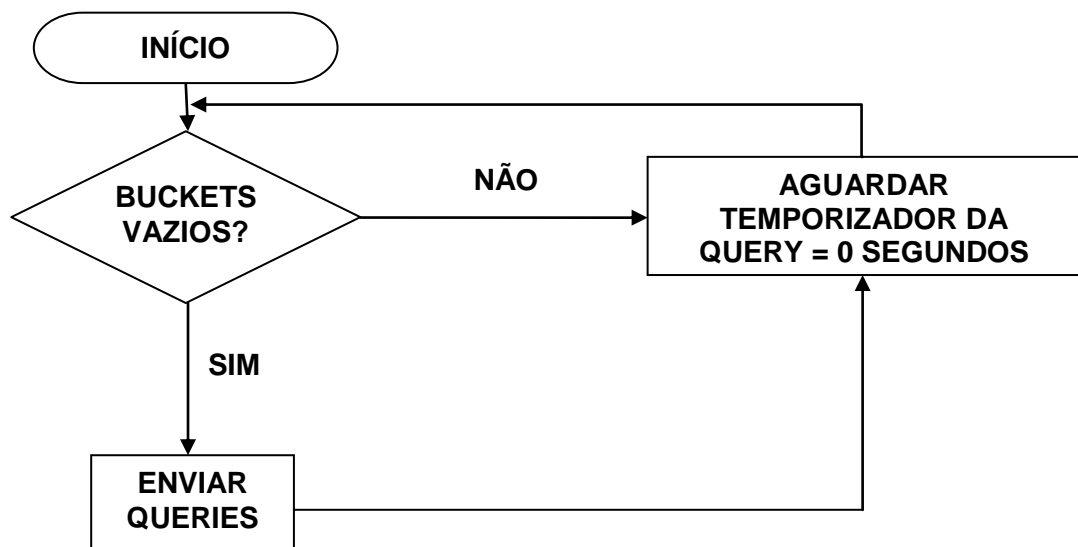


Figura 4.1 - Processo de envio de mensagens QUERY no protocolo XOR

Inicialmente o nó transmite uma mensagem do tipo *QUERY*, aos nós vizinhos físicos, que são aqueles localizados a um salto de distância. Nestas mensagens estão os *buckets* que ainda não estão completamente preenchidos. Em seguida, os nós vizinhos, ao receberem esta *QUERY*, irão verificar quais *buckets* do nó remetente da *QUERY* ainda precisam ser preenchidos e verificam em seus *buckets* se algum ou alguns dos nós armazenados neles atendem ao pedido do nó originador da *QUERY*. Caso o nó requisitado tenha algum nó ou nós que atendam ao pedido do nó originador da *QUERY*, o nó requisitado retorna tal informação em uma mensagem *RESPONSE* ao nó originador da *QUERY*, que irá guardar os novos vizinhos virtuais nos seus *buckets*, associando um vizinho físico pelo qual recebeu a mensagem *RESPONSE*. Tal processo, utilizado neste trabalho, é repetido a cada *QUERY\_Timeout* ( $T_Q$ ) = 5 segundos, mas este valor pode ser alterado. Caso exista a necessidade de inserir novos vizinhos em *buckets* que ainda precisam de informação, o nó irá transmitir uma mensagem *QUERY*, perguntando aos novos vizinhos por nós que ainda possam ser inseridos nesses *buckets* vazios.

A alta mobilidade existente nas VANETs faz com que cada nó fique armazenado no *bucket* por  $T_\beta = 5$  segundos. Além disso, para evitar um maior tráfego de sinalização e tempo de preenchimento da tabela de rotas, as mensagens de *QUERY* têm o seu número de saltos limitados, por um parâmetro chamado de *HOP\_QUERY* ( $H$ ), que é variado neste trabalho de 2 a 5.

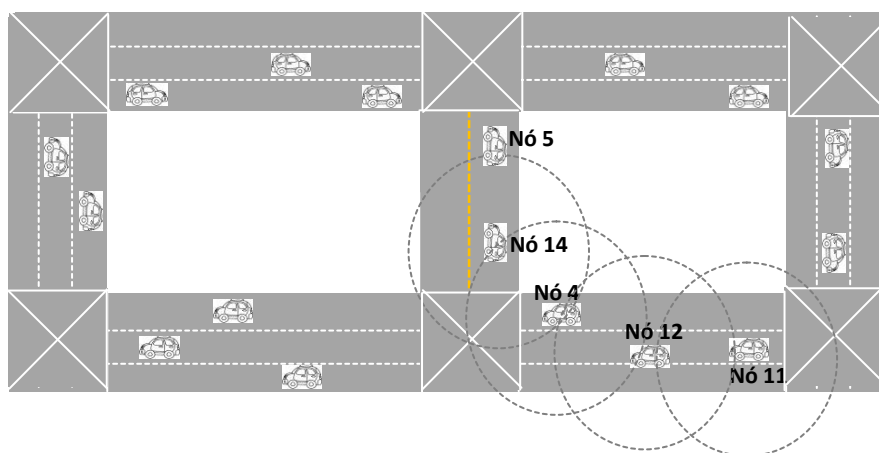
### 4.3 Processo de Roteamento

Neste processo do protocolo XOR, quando um nó deseja enviar um pacote para outro nó, é utilizada a regra de encaminhamento abaixo e este processo é dividido em algumas fases. Para efeitos de entendimento da equação abaixo  $id$  representa cada identificador presente no *bucket*  $\beta_\alpha$ .

$$n_R = \underset{id \in \beta_\alpha}{\operatorname{argmin}} \{d(id, b)\}, \quad (4)$$

Inicialmente é necessário identificar qual será o *bucket* utilizado para roteamento dos pacotes. Para isso, a equação 2 é utilizada. Depois de definido o *bucket*  $\beta_\alpha$ , o nó que está dentro deste *bucket*  $\beta_\alpha$  que melhor minimiza a distância lógica até o nó destino é escolhido como o próximo nó a receber o pacote.

Na Figura 4.2 uma exemplificação de como é realizado o processo de roteamento de pacotes no protocolo XOR. O nó 11, cuja tabela de roteamento está representada na Tabela 4.1, deseja enviar um pacote para o nó 5. Primeiramente o nó 11 irá verificar qual *bucket* será escolhido, através da resolução da equação 2. Resolvendo a equação, o *bucket* escolhido é o  $\beta_0$ . Em seguida é verificado qual o próximo salto a ser escolhido aplicando a equação número 4. Dessa forma, é descoberto o vizinho virtual 4, e assim, o pacote é roteado para o vizinho físico associado, que é o nó 12. Ao receber o pacote, o nó 12 realizará a operação XOR para definir qual o *bucket* a ser pesquisado em busca do próximo nó a ser escolhido. Levando em consideração a forma com que as tabelas de roteamento são construídas, o nó 12 deverá ter o nó 4 inserido em sua tabela, assim encaminhando o pacote para o nó 4. O nó 4, ao receber o pacote, definirá novamente o *bucket* a ser pesquisado e neste *bucket* é descoberto o nó 5 (nó de destino), que possui como vizinho físico o nó 14. Dessa forma, é só encaminhar o pacote para o nó 14, que em seguida, irá entregar diretamente ao nó 5 (vizinho físico).



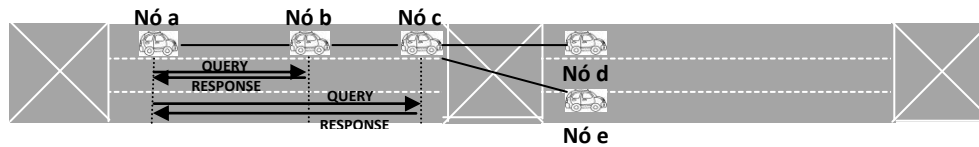
#### Relação de vizinhança dos nós:

- Nó 11 <---> Nó 12 – Vizinhos físicos
- Nó 12 <---> Nó 4 – Vizinhos físicos
- Nó 4 <---> Nó 14 – Vizinhos físicos
- Nó 14 <---> Nó 5 – Vizinhos físicos
- Nó 11 <---> Nó 4 – Vizinhos virtuais
- Nó 4 <---> Nó 5 – Vizinhos virtuais

**Figura 4.2 - Exemplo de encaminhamento do protocolo XOR**

#### 4.4 Processo de Aprendizagem

O processo de aprendizagem é responsável por obter as informações sobre os outros nós existentes na rede, sem custo algum. Graças ao mecanismo de obter informações dos pacotes que atravessam os nós, é possível que tais nós obtenham informações sobre os outros nós. Na Figura 4.3 é mostrado um exemplo do funcionamento do processo de aprendizagem onde o nó *a* precisa preencher os seus *buckets*. Primeiramente é enviada uma mensagem *QUERY* para o único vizinho que tem em sua tabela de encaminhamento, no caso o nó *b*. O nó *b* envia uma mensagem *RESPONSE* para o nó *a* informando da existência do nó *c*. Quando o nó *a* envia a mensagem *QUERY* para o nó *c*, o nó *c* aprende sobre a existência do nó *a*. Quando o nó *c* envia a mensagem *RESPONSE* para o nó *a*, tal mensagem passa também pelo nó *b*, que, de forma passiva, acaba aprendendo sobre a existência dos nós *d* e *e*.



**Figura 4.3 - Exemplo do processo de aprendizagem**

Sempre que há o recebimento de uma mensagem *QUERY* ou *RESPONSE*, está acontecendo o processo de aprendizagem. Todo nó que recebe tais mensagens verifica, dentro do corpo da mensagem, todos os identificadores ali contidos e inserindo-os em sua tabela de roteamento assim como os vizinhos físicos associados a tais identificadores. Dessa forma são disponibilizadas mais informações nas tabelas de roteamento com custo de aprendizagem praticamente zero, pois todo o processo acontece de forma passiva, e cada nó obtém mais informações para completar os *buckets* da tabela de roteamento.

Tendo mais informações na sua tabela de roteamento, conseqüentemente é maior o conhecimento sobre os nós existentes na rede, aumentando a capacidade de rotear pacotes para novos destinos.

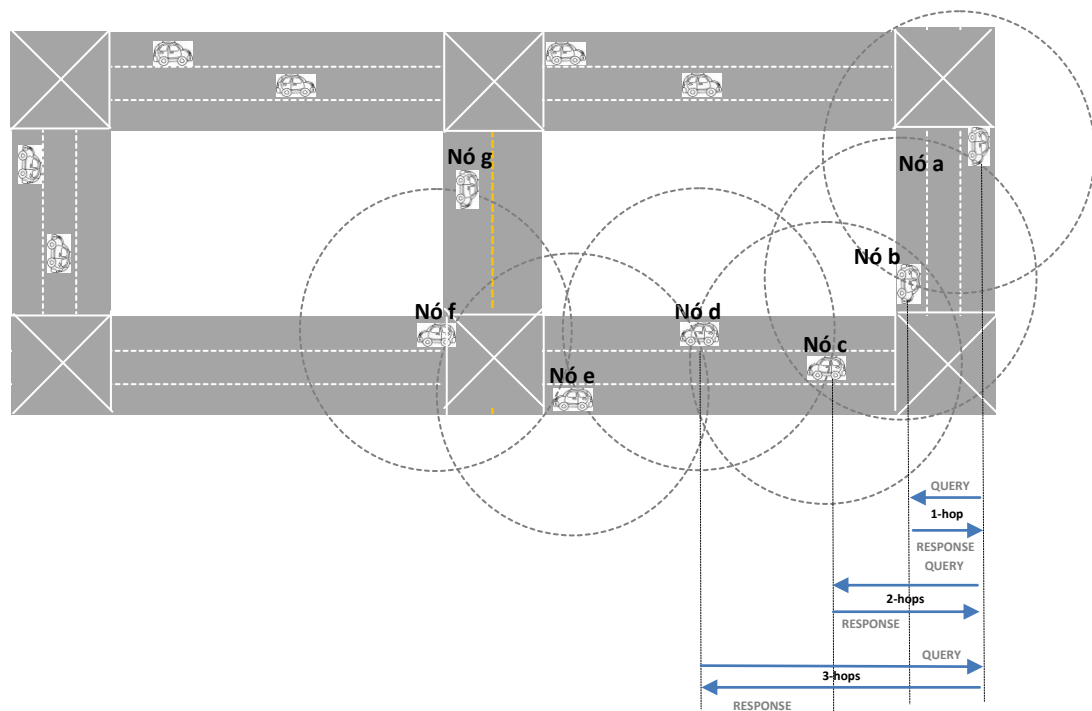
#### **4.5 Conceito de Visibilidade Local**

Com o objetivo de reduzir a quantidade de mensagens de sinalização geradas para criação das tabelas de roteamento e alcançar estabilidade na rede, é proposto o conceito de visibilidade local. Tal conceito possui como característica dar prioridade em adicionar nas tabelas de roteamento dos nós os seus vizinhos físicos mais próximos. Por isso, em uma dada mensagem *QUERY* gerada por um nó *a* para um nó *b*, o nó *b* dará prioridade em inserir na sua mensagem *RESPONSE* não somente os nós que de fato atendam aos *buckets* ainda vazios do nó *a*, mas que também estejam fisicamente mais próximos, em número de saltos.

Para melhor entendimento do conceito apresentado, vamos imaginar que um nó *b* recebe uma mensagem *QUERY* do nó *a* pedindo por identificadores que preencham o seu  $\beta_0$ . Considere que o nó *b* tem informação acerca de dois nós que

podem preencher  $\beta_0$ . O primeiro nó é o nó *c*, localizado a 1 salto de distância do nó *b* e o segundo é o nó *d*, localizado a 2 saltos de distância de *b*. O conceito de visibilidade local faz com que o nó *b* gere uma mensagem *RESPONSE* com o nó que está fisicamente mais próximo (nó *c*) que está localizado a 1 salto de distância. Dessa forma, o conceito de visibilidade local busca gerenciar o atraso fim-a-fim e dar uma maior estabilidade à rede.

Além do conceito de priorizar a inserção de nós fisicamente mais próximos nas mensagens *RESPONSE*, o conceito de visibilidade local apresenta um mecanismo chamado *HOP\_QUERY (H)*, que permite limitar a quantidade de saltos da mensagem *QUERY* a partir do nó originador da *QUERY*. Isso é feito com o objetivo de evitar inundação na rede, que gera maior consumo de banda e degrada o desempenho da rede. Na Figura 4.4 é mostrada uma situação em que a mensagem *QUERY* é limitada a 3 saltos. Mas mesmo com este mecanismo de controle do alcance da *QUERY* controlando a troca de mensagens de sinalização durante o processo de descoberta, o protocolo não impede que nós fora do alcance da *QUERY* sejam adicionados nas tabelas de roteamento. Na Figura 4.4, por exemplo, o nó *a* pode enviar uma mensagem *QUERY* somente até o nó *d*, que está localizado a 3 saltos de distância. Mas é possível que o nó *a* receba informações acerca dos nós *e*, *f* e *g*, que estão fora do alcance da mensagem *QUERY* gerada pelo nó *a*.



**Figura 4.4 - Mensagem *QUERY* limitada a 3 saltos.**

#### 4.6 Considerações Finais

O mecanismo de controle do número de saltos da *QUERY* e o conhecimento limitado da rede são fatores inovadores, onde o processo de adição dos identificadores das tabelas de roteamento é melhorado.

Contudo, conforme é aumentado o número de saltos da *QUERY*, consoante ao aumento no número de nós, faz com que seja gerada uma quantidade excessiva de mensagens de controle pelo protocolo XOR, em busca de informações que preencha os *buckets* vazios. Isto faz com que o atraso aumente, o que também pode impactar no desempenho geral do protocolo.

Para conseguir contornar tal situação, no próximo capítulo é proposto o protocolo XORBF, a principal contribuição deste trabalho, que consiste em utilizar a técnica dos filtros de Bloom nas mensagens de *QUERY* do protocolo XOR, de forma a reduzir a quantidade de mensagens geradas e otimizar o preenchimento das tabelas de roteamento. Em especial, esta proposta apresenta melhorias significativas nos

cenários com maior densidade de nós e com maior número de saltos da mensagem de *QUERY*.



## 5 Protocolo Proposto

Neste capítulo é apresentado o protocolo XORBF que utiliza filtros de Bloom nas mensagens *QUERY* do protocolo baseado em OU-Exclusivo (XOR). Conforme mencionado anteriormente, o cenário urbano possui uma série de questões complexas a serem investigadas, sendo o foco deste trabalho. Neste capítulo serão detalhadas as características do protocolo XORBF, apresentando todas as mudanças ocorridas no protocolo XOR para redes móveis com a utilização dos filtros de Bloom.

O capítulo está organizado em 5 partes. Na primeira parte são apresentados os filtros de Bloom tais como a sua estrutura, o seu funcionamento, entre outras características. Na segunda parte, é explicado como é realizado o cálculo do falso positivo nos filtros de Bloom, um exemplo de utilização dos filtros de Bloom nas mensagens de *QUERY* do protocolo XOR e como os identificadores são adicionados. Na terceira parte é explicado como funciona a técnica de filtros de Bloom no protocolo baseado na métrica de OU-Exclusivo, originando assim, o protocolo XORBF. Na quarta parte é mostrado de que forma é gerada a mensagem *QUERY*. Na quinta parte é explicado como a mensagem *QUERY* é interpretada pelo nó que a recebe, para assim, gerar uma mensagem *RESPONSE* com os identificadores requisitados pelo nó originador da *QUERY*.

## 5.1 Filtros de Bloom

Burton Bloom apresentou os filtros de Bloom em 1970 [45], e inicialmente foram amplamente utilizados em aplicações voltadas para banco de dados. Na área de redes de computadores, os filtros de Bloom chamavam pouca atenção dos pesquisadores até que, em 1995, devido à popularização da Internet, os filtros de Bloom ganharam maior visibilidade dos pesquisadores e da indústria desta área [46]. Aplicações como redes *peer-to-peer*, roteamento de pacotes, medição de dados e sistemas de checagem são exemplos de aplicações que podem se beneficiar do uso dos filtros de Bloom.

Os filtros de Bloom são estruturas de dados probabilísticas que possuem como principal vantagem frente a outras técnicas existentes a economia de espaço, por causa da sua estrutura compacta para armazenamento de dados e que elementos inseridos dentro de sua estrutura sejam consultados. Porém, as consultas aos dados armazenados podem resultar em falsos positivos, isto é, um dado elemento ser dito como presente em um conjunto de elementos, mas que na verdade não se encontra inserido naquele conjunto. Outra característica dos filtros de Bloom é que eles não resultam em falsos negativos, isto é, um dado elemento é dito não estar em um conjunto de elementos, quando está presente. Portanto, os filtros de Bloom possuem características que os fazem extremamente úteis e interessantes ao trabalhar nas diversas áreas da computação onde existam listas e conjuntos. Basicamente, a utilização dos filtros de Bloom consiste em duas tarefas: adicionar elementos a um conjunto e consulta de elementos dentro de um conjunto.

Num conceito mais formal, o filtro de Bloom é um vetor de  $m$  bits que representa um conjunto  $S = \{x_1, x_2, \dots, x_p\}$  de  $p$  elementos, onde todos os bits de tal vetor são inicializados com o valor de zero. O tamanho  $m$  do vetor e o número  $h$  de funções *hash* são definidos levando em consideração a quantidade de elementos esperados a serem inseridos no filtro além da taxa de falso positivo a ser aceita. Para

cada elemento  $x$  inserido dentro do filtro, os bits  $f_i(x)$  são configurados para bit com valor 1 para  $1 \leq i \leq h$ . Falsos positivos são elementos que não fazem parte de  $S$ , porém, que o filtro retorna como se fizesse parte do conjunto  $S$ .

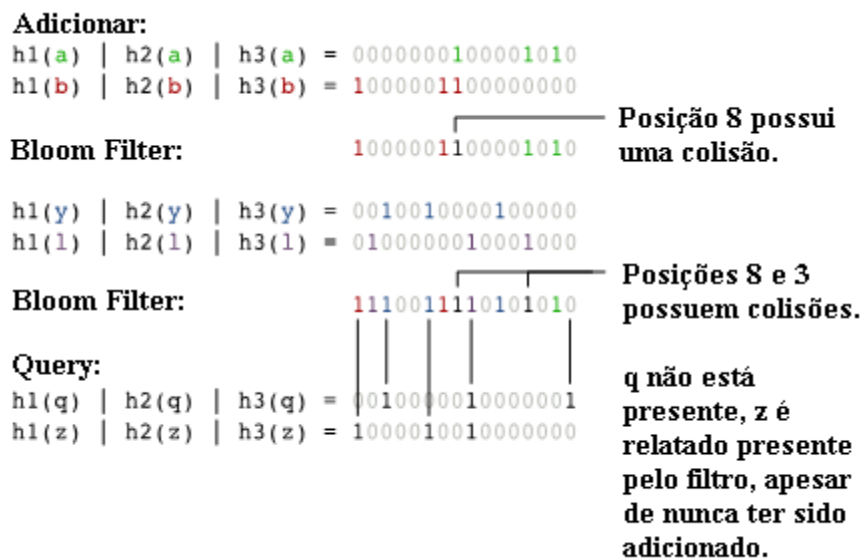
O nível de eficiência na utilização dos filtros de Bloom dependerá de três características:

- Tamanho do filtro;
- Número de funções *hash* utilizadas no filtro;
- Número de elementos adicionados ao conjunto;

Quanto maior a quantidade de elementos inseridos no filtro, maior a probabilidade de que os resultados provenientes das consultas no filtro tenham falsos positivos. Segundo [47], não importa se é utilizado um conjunto ou lista de elementos, mas em situações nas quais o espaço utilizado para armazenamento de dados seja um fator crítico e o efeito dos falsos positivos possa ser mitigado, o uso dos filtros de Bloom deveria ser considerado. Basicamente existem duas formas de reduzir as chances de falsos positivos: alterar o número de funções *hash* ( $h$ ) ou aumentar o tamanho do filtro de Bloom ( $m$ ).

Na Figura 5.1, é visto um exemplo de funcionamento de um filtro de Bloom, que envolve operações de adicionar e consultar elementos. Em tal exemplo, o filtro de Bloom é um vetor de 16 bits (*bit* 0 iniciando à direita), onde são utilizadas 3 funções *hash* ( $h_1$ ,  $h_2$  e  $h_3$ ). Lembrando que o filtro de Bloom no início do seu funcionamento está vazio, isto é, todos os bits valem zero. Depois de adicionar os elementos  $a$  e  $b$ , o filtro de Bloom possui o valor de bit alterado de zero para um nas posições 1, 3, 8, 9 e 15 e que é possível perceber que na posição 8, os elementos  $a$  e  $b$  possuem um bit em comum, caracterizando uma colisão. Após isso, ainda são adicionados os elementos  $y$  e  $l$  onde é possível ver que os bits nas posições 1, 3, 5, 7, 8, 9, 10, 13, 14 e 15 agora possuem o valor de um. Ao consultarmos pelos elementos  $q$  e  $z$ , as mesmas funções *hash* são utilizadas e os bits referentes aos elementos  $q$  e  $z$  em cada posição são verificados. Caso os três bits para um dado elemento estejam com valor

de um, assume-se que tal elemento está presente. No caso do elemento  $q$  é possível perceber que um dos bits (posição 0), está com o valor de zero, logo, sendo possível afirmar que tal elemento não está presente no filtro de Bloom. Porém, o elemento  $z$  é considerado como estando presente, pois os bits com valor um correspondentes ao elemento  $z$  correspondem com o filtro de Bloom. Contudo, é possível ver que este é um caso de falso positivo, pois os bits que correspondem ao elemento  $z$ , nas posições 7, 10 e 15, na verdade foram alterados seus valores quando os elementos  $y$ ,  $l$  e  $b$  foram adicionados ao filtro de Bloom.



**Figura 5.1 - Exemplo de adição e consulta de elementos usando filtro de Bloom [42].**

## 5.2 Cálculo da taxa de Falsos Positivos

Segundo [45], ao considerarmos que  $m$  é o comprimento (em número de bits) do filtro de Bloom,  $p$  é o número de identificadores dos nós inseridos no filtro e  $h$  o número de funções *hash* utilizadas, a probabilidade de falso positivo pode ser calculada conforme mostrada nesta seção.

Ao inserirmos um elemento dentro do filtro, a probabilidade de que um certo bit não seja modificado para 1 por uma função *hash* é:

$$1 - \frac{1}{m} \tag{5}$$

Existem  $h$  funções *hash* ( $h_1, h_2, \dots, h_k$ ) e a probabilidade de que nenhuma dessas funções *hash* tenham alterado um dado bit de zero para 1 é:

$$\left(1 - \frac{1}{m}\right)^h \quad (6)$$

Conforme vamos inserindo  $p$  identificadores ao filtro de Bloom, a probabilidade de que um dado bit no filtro  $m$  ainda tenha o valor de zero é:

$$\left(1 - \frac{1}{m}\right)^{hp} \quad (7)$$

Sendo assim, a probabilidade de que um dado bit em  $m$  seja 1 é:

$$1 - \left(1 - \frac{1}{m}\right)^{hp} \quad (8)$$

Porém, existe uma possibilidade de que falsos positivos aconteçam, isto é, de que identificadores que não foram inseridos no filtro apareçam como presentes. A probabilidade de que isso aconteça quando o elemento não faz parte do conjunto é:

$$\left(1 - \left(1 - \frac{1}{m}\right)^{hp}\right)^h \quad (9)$$

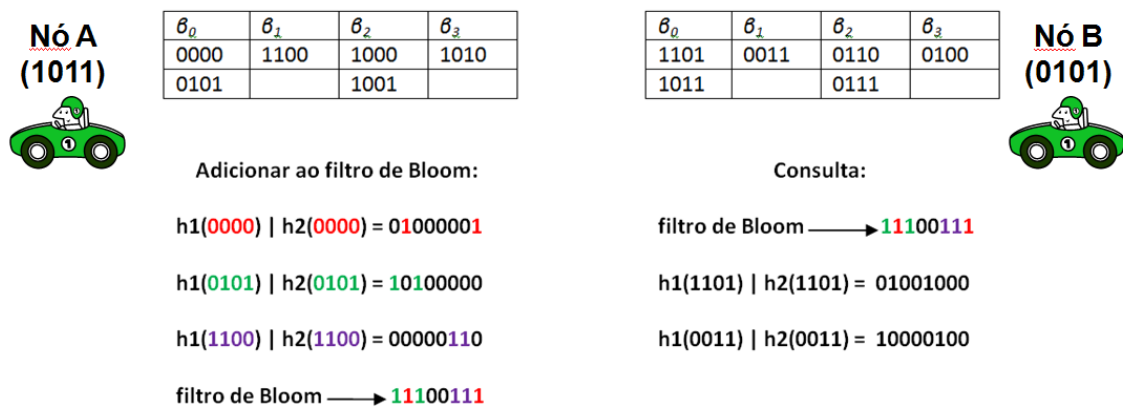
Portanto, pode ser percebido que quanto menor o número de identificadores ( $n$ ) de um dado conjunto e quanto maior o tamanho do filtro ( $m$ ), menor a probabilidade de falsos positivos. Assim sendo, para  $p = 15$ ,  $m = 128$  e  $h = 3$ , a probabilidade de falso positivo do filtro é igual a 0,0262. Percebe-se que a utilização dos filtros de Bloom traz alguns benefícios tais como:

- Vários identificadores podem ser transmitidos simultaneamente sem aumentar o tamanho do filtro;
- Qualquer identificador, independente do seu tamanho, pode ser adicionado ao, permitindo assim uma enorme flexibilidade.

### 5.3 Utilizando filtros de Bloom no protocolo baseado em XOR

Em redes densas com um número elevado de nós, como as VANETs urbanas, existe uma chance de ocorrer o falso positivo, devido ao elevado número de identificadores de nós que podem vir a ser inseridos nos filtros de Bloom. A taxa de

falso positivo pode ser minimizada ao analisar o filtro e então configurar os diferentes parâmetros do filtro (número de funções *hash*, taxa de falso positivo aceitável e o número de identificadores a serem inseridos no filtro, o tamanho do filtro) em busca de um valor ótimo. A Figura 5.2 mostra um exemplo da utilização do filtro de Bloom nas mensagens *QUERY* do protocolo XOR.



**Figura 5.2 – Exemplo da utilização do filtro de Bloom na mensagem *QUERY***

Na Figura 5.2 é mostrado um exemplo da comunicação entre dois nós a e b com identificadores 1011 e 0101, respectivamente. Um filtro de Bloom de 8 bits, onde cada *bit* representa uma posição iniciando da esquerda, com valor de 0 (zero) até o valor de 7, é utilizado para armazenar os identificadores do nó *a*. É considerado que três identificadores (0000, 0101 e 1100) são adicionados no filtro de Bloom utilizando duas funções *hash*. O nó *b* recebe a mensagem *QUERY* com o filtro de Bloom e verifica o identificador 1101, presente em sua tabela de roteamento, retornando corretamente que tal identificador não faz parte do filtro de Bloom, pois ao aplicar as mesmas funções *hash* no identificador 1101, é possível ver que na posição 4 o bit está com valor 1, e que no filtro de Bloom a posição 4 está com o valor de 0, significando que tal identificador não está presente. Porém, o filtro de Bloom retorna que o identificador 0011 está presente, mesmo ele, não estando presente no filtro de Bloom, pois ao aplicar as funções *hash* no identificador 0011, retorna nas posições 0 e 5 os *bits* com valor 1 e que no filtro de Bloom tais posições também possuem o valor 1.

Nos cenários das VANETs, onde a mobilidade dos nós é um fator crítico, e que as tabelas de roteamento são constantemente atualizadas, especialmente nas situações em que exista uma grande quantidade de nós, o filtro de Bloom, devido a sua estrutura compacta, consegue otimizar o preenchimento da tabela de roteamento, obtendo maior número de identificadores consoante redução significativa da quantidade de informação gerada pelas mensagens *QUERY*.

#### **5.4 A mensagem *QUERY***

Uma das características do protocolo XOR é o controle do número de saltos da mensagem *QUERY* (fator *H*). Quanto maior a quantidade de informação gerada em busca de identificadores para preenchimento dos *buckets*, maior é a carga imposta à rede. Este método busca preencher os *buckets* que ainda não estão completamente preenchidos em um cenário de alta mobilidade e, por isso, não é muito vantajoso, pois muitas mensagens de sinalização (*QUERIES* e *RESPONSES*) são geradas na rede. Isto reduz, em cenários com alta densidade de nós, a capacidade e a escalabilidade da rede.

Para obter redução na geração de mensagens de sinalização é utilizada a técnica dos filtros de Bloom nas mensagens *QUERY* geradas pelo protocolo XOR. Basicamente, a ideia é introduzir nas mensagens *QUERY* enviadas para os vizinhos o filtro de Bloom com os identificadores dos *buckets* que ainda não estão preenchidos reduzindo a quantidade de informação a ser enviada na *QUERY*.

Inicialmente o nó gerador da *QUERY* irá verificar quais *buckets* ainda estão precisando de informações, isto é, se faltam informações de forma a completar a tabela de roteamento. No Algoritmo 5.1 é possível ver de que forma ocorre a geração da mensagem da *QUERY* já com a estrutura do filtro de Bloom incluída, de forma a conseguir preencher os *buckets* ainda não preenchidos completamente. Os parâmetros de entrada são os identificadores existentes nos *buckets* do nó de origem

da *QUERY* (*ids\_buckets*). Os parâmetros de saída são o nó que está realizando a consulta ( $N_{orig}$ ), o vetor de nós que conterá o filtro de Bloom ( $B_i$ ) assim como os *buckets* ainda não completamente preenchidos (*buckets\_necessarios*). A operação é dividida da seguinte forma:

- **Passo 1:** É realizada uma consulta aos *buckets* vazios do nó que vai realizar a consulta ( $N_{orig}$ );
- **Passo 2:** Cada id existente nas entradas dos *buckets* vazios são consultados
- **Passo 3:** É feito *loop* em todas as funções *hash* (*h*). São aplicadas funções *hash* nos identificadores encontrados dentro das entradas dos *buckets* vazios. Cada função *hash* aplicada em um identificador é adicionado em uma variável *i*, que é a posição do vetor do filtro de Bloom que terá o seu valor alterado de 0 (zero) para 1 (um);
- **Passo 4:** Em seguida, uma determinada posição do vetor  $B_i$ , que é o filtro de Bloom terá o valor modificado de 0 (zero) para 1 (um);
- **Passo 5:** O nó de origem adiciona na mensagem *QUERY* um vetor de nós contendo o filtro de Bloom ( $B_i$ ), quais *buckets* ainda não estão completos em busca de ids que preencham os *buckets* ainda não totalmente preenchidos (*buckets\_necessarios*), além do nó de origem ( $N_{orig}$ ) que serão enviados posteriormente para os vizinhos.

```

Parâmetros de entrada: ids_buckets
Parâmetros de saída: Bi, Norig, buckets_necessarios
1 for cada_bucket_vazio do                                /* Passo 1 */
2   for cada_id_do_bucket_vazio do                       /* Passo 2 */
3     for j:1..h do                                       /* Passo 3 */
4       i = fi(id_do_bucket)
5       if Bi == 0 then                                    /* Passo 4 */
6         Bi = 1
7       end
8     end
9   end
10 end
11 QUERY.enviar(Bi, Norig, buckets_necessarios)      /* Passo 5 */

```

**Algoritmo 5.1 – Algoritmo de preenchimento do filtro de Bloom e geração da mensagem de *QUERY*.**



A mensagem *QUERY* que já se encontra modificada com a introdução da estrutura do filtro de Bloom com os identificadores dos *buckets* que ainda não estão preenchidos é enviada para os nós vizinhos localizados a  $H$  saltos. Na Figura 5.3 é exemplificado o processo de envio de mensagens de *QUERY* com a utilização do filtro de Bloom na sua estrutura.

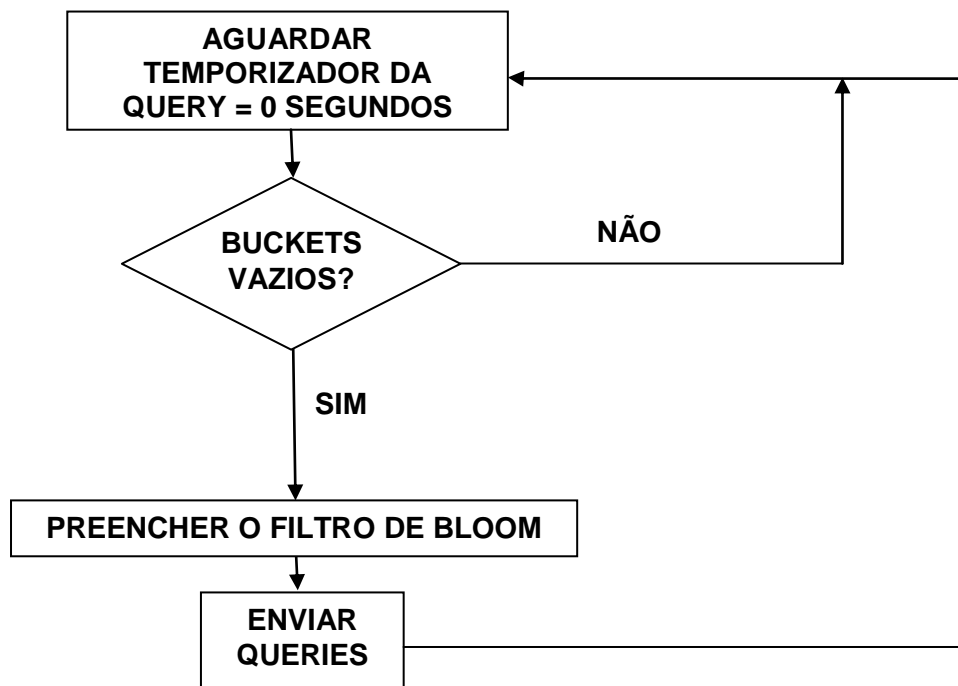


Figura 5.3 – Processo de envio da mensagem de *QUERY* com o filtro de Bloom.

### 5.5 A mensagem *RESPONSE*

Quando os vizinhos recebem a mensagem *QUERY*, eles verificam o conteúdo do filtro de Bloom para saber quais identificadores não estão contidos no filtro de Bloom e quais identificadores podem estar presentes considerando uma taxa de falso positivo. Assim que foi verificado todo o conteúdo do filtro de Bloom, o nó que recebeu a mensagem *QUERY* irá verificar em sua tabela de roteamento quais identificadores servem para o nó originador da *QUERY* de forma a preencher seus *buckets* que ainda não completamente preenchidos e assim é enviada uma mensagem *RESPONSE* (sem alteração do formato original existente no protocolo XOR) com os identificadores

viáveis para o nó originador da *QUERY*. No Algoritmo 5.2 é mostrado este processo cujos passos são descritos a seguir, em que os parâmetros de entrada são o filtro de Bloom ( $B_i$ ), o nó que realiza a pesquisa ( $N_{orig}$ ) e o os buckets que ainda não estão completamente preenchidos (*buckets\_necessarios*). Seguem os passos do Algoritmo 5.2 abaixo:

- **Passo 1:** O nó que recebeu a *QUERY* irá verificar todos os seus *buckets*;
- **Passo 2:** Cada identificador existente nas entradas dos *buckets* do nó que recebeu a *QUERY* são consultados;
- **Passo 3:** É feito um *loop* em todas as funções *hash* ( $h$ ) e enquanto o valor do bit no filtro valer 1. São aplicadas as mesmas funções *hash* nos identificadores encontrados. O filtro de Bloom recebido será checado para saber quais os *ids* que o nó de origem provavelmente possui, dada uma taxa de falso positivo. Caso um dado bit estiver com o valor de 0 (zero) é certo que ele não está presente no filtro.
- **Passo 4:** Se o valor de algum bit for 0 (zero), é certo que ele não está presente no filtro. Assim, para cada entrada dos vários *buckets* é realizada uma operação XOR entre o nó  $N_{orig}$  e o nó que está contido no *bucket* do nó que recebeu a *QUERY*. Do valor resultado é realizada uma transformação para representação binária e o cálculo do valor do prefixo;
- **Passo 5:** É realizada uma consulta pelos *buckets* necessários pelo nó que originou a *QUERY*. Em seguida é realizada uma comparação entre o prefixo do nó que está contido no *bucket* e os *buckets* requisitados. Depois, é inserido, no vetor de saída os identificadores que atendam ao nó de origem da *QUERY*;
- **Passo 6:** É enviada a mensagem *RESPONSE* com o vetor de *ids* que servem para o nó que requisitou a *QUERY*.

```

Parâmetros de entrada:  $B_i$ ,  $N_{orig}$ , buckets_necessarios
Parâmetros de saída: vetor_de_nós

1   m = 1
2   j = 1
3   for cada_bucket do
4       for cada_id_do_bucket do
5           while m==1 and j<=h do
6               i = fi(id_do_bucket)
7               if  $B_i = 0$  then
8                   m=0
9               end
10              j=j+1
11          end
12          if m==0 then
13              result_xor = XOR(id_do_bucket, Norig)
14              prefixo = calculo_do_prefixo(result_xor)
15              for buckets_necessarios do
16                  if prefixo == bucket_necessario then
17                      vetor_de_nos.inserir(id)
18                  end
19              end
20          end
21      end
22  end
23  RESPONSE.enviar(vetor_de_nos)

```

/\* Passo 1 \*/  
/\* Passo 2 \*/  
/\* Passo 3 \*/  
/\* Passo 4 \*/  
/\* Passo 5 \*/  
/\* Passo 6 \*/

**Algoritmo 5.2 – Algoritmo de verificação do filtro de Bloom e geração da mensagem de RESPONSE**

Depois de todo o processo efetuado pelo nó consultado, através da *QUERY* em busca de ids, é gerada a mensagem *RESPONSE*, sendo esta mensagem retornada para o nó requisitante.

No decorrer deste capítulo foi apresentado o protocolo XORBF. Foram abordados como ocorre a inserção dos ids no filtro de Bloom assim como o envio da *QUERY* solicitando mais identificadores para o preenchimento da tabela de roteamento do nó requisitante. Além disso, foi mostrado como o nó requisitado trata a mensagem *QUERY* de forma a retornar somente os nós necessários pelo nó requisitante. No próximo capítulo será abordada uma avaliação do protocolo XORBF comparando-o com outros protocolos de roteamento em um cenário urbano.

## **6 Análise de Desempenho**

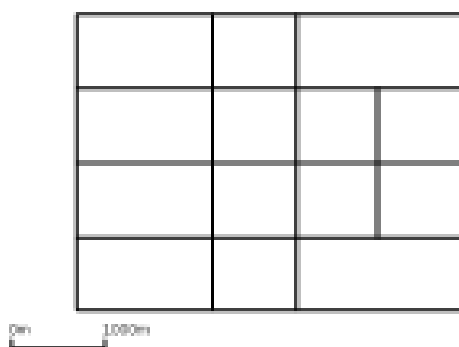
Depois de todos os conceitos apresentados, em especial o protocolo XOR e os filtros de Bloom, neste capítulo serão apresentadas as simulações realizadas e os resultados obtidos. O cenário avaliado é descrito com maiores detalhes nas seções que se seguem. Com os resultados alcançados é possível comparar o desempenho do protocolo XOR sem os filtros de Bloom, chamado daqui em diante de XOR, o protocolo XOR com os filtros de Bloom, a partir deste ponto chamado de XORBF, e os protocolos de roteamento baseados em topologia OLSR e AODV e o protocolo de roteamento geográfico GPSR. Destaca-se que uma parte desses resultados foi publicado no trabalho [48].

Este capítulo está dividido em 3 partes: na primeira parte são apresentadas as simulações, e as métricas utilizadas para avaliar o desempenho dos vários protocolos. Na segunda e terceira parte são apresentados e discutidos os resultados obtidos nas simulações.

## 6.1 Cenário de Simulação

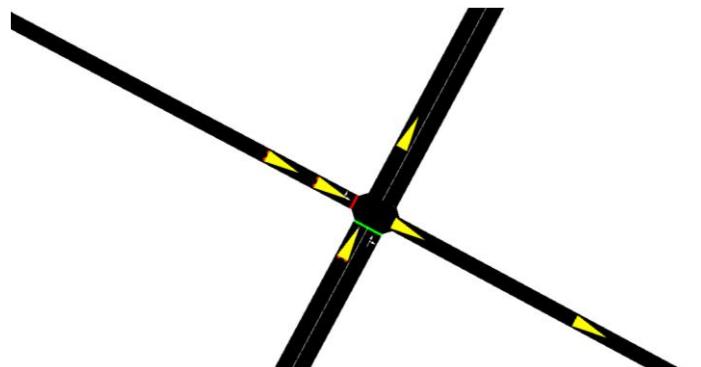
Como já foi descrito no Capítulo 4, o protocolo XOR apresenta um conceito conhecido como visibilidade local, que visa preencher os *buckets* com os nós que estão mais próximos em número de saltos. O objetivo desse protocolo é lidar com a mobilidade frequente existente nas VANETs, assim como reduzir o tráfego de mensagens de sinalização. Portanto, nas simulações do protocolo XOR foram utilizados o valor de *HOP\_QUERY* (*H*) variando de 2 a 5 de forma a avaliar o impacto do número de saltos dados por uma mensagem no preenchimento de *buckets* do nó originador da *QUERY*.

A VANET simulada foi obtida utilizando a ferramenta *TraNS* [49], que integra o gerador de mobilidade veicular SUMO [50]. O gerador de mobilidade é responsável por definir todos os parâmetros que envolvem a mobilidade dos veículos dentro de um dado cenário tais como: velocidade dos veículos, quantidade de veículos, rotas a serem percorridas, aceleração, desaceleração, tempo de mobilidade dos nós, entre outros. O cenário, como mostrado na Figura 6.1, foi definido de forma a representar um cenário urbano, com cruzamentos, semáforos presentes em cada cruzamento e vias bidirecionais com até 3 faixas. Tal cenário foi extraído da base TIGER [51], base com mapas geográficos, mantida pelo U.S Census Bureau e entretanto, esta base contém informações geográficas somente dos Estados Unidos. O cenário da Figura 6.1 possui as dimensões de 4.100 m. x 3.100 m., que é significativo ao avaliar diferentes densidades de veículos.



**Figura 6.1 - Cenário urbano utilizado na simulação.**

Na Figura 6.2, é mostrada a mobilidade dos veículos no mapa utilizado, mostrado na Figura 6,1, onde é visto a presença de semáforos, várias faixas, sentidos de circulação permitidos para os veículos, mostrando um cenário urbano realístico.



**Figura 6.2 – Parte do mapa urbano utilizado na simulação mostrando características realísticas**

Nas simulações, os pontos de origem, destino, e as suas rotas dos veículos foram definidos aleatoriamente. Cada veículo foi inserido no cenário para iniciar a sua mobilidade a cada 1 segundo, portanto no instante 0, o mapa está completamente vazio e o instante de parada de cada veículo foi definido aleatoriamente pelo gerador de mobilidade.

Foram configurados 7 cenários diferentes de mobilidade, onde o número de veículos varia de 40 a 280 veículos. Além disso, de forma a representar a velocidade dos veículos em cenários urbanos, a velocidade máxima dos veículos foi definida em 20 m/s (72km/h). Outro item configurado foi o alcance de rádio para os veículos, definido em 400 metros, pois de acordo com [52], muitas aplicações de entretenimento podem ser utilizadas com o alcance de rádio a 400 metros. Outro ponto a dar destaque são as taxas de falso positivo alcançadas em cada cenário, onde essa taxa de falso positivo foi calculada levando em consideração a quantidade média de elementos inseridos no pior caso ( $H=5$ ) para cada cenário. De forma a resumir as informações e facilitar a visualização e o entendimento, as características de simulação dos 7 cenários são mostrados na Tabela 6.1.

**Tabela 6.1: Características da simulação**

<b>Cenário</b>	<b>Número de veículos</b>	<b>Velocidade máxima (m/s)</b>	<b>Raio de alcance do rádio (m)</b>	<b>Tempo da simulação (s)</b>	<b>Taxa de falso positivo</b>
<b>Cenário 1</b>	40	20	400	437	3,26e-5
<b>Cenário 2</b>	80	20	400	424	2,79e-5
<b>Cenário 3</b>	120	20	400	561	2,37e-5
<b>Cenário 4</b>	160	20	400	598	5,82e-5
<b>Cenário 5</b>	200	20	400	556	3,69e-5
<b>Cenário 6</b>	240	20	400	566	3,19e-5
<b>Cenário 7</b>	280	20	400	614	7,36e-5

Após definidas todas as características da mobilidade veicular, foram gerados *traces* de mobilidade para cada cenário. Esses *traces* são arquivos que contém informações acerca da localização dos veículos e da sua velocidade dentro do cenário em um dado instante de tempo. O próximo passo, após gerar os arquivos contendo a mobilidade dos veículos em um cenário, é avaliar o comportamento dos protocolos de rede. Para isso, foi utilizado o simulador NS-2 na sua versão 2.33 [53], onde foi importado o arquivo de *trace* de mobilidade gerado pela ferramenta TraNS, explicada anteriormente e diversos parâmetros envolvendo detalhes da camada física, sub-camada MAC, entre outros foram configurados. Foi adotado o padrão 802.11 às taxas de 11Mbps e 2Mbps para o tráfego de dados na forma *unicast* e *broadcast*, respectivamente. O modelo de propagação utilizado foi o *TwoRayGround* [54].

Para a avaliação do desempenho dos protocolos de roteamento, foi utilizada uma aplicação responsável por gerar pedidos de encaminhamento, onde cada nó é responsável por enviar os pacotes que irão dar origem aos pedidos de encaminhamento e tentar criar caminhos para comunicação. Dois tipos de pacotes especiais chamados de *ping* e *probe* foram criados de forma a auxiliar na extração das métricas e estes pacotes guardam toda a sequência de nós pelos quais passaram.

Para gerar um caminho de comunicação, primeiramente é enviado, de forma aleatória, um pacote do tipo *ping* para um nó de destino, também definido aleatoriamente. Caso o nó de destino receba tal pacote, o caminho é considerado criado e a cada 1 segundo é enviado um pacote do tipo *probe* entre origem-destino.

Há duas situações em que o caminho é considerado quebrado e, assim, a comunicação é interrompida:

- O nó de destino parou de receber *probes*;
- A sequência de nós atravessados para a criação de caminho com o uso do pacote de *ping* tenha sido alterada, isto é, não é o caminho originalmente criado.

No caso da simulação realizada, foram enviados pacotes para criação de caminhos a uma taxa de 3 pedidos de encaminhamento por segundo (*ping*). Os pedidos de encaminhamento são inicializados a partir do centésimo segundo, pois neste instante há uma quantidade razoável de veículos no cenário. De forma a obter resultados estatisticamente mais confiáveis, foram realizadas 30 rodadas de simulações para cada cenário e para cada protocolo. A cada simulação, a semente do gerador de números aleatórios da aplicação de tráfego foi alterada para obter resultados diferentes com um intervalo de confiança de 95%.

Inicialmente foi reunido um conjunto de amostras independentes, e em seguida foi realizado um tratamento estatístico das mesmas. Da teoria da estatística, se o número de amostras ( $n$ ) é  $\geq 30$ , mesmo sem conhecermos a distribuição da população, os limites do intervalo de confiança para a média ( $\bar{X}$ ) poderão ser calculados com base na distribuição normal. Da mesma forma é utilizado o desvio padrão amostral  $s$ . Neste caso, o intervalo de confiança para a média  $\mu$  é dada pela equação:

$$\left[ \bar{X} - z_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}}; \bar{X} + z_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}} \right] \quad (10)$$

onde o intervalo de confiança estará compreendido no intervalo que é mostrado na equação com uma probabilidade de  $1-\alpha$ .



No estudo realizado para comparar os protocolos de roteamento XORBF, XOR, OLSR, AODV e GPSR, foi utilizado o parâmetro  $\alpha = 5\%$ , que garante um intervalo de confiança de 95%.

Outros parâmetros referentes ao protocolo XOR foram utilizados na simulação tais como o tempo em que o id de um nó fica armazenado na tabela de roteamento ( $T_\beta$ ) = 5 segundos e o  $QUERY\_Timeout$  ( $T_Q$ ) = 5 segundos e o fator  $K=1$ .

Para o protocolo XORBF, foram utilizados os mesmos parâmetros das variáveis  $T_\beta$ ,  $T_Q$  e  $K$  do protocolo XOR. Além disso, foram utilizadas 3 funções *hash* e o tamanho do filtro foi definido em 128 *bits* em busca de uma baixa taxa de probabilidade de falso-positivo. Além disso, foi verificada a quantidade média de elementos inseridos no filtro de Bloom para o valor do fator  $H$  mais alto ( $H=5$ ) em cada cenário sua respectiva taxa de falso positivo.

### 6.1.1 Métricas de Avaliação

A seguir serão descritas as métricas de avaliação utilizadas neste cenário:

- **Caminhos disponíveis/criados** – Um caminho disponível é a situação em que um nó de origem consegue enviar um pacote de *ping*, responsável por originar o caminho, e este é recebido pelo nó de destino. Caso haja uma comunicação entre os dois nós (origem-destino), logo é considerado um caminho disponível/criado.
- **Tempo de duração dos caminhos** – Considerando um caminho criado com sequência de nós  $z$  chamado de  $C = \{v_1, v_2, \dots, v_z\}$  em um instante  $t_1$ , a duração dos caminhos é considerado o maior intervalo de tempo  $[t_1, t_2]$  para a sequência  $C$ .
- **Comprimento dos caminhos** – Em redes *ad hoc*, o comprimento dos caminhos é representado pela quantidade de saltos que um pacote dá entre um nó de origem ( $v_1$ ) e o seu destino ( $v_z$ ). Assim, para um dado caminho  $C =$

$\{v_1, v_2, \dots, v_z\}$ , consistindo em  $z$  nós, o comprimento dos caminhos corresponde a  $z - 1$  saltos.

- **Atraso na entrega dos pacotes** - Considerando um caminho criado com sequência de nós  $z$  chamado de  $C = \{v_1, v_2, \dots, v_z\}$  onde  $v_1$  é o nó de origem e  $v_z$  é o nó de destino, o atraso do pacote é a diferença entre o tempo final  $t_f$  (o pacote chega ao nó de destino  $v_z$ ) e o tempo inicial  $t_i$  (o pacote sai do nó de origem  $v_1$ ).

### 6.1.2 Resultados das Simulações

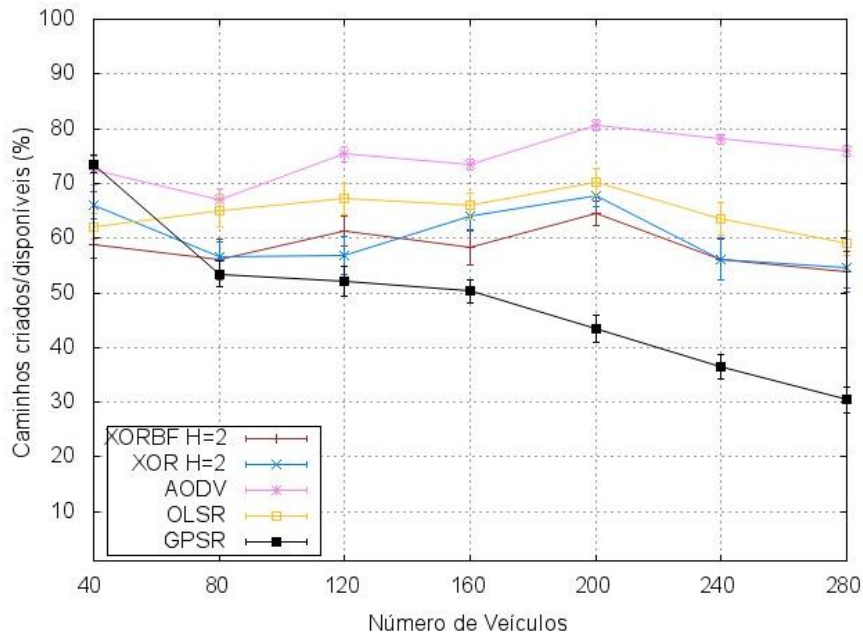
Para melhor caracterizar os protocolos XOR e XORBF, foi realizada uma comparação entre estes protocolos, para diferentes números de saltos da mensagem *QUERY* ( $H$ ), os resultados foram comparados com outros 3 protocolos, sendo 2 deles baseados na topologia (OLSR e AODV) e um protocolo de roteamento geográfico (GPSR) para as métricas definidas na seção 6.1.1.

As Figuras 6.3, 6.4, 6.5 e 6.6 mostram os resultados referentes à taxa de caminhos disponíveis/criados para diferentes protocolos quando  $H=2$ ,  $H=3$ ,  $H=4$  e  $H=5$ .

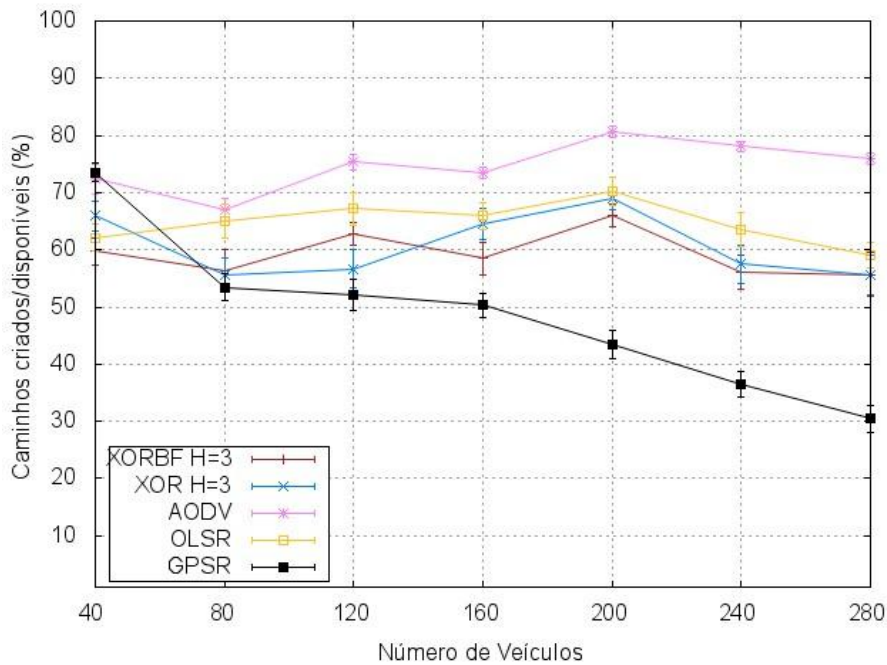
Conforme a figura 6.3 o protocolo XORBF possui um comportamento bem semelhante ao do protocolo XOR. A princípio, é possível deduzir que a implementação dos filtros de Bloom não melhora o desempenho do protocolo XORBF nos cenários com menor densidade de veículos e que no valor de  $H=2$  os valores não impactam de forma significativa os resultados.

Entretanto, como pode ser vista nas Figuras 6.4, 6.5 e 6.6, quanto maior o valor de  $H$ , maior a carga de sinalização gerada pelo protocolo XOR, em especial nos cenários com maior densidade de nós, onde a estrutura de filtros de Bloom, devido ao fato de conseguir compactar informações, consegue reduzir a carga de sinalização, otimizando o preenchimento das tabelas de roteamento dos nós e alcançando

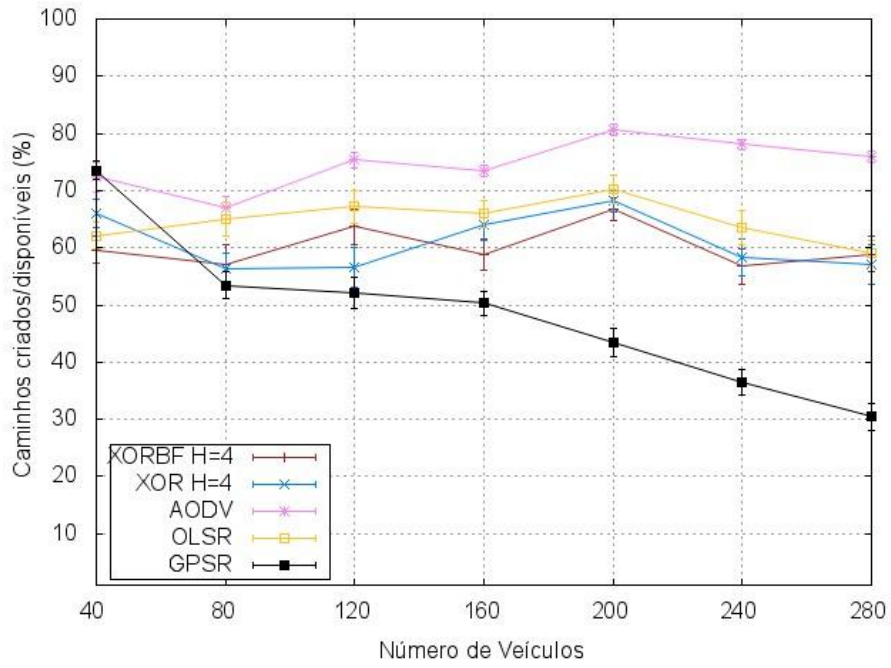
resultados melhores que o protocolo XORBF e o protocolo OLSR no cenário de 280 veículos.



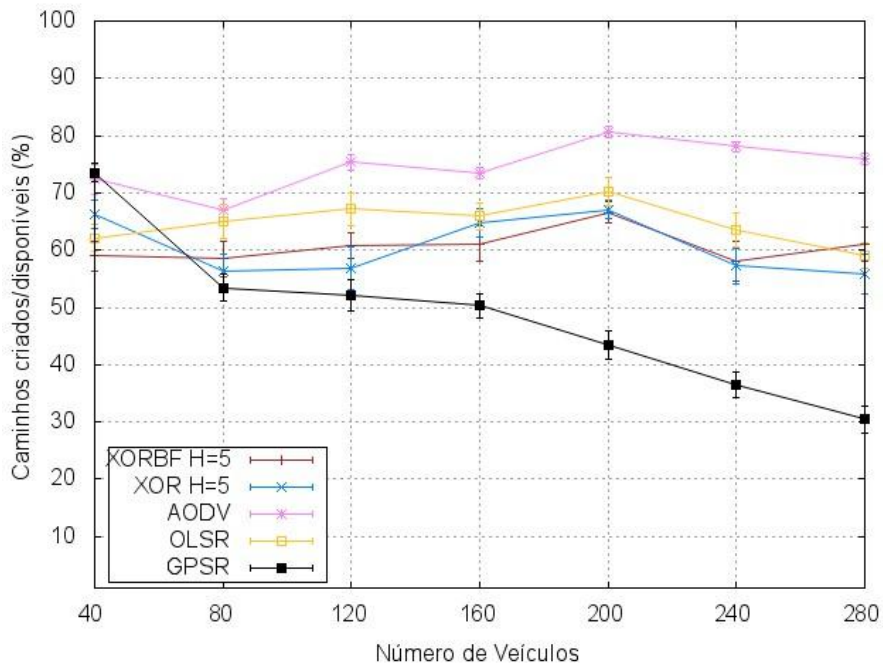
**Figura 6.3 - Taxa de caminhos disponíveis/criados pelos protocolos XOR ( $H=2$ ), XORBF ( $H=2$ ), GPSR, AODV e OLSR.**



**Figura 6.4 - Taxa de caminhos disponíveis/criados pelos protocolos XOR ( $H=3$ ), XORBF ( $H=3$ ), GPSR, AODV e OLSR.**



**Figura 6.5 - Taxa de caminhos disponíveis/criados pelos protocolos XOR ( $H=4$ ), XORBF ( $H=4$ ), GPSR, AODV e OLSR.**



**Figura 6.6 - Taxa de caminhos disponíveis/criados pelos protocolos XOR ( $H=5$ ), XORBF ( $H=5$ ), GPSR, AODV e OLSR.**

No cenário com menor densidade (40 nós) o protocolo XORBF em todas as variações de  $H$  (2 a 5) tem o seu desempenho prejudicado em torno de 11% quando comparado ao protocolo XOR com as mesmas variações de  $H$ .

No cenário com 120 nós, o protocolo XORBF possui desempenho maior que o protocolo XOR para todos os valores de  $H$ , alcançando um desempenho até 11% superior com o valor de  $H=4$ . No cenário de 200 nós é possível perceber que o protocolo XORBF e XOR possuem desempenhos praticamente iguais em todos os valores de  $H$  (2 a 5), variando no máximo a 5% de ganho para o XOR com o valor de  $H=2$ .

Já no cenário com 280 veículos, é possível ver que a variação do XORBF consegue superar até mesmo o protocolo OLSR, mostrando, inicialmente, que a técnica dos filtros de Bloom, consegue melhorar levemente as consultas por identificadores nos cenários com maior quantidade de nós, conseguindo criar mais caminhos. Assim, inicialmente é mostrado que o fato de otimizar as consultas por identificadores e, assim, obtendo um maior número de identificadores, com menor carga de controle gerada na rede, faz com que o protocolo XORBF tenha resultados superiores em cenários com maior densidade de nós avaliado neste trabalho.

Pode-se ver que o desempenho do protocolo GPSR tende a cair conforme é aumentada a quantidade de veículos, pois tal protocolo precisa trocar informações sobre a localização dos nós, de forma a manter as tabelas de roteamento dos nós sempre atualizadas e isso, faz com que haja alto volume de mensagens de sinalização, o que consome banda, congestionando a rede e resulta em aumento no número de pacotes perdidos e em baixa taxa de criação de caminhos. Além disso, o protocolo GPSR não lida com as características dos cenários urbanos, devido a alcançar um máximo local e entrar em modo de perímetro.

O protocolo XORBF alcança resultados interessantes, superando de maneira significativa o protocolo GPSR, assim como superando também o protocolo OLSR no cenário de 280 veículos. O fato de privilegiar a inserção de nós fisicamente mais

próximos, o que é interessante no cenário de alta mobilidade das VANETs, faz com que seja menor a quantidade de mensagens de controle enviadas à rede, melhorando assim, a eficiência no preenchimento das tabelas de roteamento. O protocolo AODV foi o que obteve melhores resultados, porém deve ser lembrado que o protocolo utiliza a técnica de inundação de muitas mensagens de controle para fazer a manutenção das suas rotas, e, portanto, não é garantia de que a qualidade dos caminhos seja satisfatória, o que poderá ser comprovado na avaliação dos resultados que seguem.

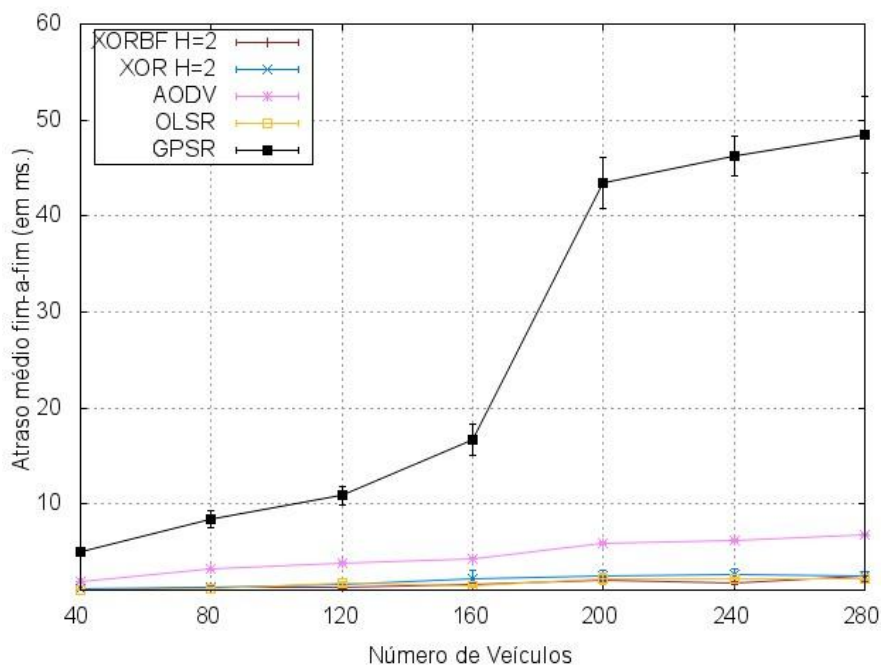
Em seguida, da Figura 6.7 até 6.14, são comparados os valores de atraso na entrega dos pacotes do protocolo XORBF e XOR com as mesmas variações de  $H$  em conjunto com os outros protocolos (OLSR, AODV e GPSR). De forma a facilitar a leitura dos valores de atraso mais reduzidos, nas Figuras 6.8, 6.10, 6.12 e 6.14 são mostrados os resultados dos atrasos em menor escala no eixo Y.

De forma geral, o comportamento do protocolo XOR mostra que conforme é aumentado o valor de  $H$  e o número de veículos, tende a resultar em valores maiores de atraso, devido a uma maior quantidade de mensagens de controle geradas com o objetivo de preencher os *buckets* dos nós. O protocolo OLSR apresenta bons resultados devido ao esquema de seleção dos nós MPR, reduzindo assim, o número de mensagens de controle. O protocolo XORBF na sua variação  $H=2$ , em praticamente todos os cenários consegue obter desempenho praticamente similar ao protocolo OLSR.

Outro fato a se destacar é que a técnica de filtros de Bloom se mostra eficiente em cenários com alta densidade de nós. O protocolo XORBF quando comparado com o protocolo XOR no cenário de 200 nós com o valor de  $H=5$ , alcança desempenho superior em cerca de 95%, pois, o protocolo XOR, conforme é aumentado o número de nós, necessita enviar mensagens de *QUERIES* de forma a preencher os *buckets* mais elevados, cuja probabilidade de se encontrar os nós que preenchem tais *buckets* é reduzida, o que gera maior tráfego de sinalização. Assim, tende a aumentar o número de colisões, gerando maior atraso. O mesmo

comportamento pode ser visto nos cenários de 240 e 280 nós, mostrando que a técnica de filtros de Bloom é eficiente na redução das mensagens de sinalização, conforme pode ser visto na redução do atraso alcançado pelo XORBF na Figura 6.13.

Como já era esperado, o atraso na entrega dos pacotes do protocolo GPSR tende a aumentar conforme é elevada a densidade de veículos, devido às mensagens de sinalização geradas pelo protocolo. O protocolo AODV também perde na comparação devido a sua técnica de inundação que necessita atualizar as tabelas de roteamento constantemente devido às quebras de rotas que ocorrem por causa da alta mobilidade dos nós que leva a constantes mudanças na topologia da rede.



**Figura 6.7 – Atraso na entrega dos pacotes ( $H=2$ ).**

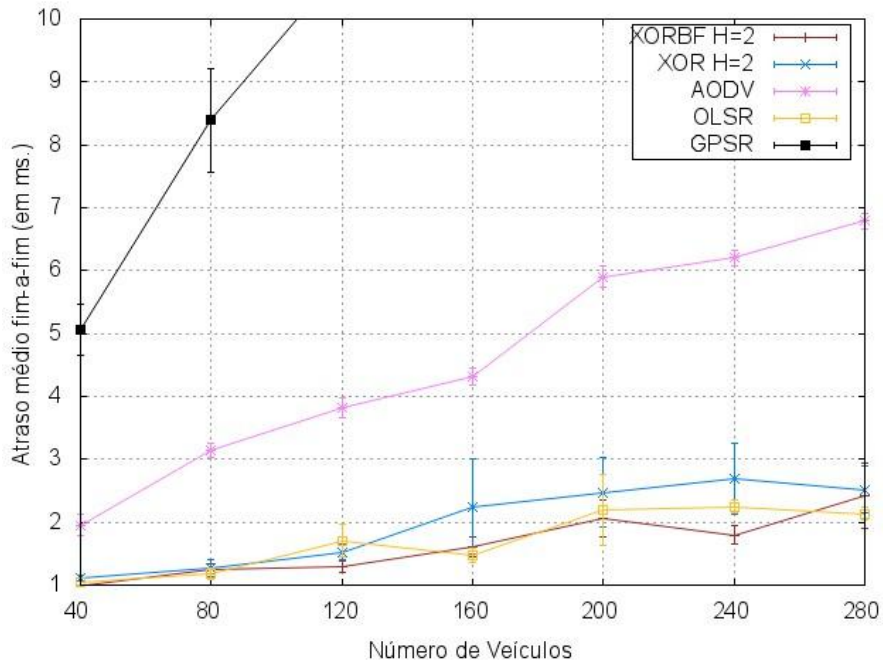


Figura 6.8 – Atraso na entrega dos pacotes ( $H=2$  com zoom em 10 ms.).

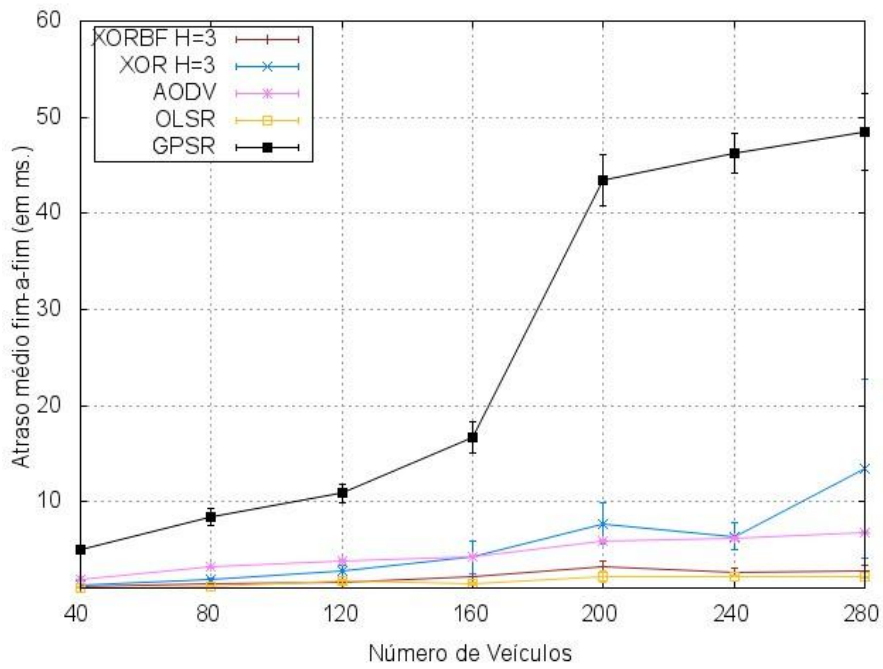


Figura 6.9 – Atraso na entrega dos pacotes ( $H=3$ ).



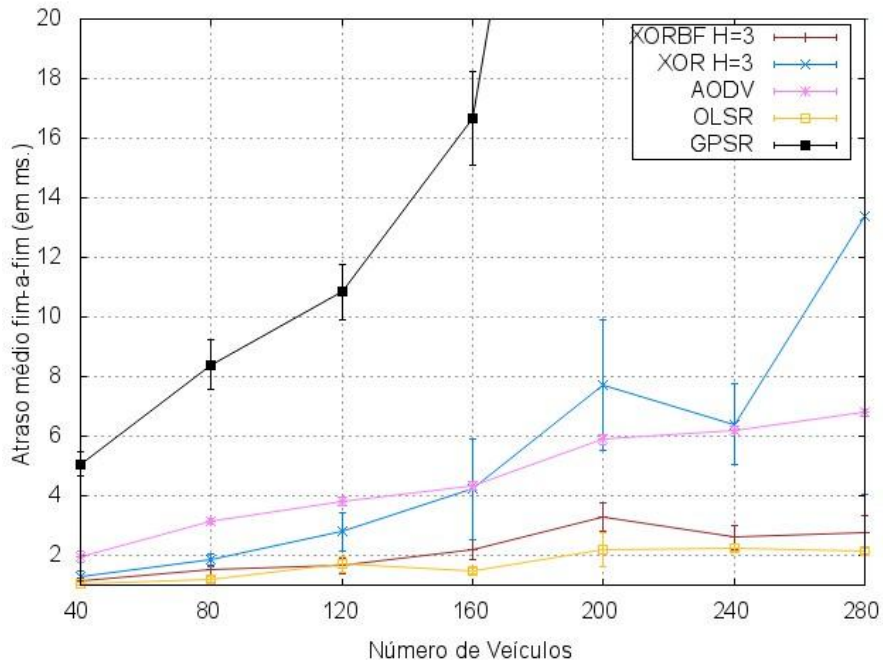


Figura 6.10 – Atraso na entrega dos pacotes ( $H=3$  com zoom em 20 ms.).

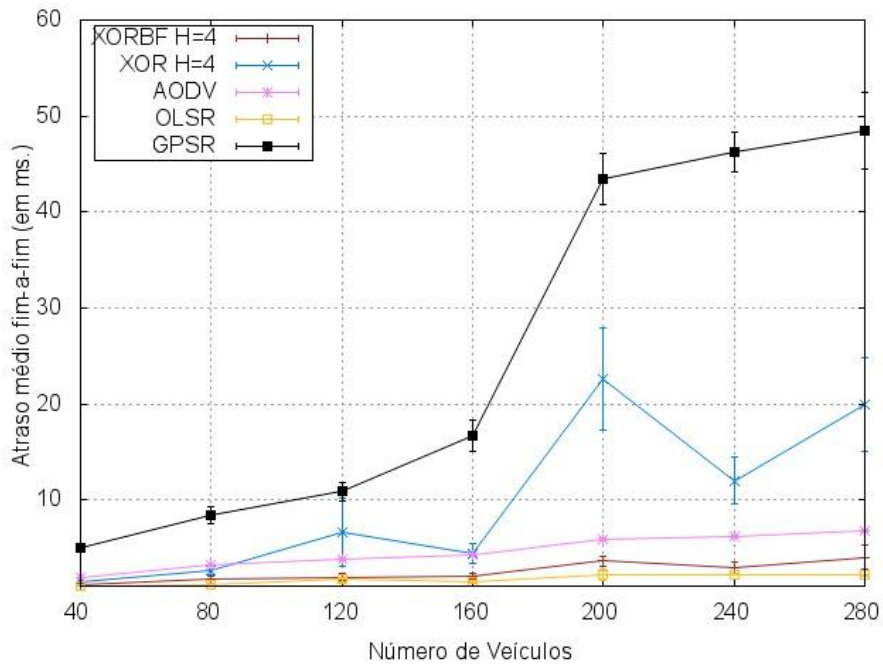


Figura 6.11 – Atraso na entrega dos pacotes ( $H=4$ ).

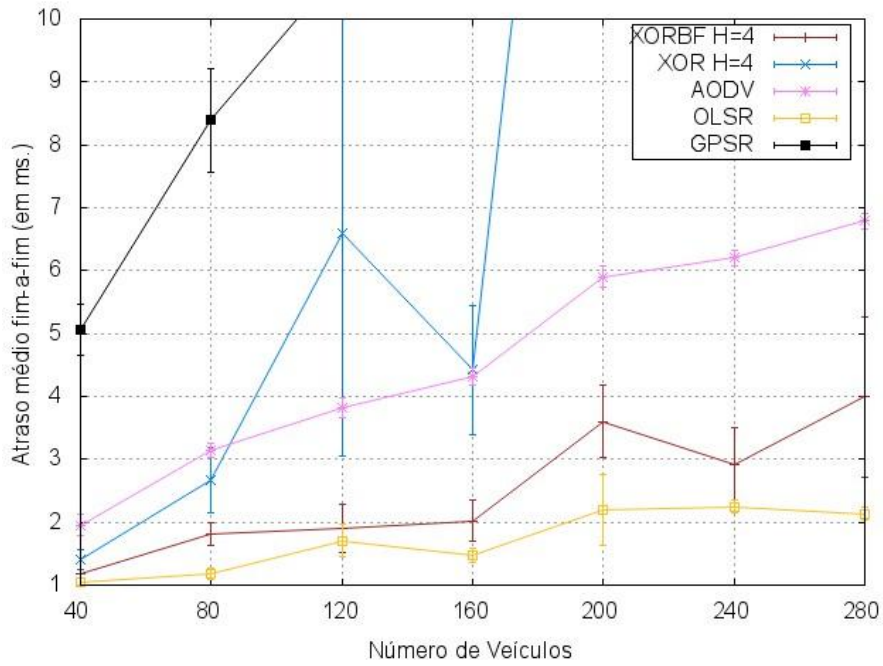


Figura 6.12 – Atraso na entrega dos pacotes ( $H=4$  com zoom em 10 ms.).

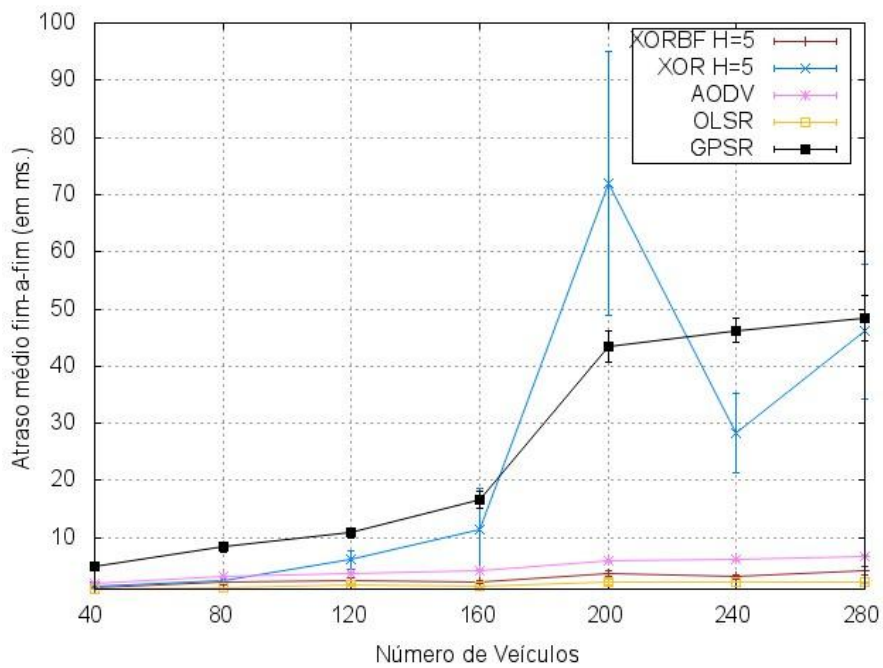
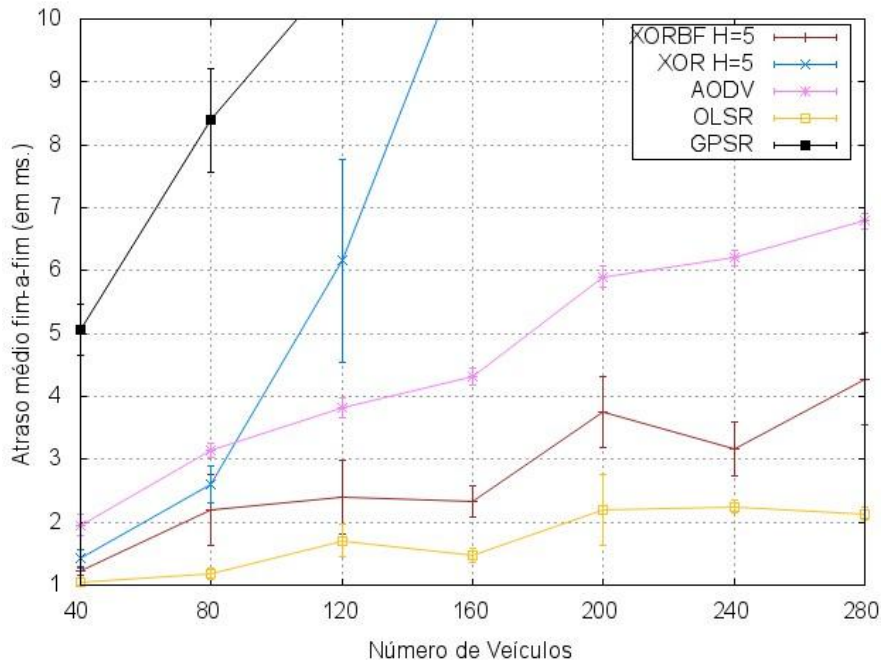


Figura 6.13 – Atraso na entrega dos pacotes ( $H=5$ ).



**Figura 6.14 – Atraso na entrega dos pacotes ( $H=5$  com zoom em 10 ms.).**

Outra métrica que também foi avaliada é o comprimento dos caminhos, em número de saltos. Os resultados de tal métrica serão mostrados nas Figuras 6.15, 6.16, 6.17 e 6.18.

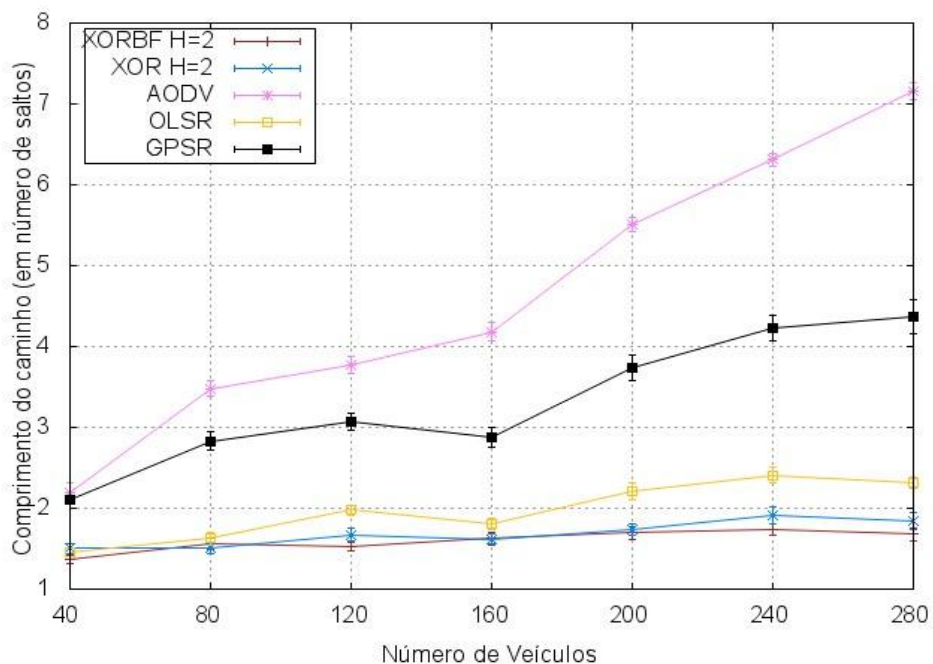
É possível observar que os protocolos AODV e GPSR tiveram valores de comprimento médio dos caminhos superiores aos outros protocolos avaliados, levando a conclusão de que a probabilidade dos caminhos criados por tais protocolos sejam defeituosos é alta. Como é visto na Figura 6.15, o protocolo AODV devido ao fato de inundar a rede com mensagens de sinalização em busca de nós para preenchimento e atualização das tabelas de roteamento e o protocolo GPSR por entrar em situações de máximo local, conforme há o aumento da densidade dos nós, maior é o número de saltos dados.

Os caminhos criados por estes protocolos tendem a não serem caminhos de boa qualidade, pelo fato de que os caminhos terão curto tempo de duração, devido à suscetibilidade de indisponibilidade dos caminhos por causa da quantidade de saltos.

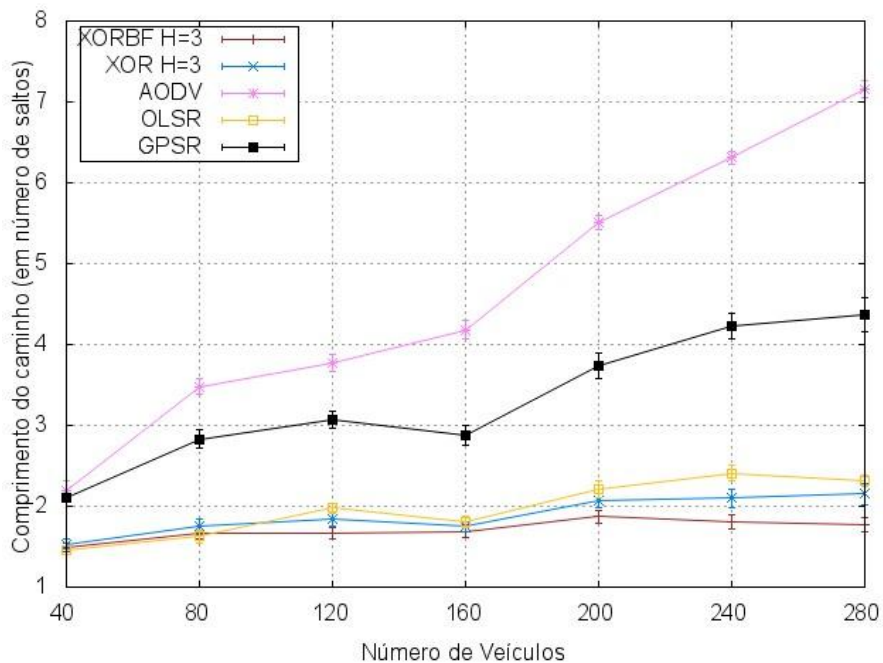
Em relação aos valores obtidos pelos protocolos XORBF e XOR, é importante notar que os recursos incorporados nestes protocolos tais como a visibilidade local e o

mecanismo de limitação de alcance da mensagem de *QUERY*, faz com o processo de encaminhamento das informações seja eficiente. Com a diminuição de mensagens de sinalização, por causa da utilização dos filtros de Bloom, o protocolo XORBF consegue superar o protocolo XOR em praticamente todos os cenários e em todas as variações de *H*.

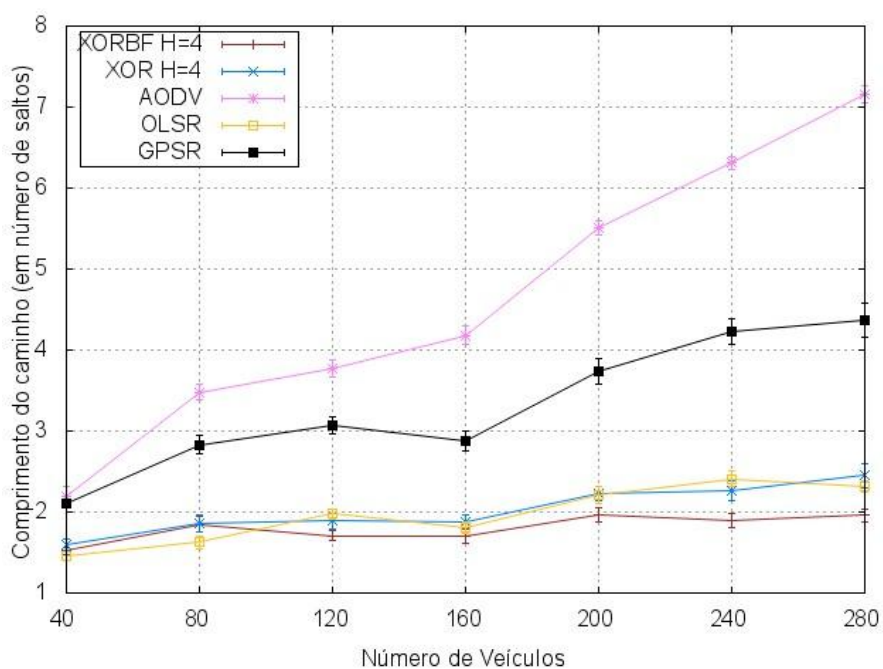
Além disso, é possível observar que, praticamente em todos os cenários variando o valor de *H*, o protocolo XORBF supera o protocolo OLSR, especialmente nos cenários com maior densidade veicular de 200, 240 e 280 nós. Assim, conclui-se que a probabilidade de quebra de enlace dos caminhos é pequena, e assim, tais caminhos tendem a possuir maior tempo de duração. Logo, esses caminhos são de melhor qualidade, quando comparado com os caminhos dos outros protocolos avaliados.



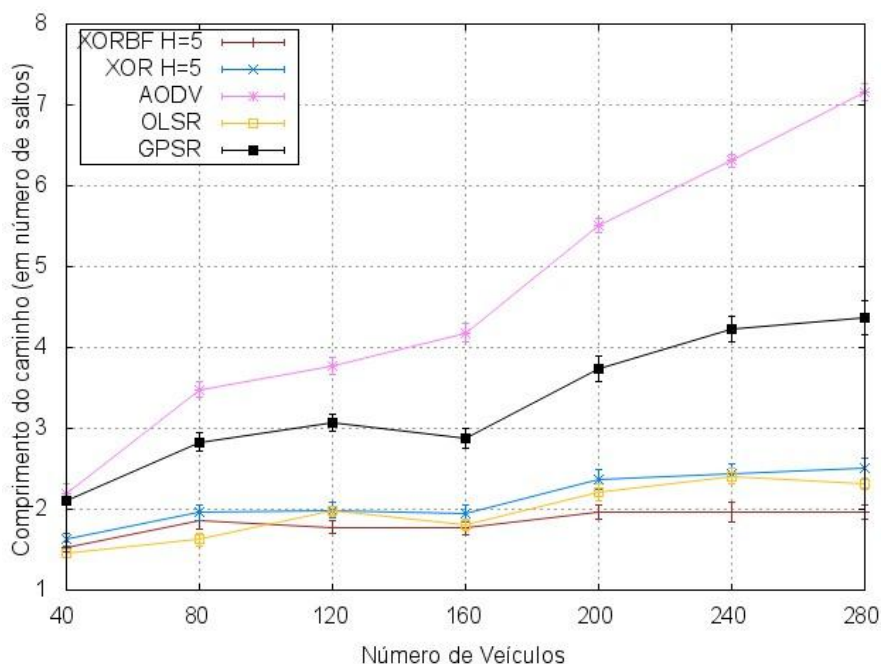
**Figura 6.15 – Comprimento dos caminhos criados pelos protocolos XOR ( $H=2$ ), XORBF ( $H=2$ ), AODV, OLSR e GPSR.**



**Figura 6.16 – Comprimento dos caminhos criados pelos protocolos XOR ( $H=3$ ), XORBF ( $H=3$ ), AODV, OLSR e GPSR.**



**Figura 6.17 – Comprimento dos caminhos criados pelos protocolos XOR ( $H=4$ ), XORBF ( $H=4$ ), AODV, OLSR e GPSR.**



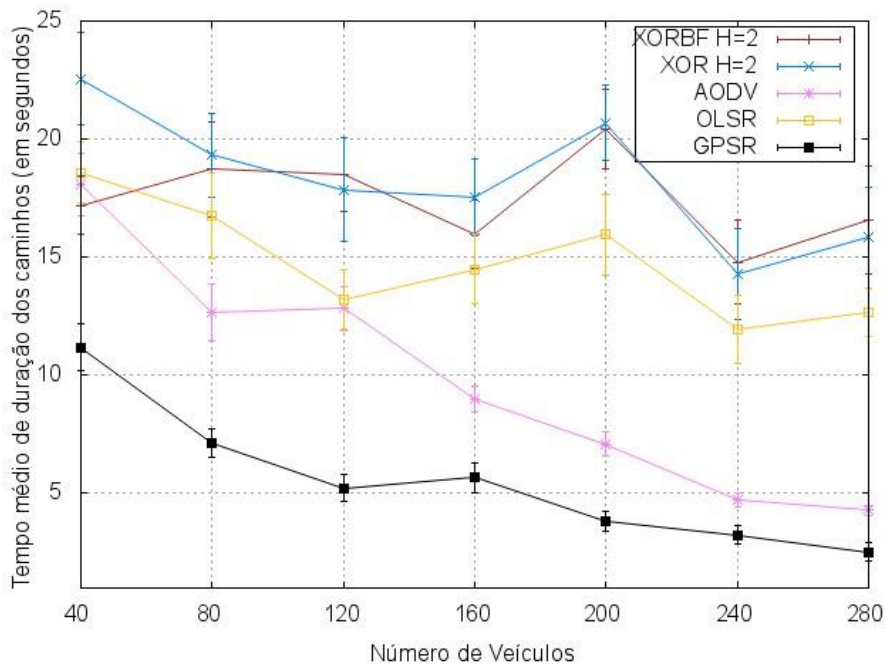
**Figura 6.18 – Comprimento dos caminhos criados pelos protocolos XOR ( $H=5$ ), XORBF ( $H=5$ ), AODV, OLSR e GPSR.**

Outra métrica avaliada é a duração média dos caminhos criados de forma a verificar a qualidade dos caminhos.

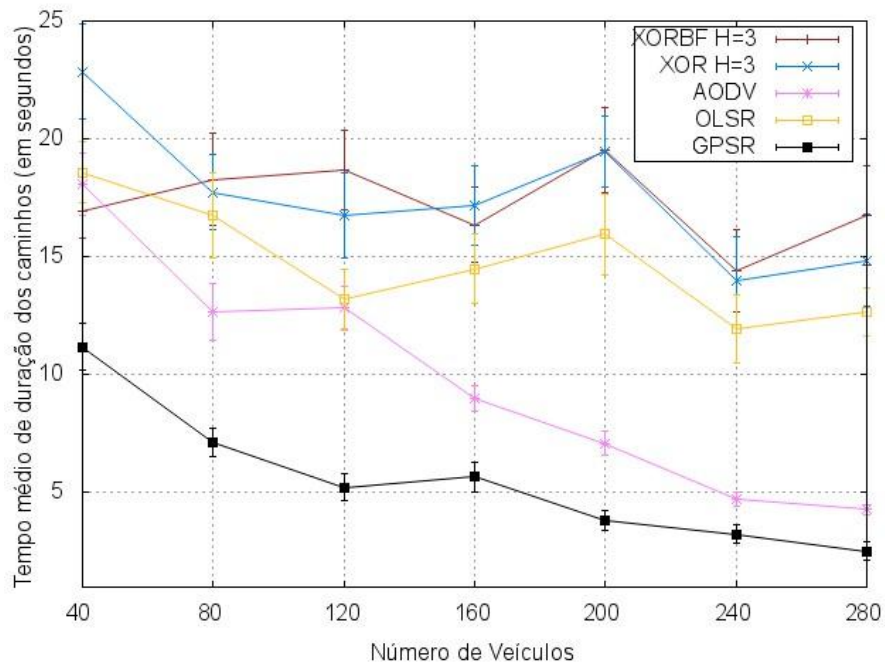
Nas Figuras 6.19 até a 6.22, é possível ver o tempo médio de duração dos caminhos criados pelos protocolos nos diversos cenários simulados. Percebe-se que o protocolo XOR em todas as variações de  $H$  no cenário de 40 nós supera todos os outros protocolos. Porém, conforme há o aumento da variável  $H$  e o aumento no número de veículos, a técnica dos filtros de Bloom é eficiente no preenchimento dos *buckets*, e na redução das mensagens de controle, o que impacta de tal forma que o protocolo XORBF tenha caminhos com maior tempo de duração, quando comparado com todos os outros protocolos avaliados nos cenários de 80 nós e superiores.

Os protocolos AODV e GPSR possuem desempenhos insatisfatórios, mostrando que em cenários com maior densidade de nós os mesmos não se mostram como boas soluções de roteamento. O protocolo OLSR mesmo com a existência dos nós MPR, o que reduz a quantidade de mensagens de controle, não supera os

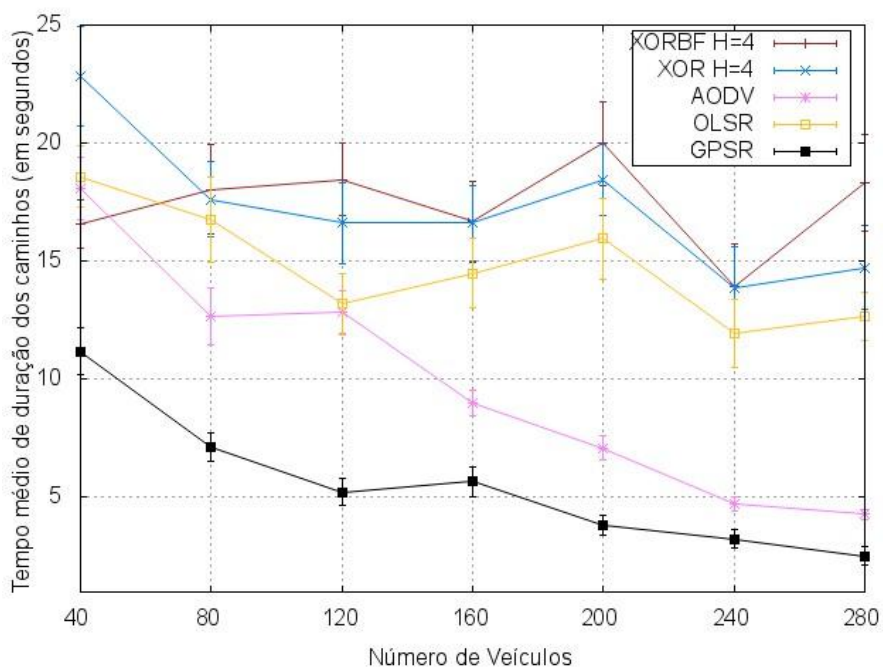
protocolos baseados nas métricas XOR (XOR e XORBF) nos cenários com 80 nós e superiores.



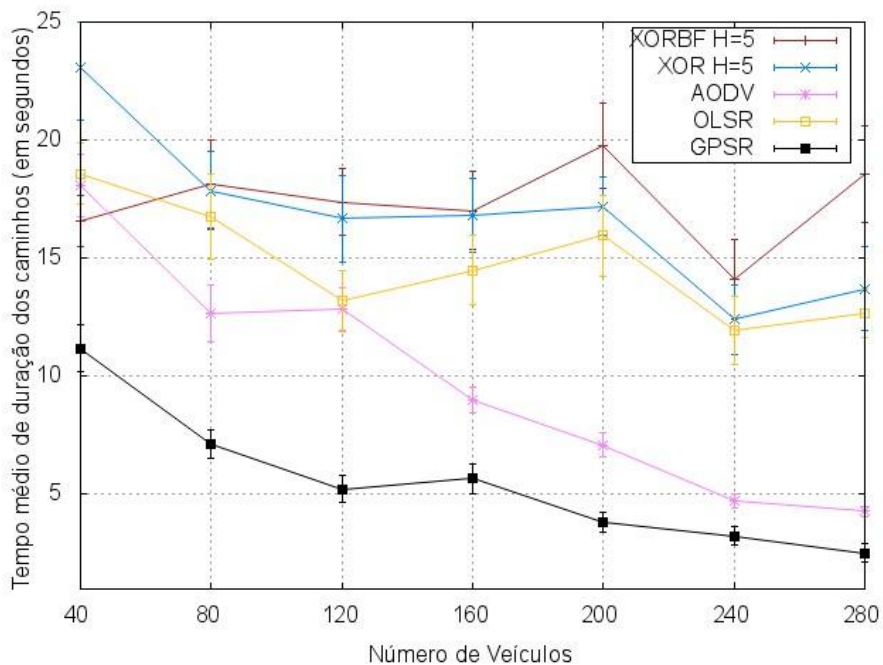
**Figura 6.19 – Tempo médio de duração dos caminhos criados pelos protocolos XOR ( $H=2$ ), XORBF ( $H=2$ ), AODV, OLSR e GPSR.**



**Figura 6.20 – Tempo médio de duração dos caminhos criados pelos protocolos XOR ( $H=3$ ), XORBF ( $H=3$ ), AODV, OLSR e GPSR.**



**Figura 6.21 – Tempo médio de duração dos caminhos criados pelos protocolos XOR ( $H=4$ ), XORBF ( $H=4$ ), AODV, OLSR e GPSR.**



**Figura 6.22 – Tempo médio de duração dos caminhos criados pelos protocolos XOR ( $H=5$ ), XORBF ( $H=5$ ), AODV, OLSR e GPSR.**



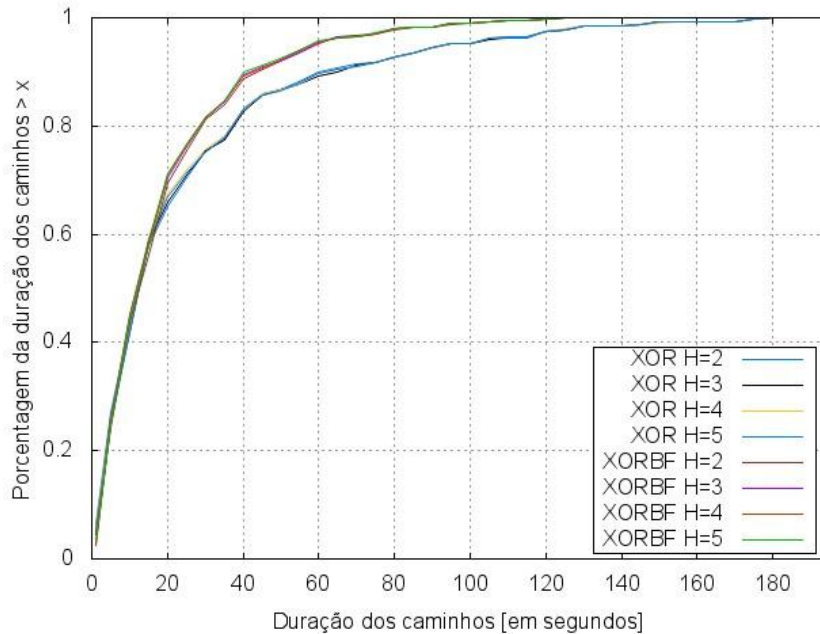
Outra maneira utilizada para mostrar a qualidade dos caminhos criados é a utilização da função de densidade acumulada empírica (FDAE) da duração dos caminhos criados. Dessa forma, é possível analisar os resultados da duração dos caminhos em termos percentuais. Os resultados mostrados foram obtidos através da relação percentual entre a quantidade de caminhos criados e a duração de tais caminhos. No eixo  $y$  está o percentual da duração dos caminhos menor que um dado  $x$  (o valor 1 representa 100%) enquanto que no eixo  $x$  é mostrada a duração dos caminhos em segundos.

Para facilitar a visualização e reduzir o número de gráficos, foi realizada, primeiramente, uma comparação entre as variações de  $H=2$  a  $H=5$  dos protocolos XOR e XORBF, e os resultados mais significativos desses protocolos, em seguida, são mostrados em outro gráfico comparando-os com os protocolos OLSR, AODV e GPSR em cada cenário. Na Figura 6.23 é mostrado um gráfico realizando a primeira comparação citada anteriormente e, na Figura 6.24, a segunda comparação para o cenário com 40 nós.

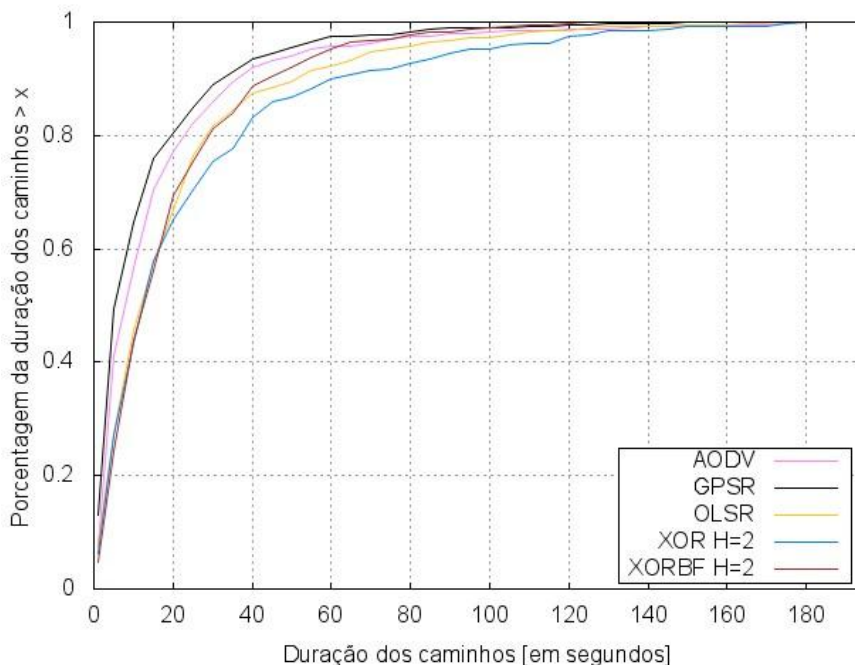
Na Figura 6.23 mostra que conforme é aumentado o valor de  $H$ , pior são os resultados dada a quantidade de mensagens de sinalização geradas em busca de identificadores para preenchimento das tabelas de roteamento. Além disso, a técnica de filtros de Bloom não impacta nos resultados em cenários com baixa densidade.

Na Figura 6.24, os protocolos GPSR, OLSR e AODV criam boa parte dos caminhos com curto tempo de duração. Para os protocolos AODV e GPSR, 77% e 80%, respectivamente, dos caminhos criados por tais protocolos possuem até 20 segundos, o protocolo OLSR consegue alcançar o valor de 67% dos caminhos criados que duram até 20 segundos. Já o protocolo XOR no seu melhor resultado, com a variação de  $H=2$ , possui em torno de 65% dos caminhos durando até os 20 segundos e o protocolo XORBF, também na sua melhor variação com  $H=2$ , possui 69% dos caminhos com 20 segundos.

Portanto, a técnica dos filtros de Bloom pouco impacta no cenário de menor densidade veicular devido à menor carga de tráfego de sinalização gerado tanto pelo protocolo XOR quanto pelo protocolo XORBF.



**Figura 6.23 – FDAE da duração dos caminhos criados pelos protocolos XOR e XORBF com  $H=2$  a  $H=5$  no cenário com 40 veículos.**

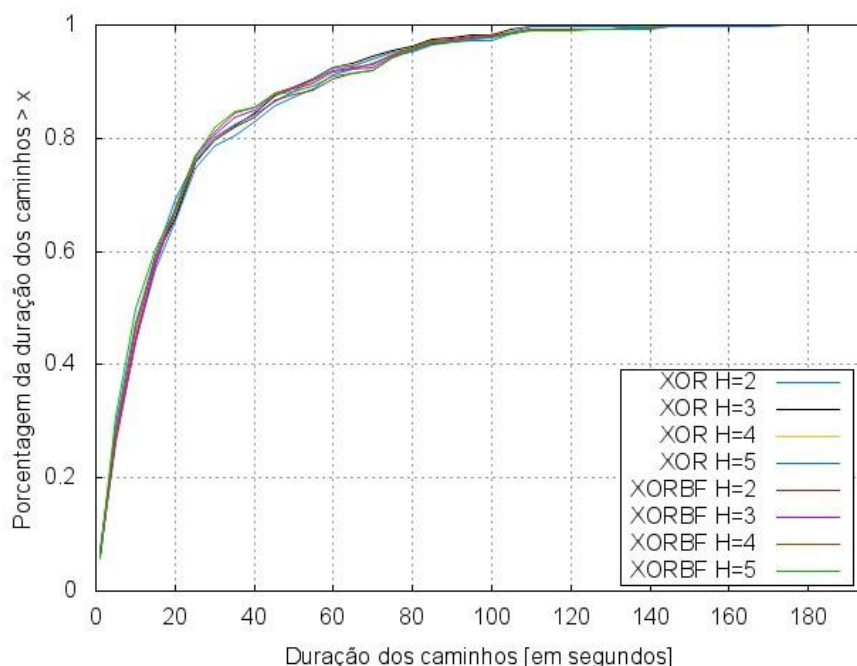


**Figura 6.24 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 40 veículos.**

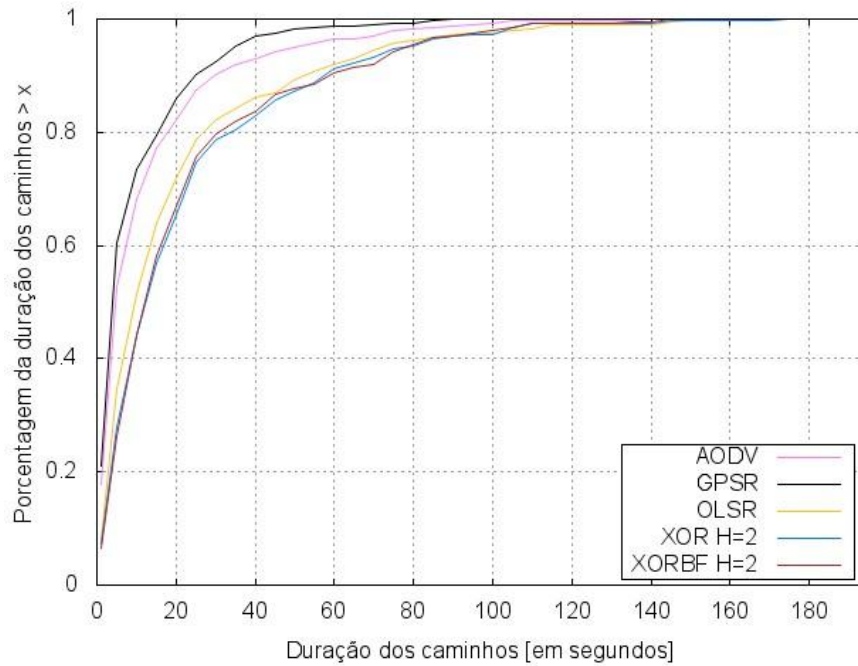
Nas Figuras 6.25 e 6.26 estão representadas as FDAEs da duração dos caminhos criados/disponíveis, dos protocolos XOR e XORBF nas suas variações de  $H=2$  a  $H=5$ , e em seguida, o melhor resultado do XOR e XORBF em comparação com os demais protocolos para o cenário de 80 nós.

Novamente é possível observar que os protocolos GPSR, AODV e OLSR têm uma quantidade significativa de caminhos com baixo tempo de duração quando comparados com o protocolo XOR nas suas variações de  $H$ .

Para os protocolos AODV, OLSR e GPSR os valores estão em torno de 82%, 72% e 86%, respectivamente, dos caminhos criados por tais protocolos possuem até 20 segundos enquanto o protocolo XOR na sua variação  $H=2$  alcança 65% dos caminhos durando menos de 20 segundos. O protocolo XORBF na mesma variação de  $H$  possui 67% dos valores possuindo menos de 20 segundos.



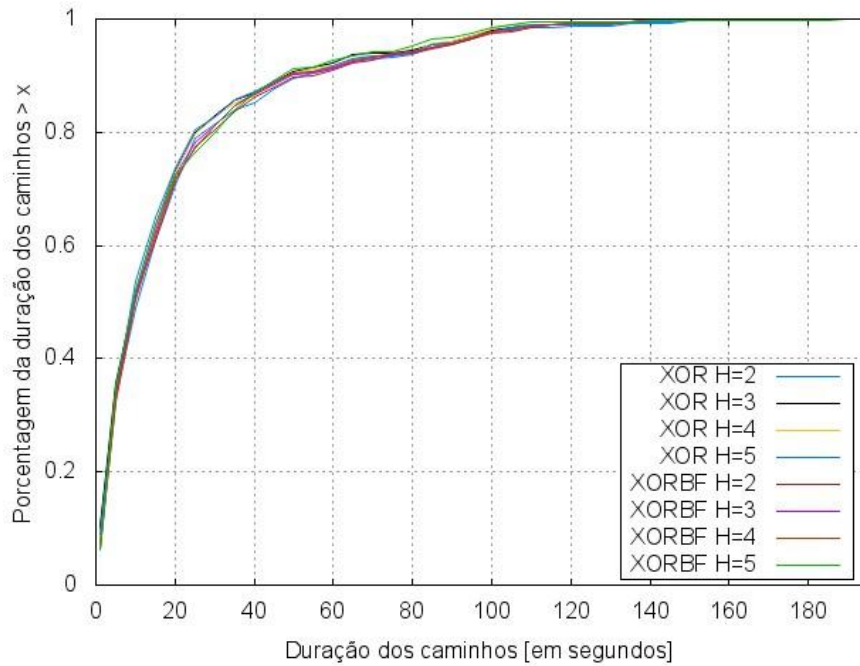
**Figura 6.25 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com  $H=2$  a  $H=5$  no cenário com 80 veículos.**



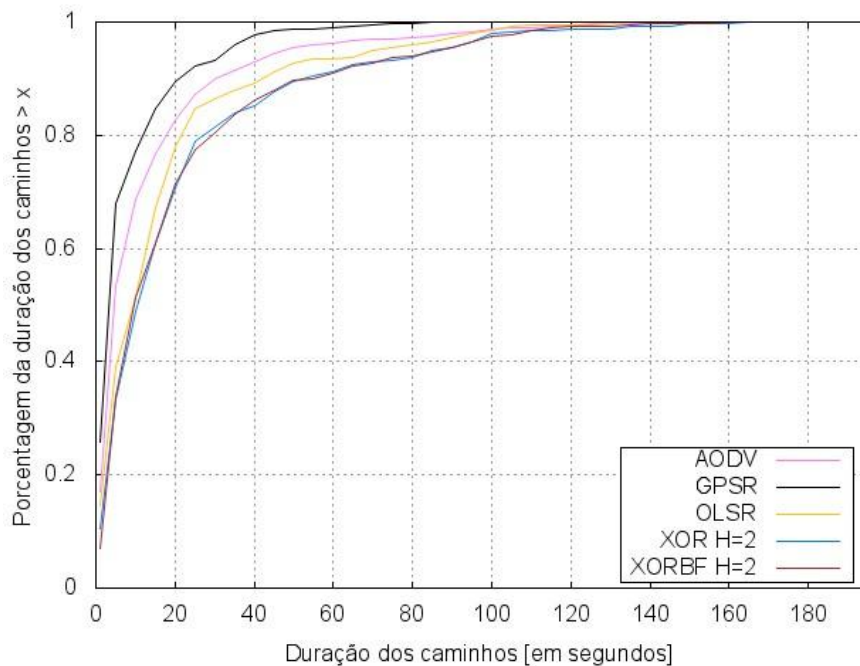
**Figura 6.26 – FDEA da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 80 veículos.**

Para a situação que representa a FDEA da duração dos caminhos criados/disponíveis para o cenário com 120 nós, que é vista nas Figuras 6.27 e 6.28, é mostrado que o protocolo GPSR consegue piorar ainda mais a sua taxa de duração dos caminhos.

Cerca de 90% dos caminhos criados pelo protocolo GPSR possuem menos de 20 segundos de duração. O protocolo AODV possui 83% dos seus caminhos durando menos de 20 segundo enquanto os protocolos OLSR, XOR e XORBF na sua variação de  $H=2$  possuem, respectivamente, 78% e 70,6% e 71,5% dos caminhos criados com duração de até 20 segundos.



**Figura 6.27 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com  $H=2$  a  $H=5$  no cenário com 120 veículos.**



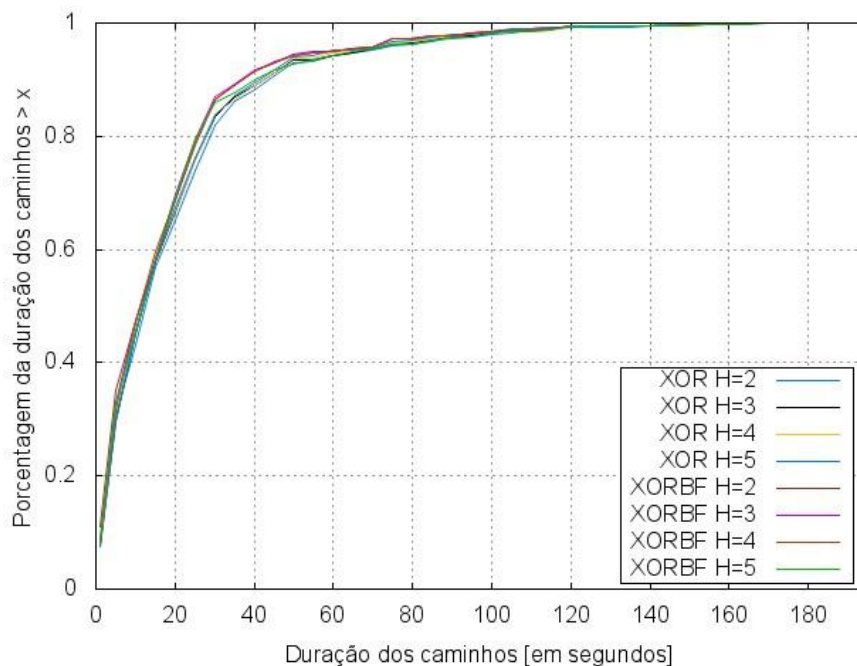
**Figura 6.28 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 120 veículos.**

A duração dos caminhos mostradas nas Figuras 6.29 e 6.30, que caracteriza a FDAE do cenário com 160 nós, mostra resultados dos protocolos XORBF e XOR nas

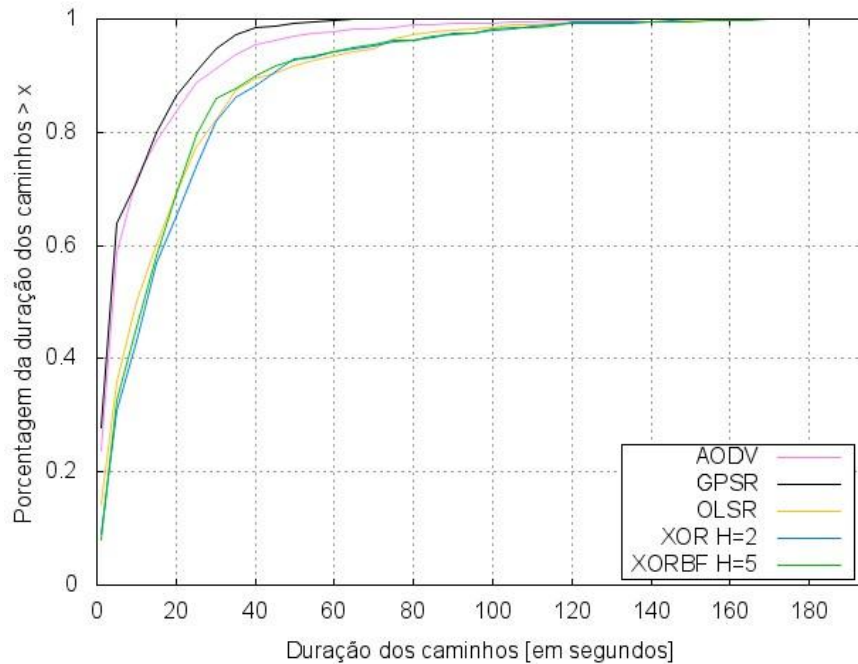
suas variações de  $H=2$  a  $H=5$ , e o melhor resultado destes é comparado com os outros protocolos analisados.

Enquanto 86% dos caminhos criados pelo protocolo GPSR e 84% dos caminhos criados pelo protocolo AODV duram menos de 20 segundos, o protocolo XOR na sua variação  $H=2$ , possui em torno de 65% dos caminhos durando menos de 20 segundos, vencendo também o protocolo OLSR que alcança 69%. Porém o protocolo XORBF na sua variação de  $H=2$  alcança a taxa de 68% dos caminhos com duração de até 20 segundos.

Apesar disso, os protocolos XORBF, XOR com variações de  $H=5$  e OLSR alcançam praticamente a mesma porcentagem de caminhos criados que duram até os 100 segundos, respectivamente, 98,2%, 98,4% e 98,5%, mostrando que o protocolo XORBF cria uma quantidade de caminhos com maior qualidade conforme se aumenta o  $H$  em cenários com densidade maiores de veículos.



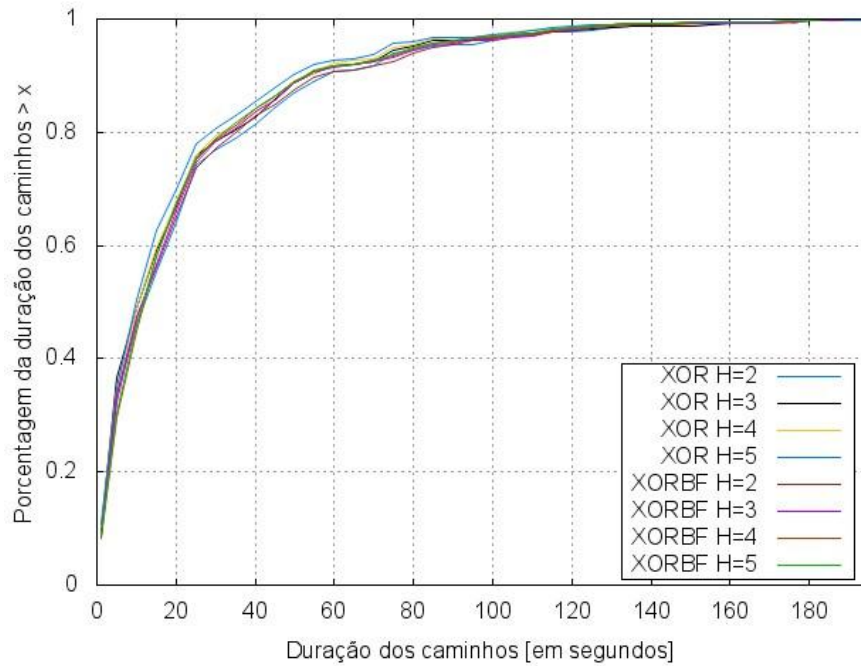
**Figura 6.29 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com  $H=2$  a  $H=5$  no cenário com 160 veículos.**



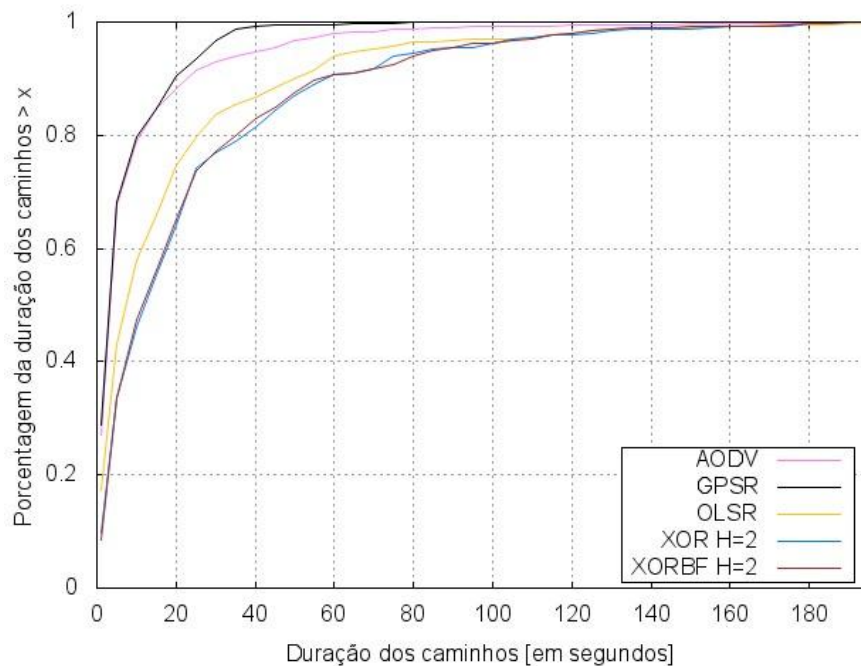
**Figura 6.30 – FDEA da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 160 veículos.**

A seguir, nas Figuras 6.31 e 6.32 é apresentado a FDEA do cenário com 200 nós variando o fator  $H$  dos protocolos XOR e XORBF. O protocolo GPSR e o AODV conseguem piorar ainda mais os seus resultados quando comparados com os outros cenários.

Na utilização destes protocolos, 88% dos caminhos criados pelo protocolo AODV e 90% dos caminhos criados pelo protocolo GPSR duram, no máximo 20 segundos. O protocolo XOR com  $H=2$  alcança em torno de 64% dos caminhos criados com duração de até 20 segundos, novamente superando também o protocolo OLSR que alcança 75% dos caminhos criados com no máximo 20 segundos de duração. O protocolo XORBF na variação de  $H=2$  possui 65% dos caminhos criados com duração máxima de 20 segundos.



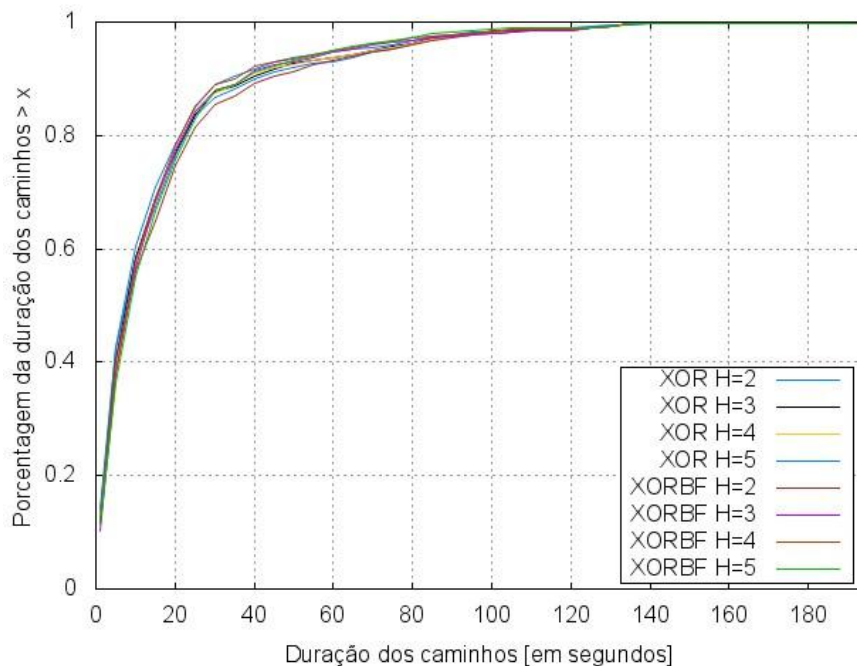
**Figura 6.31 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com  $H=2$  a  $H=5$  no cenário com 200 veículos.**



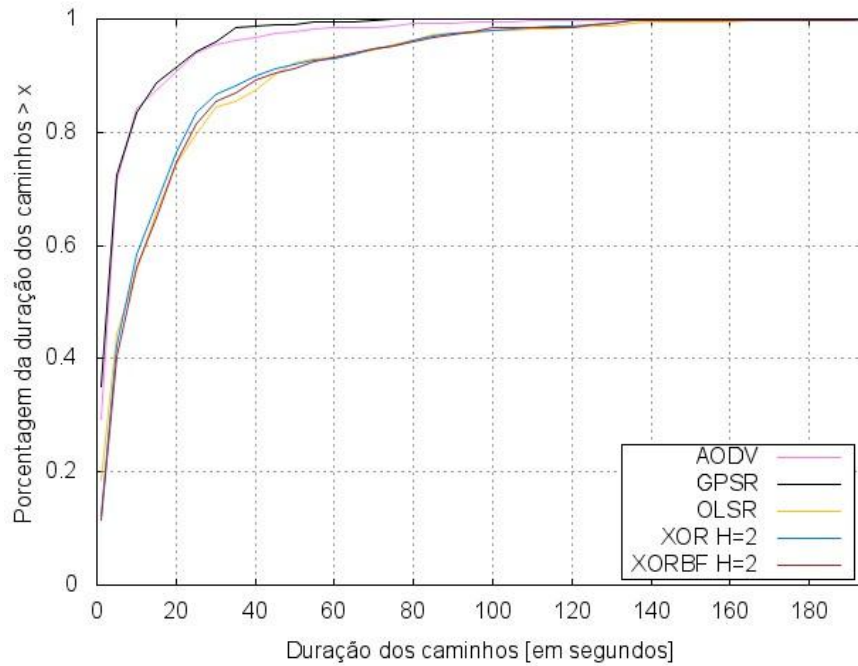
**Figura 6.32 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 200 veículos.**



A FDAE do cenário com 240 nós variando o fator  $H$  dos protocolos XOR e XORBF é mostrado na Figura 6.33, assim como o melhor resultado do protocolo XOR e XORBF é mostrado de forma comparativa aos outros protocolos avaliados na Figura 6.34. Os protocolos GPSR e AODV continuam a tendência de piora nos resultados conforme o aumento de nós. Na utilização destes protocolos, 91% dos caminhos criados pelo protocolo AODV e 92% dos caminhos criados pelo protocolo GPSR duram até os 20 segundos. O protocolo XOR na sua variação  $H=2$  alcança em torno de 76% dos caminhos criados com duração de até 20 segundos, enquanto o protocolo OLSR alcança 74% dos caminhos criados com no máximo 20 segundos de duração. O protocolo XORBF na variação de  $H=2$  alcança, assim como o protocolo OLSR, os mesmos 74% dos caminhos criados com duração máxima de 20 segundos.

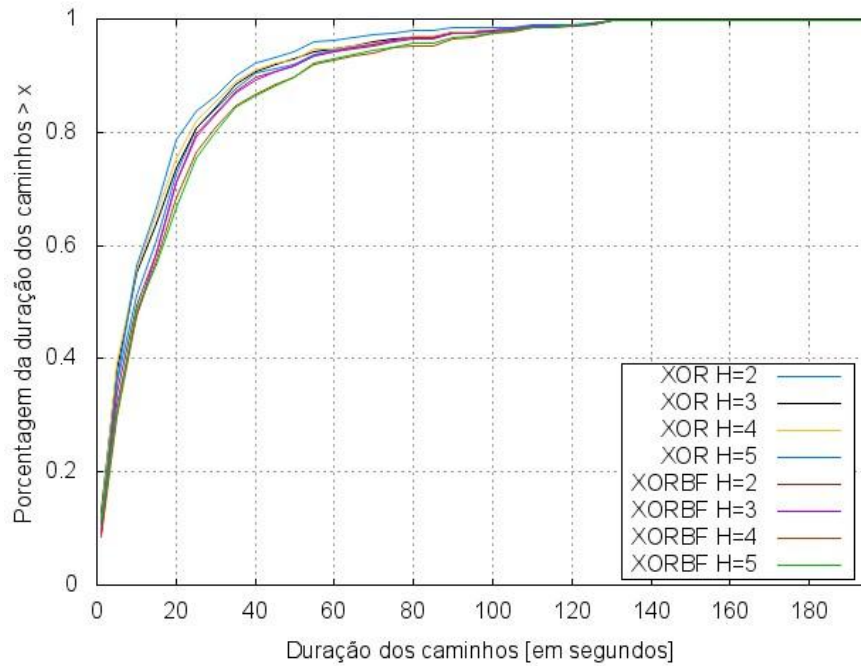


**Figura 6.33 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com  $H=2$  a  $H=5$  no cenário com 240 veículos.**

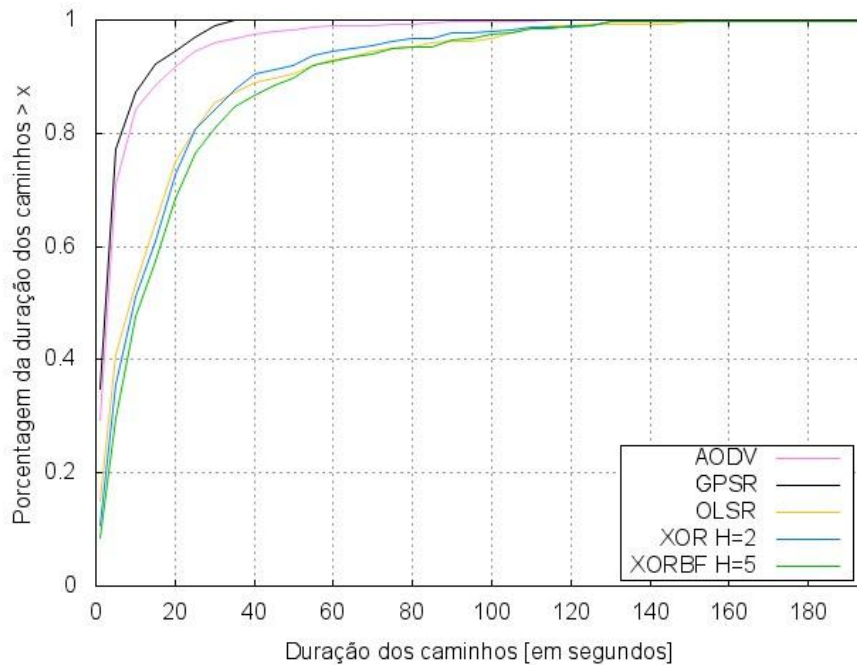


**Figura 6.34 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 240 veículos.**

Para finalizar, nas Figuras 6.35 e 6.36 são mostrados as FDAEs para o cenário com 280 nós variando o fator  $H$  dos protocolos XOR e XORBF, assim como o melhor resultado do protocolo XOR e XORBF é mostrado de forma comparativa aos outros protocolos avaliados OLSR, GPSR e AODV, respectivamente. Os protocolos GPSR e AODV continuam a piorar nos resultados conforme o aumento de nós. Na utilização destes protocolos, 92% dos caminhos criados pelo protocolo AODV e 95% dos caminhos criados pelo protocolo GPSR duram até os 20 segundos. O protocolo XOR na sua variação  $H=2$  alcança em torno de 73% dos caminhos criados com duração de até 20 segundos, enquanto o protocolo OLSR alcança 75% dos caminhos criados com no máximo 20 segundos de duração. O protocolo XORBF na variação de  $H=5$  alcança 65% dos caminhos criados com duração máxima de 20 segundos, superando de forma significativa todos os outros protocolos.



**Figura 6.35 – FDAE da duração dos caminhos dos protocolos XOR e XORBF com  $H=2$  a  $H=5$  no cenário com 280 veículos.**



**Figura 6.36 – FDAE da duração dos melhores caminhos criados pelos protocolos XOR e XORBF em comparação com os protocolos AODV, OLSR e GPSR no cenário com 280 veículos.**

### 6.1.3 Discussão dos Resultados

Analisando todos os resultados vistos na seção anterior, é possível ver que o mecanismo de alcance da *QUERY* e o de visibilidade local são interessantes conforme são vistos em alguns resultados.

A taxa de caminhos disponíveis do protocolo XORBF variando o número de saltos da *QUERY* de  $H=2$  a  $H=5$  nos diversos cenários possui uma pequena variação, onde o valor de  $H$ , portanto, não impacta nos resultados. O protocolo XORBF consegue alcançar resultados próximos ao protocolo XOR e em algumas situações ser até superior. Vale destacar que no cenário de 280 nós o protocolo XORBF com  $H=5$  supera o protocolo OLSR, o que mostra que a técnica dos filtros de Bloom melhora o preenchimento dos *buckets* em cenários com maior densidade de nós. Na Figura 6.6, para o cenário de 280 nós, é possível ver que o protocolo XORBF é o único que melhora a taxa de caminhos criados.

Em relação ao atraso na entrega de pacotes, com a utilização dos filtros de Bloom no protocolo XORBF, conforme é incrementado o valor de  $H$  e a densidade de veículos, faz com que o alto atraso na entrega dos pacotes do protocolo XOR seja mitigado. O que impacta neste resultado é o fato de gerar menor número de mensagens de sinalização de forma a encontrar nós que preencham os *buckets* vazios, o que gera menos colisões, e utilização menor da banda de transmissão, conseqüentemente, reduzindo o atraso na entrega, praticamente tendo valores iguais ao protocolo OLSR.

É possível concluir no trabalho aqui apresentado, que com o protocolo XORBF, aplicações que são sensíveis ao atraso, como notificações de acidentes, congestionamento, entre outras são viáveis, o que não pode ser dito o mesmo do protocolo XOR, onde é possível ver que a partir do cenário com 120 nós, conforme pode ser visto da Figura 6.9 até a Figura 6.14, o valor de atraso alcançado é um pouco alto para este tipo de aplicação.

Sobre o comprimento dos caminhos, cujos resultados podem ser vistos das Figuras 6.15 até a 6.18, o protocolo XORBF alcança resultados interessantes, superando, de forma geral, todos os outros protocolos, em especial nos cenários com maior densidade de nós. Esta métrica é importante, pois quanto maior o número de saltos, maior a probabilidade de quebra de conectividade, o que impacta diretamente no tempo de duração dos caminhos. Por isso é possível perceber, nas Figuras 6.19 até a Figura 6.22, que o tempo médio de duração dos caminhos dos protocolos OLSR e GPSR possuem a tendência de ter caminhos com menor tempo de duração conforme há o aumento do número de nós.

Destaca-se, ainda, os resultados obtidos pelo XORBF com  $H=5$  a partir do cenário de 80 nós, onde supera todos os protocolos avaliados. Outra forma de avaliar a qualidade dos caminhos criados pelos protocolos nos cenários avaliados é a utilização da função de densidade acumulada empírica (FDAE). Dessa forma é possível avaliar em valores percentuais a duração dos caminhos criados pelos protocolos.

Conforme visto nas Figuras 6.23 a 6.36, os resultados referentes ao protocolo XORBF são animadores, em especial para cenários com maior densidade de veículos. Mesmo tendo uma taxa de caminhos criados superior a todos os protocolos, o protocolo AODV não cria caminhos que durem muito tempo o que implica em caminhos criados, porém com pouca qualidade. O que não ocorre com o protocolo XORBF nas suas variações de  $H=2$  a  $H=5$ , em especial na variação  $H=5$ , no cenário com 280 nós, quando comparado com os outros protocolos avaliados. O protocolo XORBF em cenários com menor quantidade de nós, em geral, alcança resultados próximos ao XOR e ao OLSR. Mas em cenários com maior densidade de nós, o que de fato é uma característica dos cenários urbanos, é possível observar que o XORBF consegue obter resultados interessantes em todas as métricas analisadas perante seus concorrentes, em especial ao protocolo que lhe é mais próximo nas suas

características, o XOR, o que demonstra que a técnica dos filtros de Bloom em redes veiculares é promissora.

## 7 Conclusões

Este capítulo apresenta as conclusões do trabalho realizado e propõe alguns trabalhos futuros que podem ser realizados.

### 7.1 Discussões finais

Neste trabalho foi abordado um tema que tem chamado muita atenção por parte da comunidade científica: o roteamento em redes *ad hoc* veiculares. Protocolos tais como o AODV, GPSR e OLSR, não têm como característica lidar em ambientes com freqüentes mudanças na topologia da rede, que é o caso das VANETs, devido ao excessivo número de mensagens de controle geradas para ter as tabelas de roteamento dos nós constantemente atualizadas. Estes protocolos foram concebidos originalmente para MANETs, redes estas com menor grau de mobilidade, o que implicava em um número reduzido de mudanças na topologia, e que levava a um número reduzido de mensagens de controle na rede.

Assim, investigou-se a possibilidade de criar um novo protocolo de roteamento que fosse capaz de apresentar um melhor desempenho lidando melhor com a alta mobilidade imposta pelas redes veiculares, especificamente, para redes veiculares em cenários urbanos com maior densidade de veículos.

É possível concluir, segundo análises realizadas através das simulações, que os protocolos de roteamento existentes para redes móveis *ad hoc* ainda não estão

preparados para este tipo de cenário/rede, devido à fragilidade em lidar com as constantes mudanças na topologia deste tipo de rede.

O fator escalabilidade deve ser levado em consideração, pois os protocolos de roteamento aqui avaliados tendem a ter baixo desempenho em cenários com uma maior densidade de nós, devido ao grande número de mensagens de sinalização gerado por eles. Por isso, foi encontrado na literatura um protocolo baseado na métrica XOR voltado para redes *ad hoc*. Partindo deste protocolo, com o objetivo de buscar melhores resultados, foi proposto neste trabalho um novo protocolo chamado de XORBF, abordado no Capítulo 5.

O protocolo XORBF manteve boa parte das características inerentes ao protocolo XOR. A principal mudança foi a utilização dos filtros de Bloom, com o objetivo de reduzir as mensagens de sinalização, buscando assim um melhor desempenho nos cenários dinâmicos das VANETs.

Os resultados do Capítulo 6 mostram que o protocolo XORBF, de forma geral, alcança melhores resultados que o protocolo OLSR que vem a ser um dos principais protocolos de roteamento existentes para redes *ad hoc*, tanto no cenário com menor densidade de nós quanto no cenário com maior densidade.

Já na métrica atraso na entrega dos pacotes, o protocolo XORBF praticamente se iguala ao protocolo OLSR. É possível notar que conforme o número de saltos da *QUERY* é aumentado, o protocolo XOR sofre maior atraso o que não ocorre com o protocolo XORBF. Isso se deve ao fato de que o XORBF, ao utilizar os filtros de Bloom, conseguir reduzir o número de mensagens de sinalização, diminuindo colisões e conseguindo melhores resultados.

Na métrica de duração média dos caminhos, é mostrado que conforme o aumento do fator *H* consoante o aumento do número de nós, o protocolo XORBF consegue resultados superiores aos seus concorrentes. Por meio da técnica filtro de Bloom, consegue-se consultar maior número de nós obtendo respostas com melhor



qualidade, o que faz criar caminhos com menor suscetibilidade de quebra de enlace, aumentando assim, o tempo de vida do enlace criado.

Os resultados obtidos mostram que o protocolo XORBF é uma alternativa interessante para os ambientes veiculares urbanos, pois lida melhor com a alta mobilidade existente neste tipo de rede.

## **7.2 Trabalhos Futuros**

Como trabalhos futuros seria muito interessante verificar o desempenho destes protocolos, em um cenário urbano mais complexo/realístico com um maior número de nós. Características como a presença de semáforos, faixas duplas, vias de sentidos opostos, entre outras deveriam ser levados em consideração.

Analisar a carga de tráfego de dados também seria de grande valia. Verificar de que forma os protocolos se comportam conforme aumentam os dados para serem transmitidos é de grande interesse. Além disso, realizar comparações com outros protocolos projetados exclusivamente para VANETs.

Outro trabalho seria avaliar o comportamento dos protocolos XOR e XORBF, variando os fatores  $K$  e  $H$  em conjunto, de forma a verificar como se comportam nos cenários das VANETs.

## Referências Bibliográficas

- [1] CORSON, S.; MACKER, J., “Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations”, RFC Editor, 1999.
- [2] PERKINS, C.E.; ROYER, E.M., “Ad-hoc on-demand distance vector routing.” In: Second IEEE Workshop on Mobile Computing Systems and Applications, pp. 90–100, New Orleans, Feb. 1999.
- [3] JOHNSON, D.B.; MALTZ, D.A., “Dynamic source routing in ad hoc wireless networks”. In: Mobile Computing, Kluwer Academic Publishers, pp. 153–181, 1996.
- [4] JACQUET, P.; MUHLETHALER, P.; CLAUSEN, T. et al., “Optimized link state routing protocol for ad hoc networks”. In: IEEE International Multi Topic Conference, pp. 62–68, Pakistan, Dec. 2001.
- [5] TALEB, T.; SAKHAEI, E.; JAMALIPOUR, A. et al., “A stable routing protocol to support ITS services in VANET networks”. In: IEEE Transactions on Vehicular Technology, Vol. 56, no. 06, pp. 3337–3347, Nov. 2007.
- [6] NAUMOV, V.; BAUMANN, R.; GROSS, T., “An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces”. In: ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2006
- [7] NAMBOODIRI, V.; GAO, L., “Prediction-Based Routing for Vehicular Ad Hoc Networks”, In: IEEE Transactions on Vehicular Technology, Vol. 56, no. 04, pp. 2332–2345, Jul., 2007.
- [8] TONGUZ, O.; WISITPONGPHAN, N.; BAI, F. et al., “Broadcasting in VANET”, In: Proceedings of IEEE Workshop on Mobile Networking for Vehicular Environments (MOVE '07), pp. 7–12, Anchorage, Alaska, USA, May 2007
- [9] BISWAS, S.; TATCHIKOU, R.; DION, F.; “Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety”, In: IEEE Communications Magazine, Vol. 44, no.1, pp. 74–82, Jan. 2006
- [10] NAUMOV, V.; GROSS, T., “Connectivity-Aware Routing (CAR) in Vehicular Ad Hoc Networks”, In: Proceedings of IEEE INFOCOM 2007, pp. 1919–1927, May. 2007
- [11] NZOUONTA, J.; RAJGURE, N.; WANG, G. et al., “VANET Routing on City Roads Using Real-Time Vehicular Traffic Information”, In: IEEE Transactions on Vehicular Technology, vol.58, no. 07, pp. 3609–3626, Sept. 2009
- [12] MAYMOUNKOV, P.; MAZIERES, D., “Kademlia: A peer-to-peer information system based on the xor metric”. In: First International Workshop on Peer-to-Peer Systems, pages 53–65, 2002.

- [13] LI, F.; WANG, Y., "Routing in vehicular ad hoc networks: A survey". IEEE Vehicular Technology Magazine, Vol. 2, no. 02, pp. 12–22, Jun. 2007
- [14] YOUSEFI, S.; MOUSAVI, M.S.; FATHY, M., "Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives". In: 6th International Conference on ITS Telecommunications Proceedings, pp. 761-766, Chengdu, Jun. 2006
- [15] GERLA, M.; KLEINROCK, L., "Vehicular networks and the future of the mobile internet. In: Elsevier Computer Networks, 2010.Vol. 54, no. 4, pp. 527-544, Mar. 2010.
- [16] GPS Global Positioning System. Disponível em: <http://www.gps.gov/>. Acesso em Dezembro de 2011.
- [17] LEE, U.; GERLA, M.; "A survey of urban vehicular sensing platforms". In: The International Journal of Computer and Telecommunications Networking,
- [18] JAWHAR, I.; MOHAMED, N.; ZHANG, L., "Inter-Vehicular Communications Systems, Protocols and Middleware". In:Fifth IEEE International Conference on Networking, Architecture, and Storage, pp. 282-287, Macau, Jul. 2010
- [19] SICHITIU, M. L.; KIHIL, M., "Inter-Vehicle Communications Systems: A Survey". In: IEEE Communications Surveys & Tutorials, Vol. 10, no. 2, pp. 88-105, Aug. 2008
- [20] BOUKERCHE, A.; OLIVEIRA, H.A.B.F; NAKAMURA, E.F. et al., "Vehicular Ad Hoc Networks: A New Challenge for Localization-Based Systems". In: Proceedings of Computer Communications, pp. 2838-2849, Phoenix, Mar. 2008
- [21] WANG, S. Y., "Predicting the Lifetime of Repairable Unicast Routing Paths in Vehicle-Formed Mobile Ad Hoc Networks on Highways". In: 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 2815-2819, Barcelona, Sep. 2004
- [22] HARRY, J; BONNET, C., "A lower Bound for Vehicles' Trajectory Duration". In: Proceedings of the IEEE 62nd Semiannual Vehicular Technology Conference, pp. 2273-2277, Dallas, Sept. 2005
- [23] JARUPAN, B.; EKICI, E., "A survey of cross-layer design for VANETs". Ad Hoc Networks.Elsevier. Vol. 9, no. 5, pp. 966-983, Dec. 2010
- [24] HARRY, J.; FILALI, F.; BONNET, C., "Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy". In: IEEE Communications Surveys & Tutorials. Vol. 11, no. 04, pp. 19-41, Dec. 2009
- [25] FUBLER, H.; MAUVE, M.; HARTESTEIN, H. et al., "Location-Based Routing for Vehicular Ad Hoc Networks", In: Mobile Computing and Communication Review, Vol. 1, Number 2, 2002.
- [26] FUBLER, H.; HARTESTEIN, H.; JORG, W. et al., "Contention-Based Forwarding for Street Scenarios", In: Proceedings of the 1<sup>st</sup> International Workshop in Intelligent Transportation (WIT 2004), pp.150-160, Hamburg, Mar., 2004.
- [27] A. Garrido "Encaminhamento plano em Redes Ad-hoc Veiculares," Msc. Dissertation – Universidade Nova de Lisboa, Lisboa, Portugal., 2010
- [28] ARBABI, H.; WEIGLE, M.C.; "Highway Mobility and Vehicular Ad Hoc Networks in NS-3", In: Proceedings of the Winter Simulation Conference, pp. 2991-3003, Baltimore, 2010.

- [29] LEE, K.C.; LEE, U.; GERLA, M., "Survey of Routing Protocols in Vehicular Ad Hoc Networks". In: *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges*, IGI Global, Oct, 2009
- [30] PERKINS, C.; BHAGWAT, P., "Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers". In: *SIGCOMM*, pp. 234–244, New York, 1994.
- [31] HAERRI, J.; FILALI, F.; BONNET, C., "Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns". In: *5th IFIP Mediterranean Ad-Hoc Networking Workshop*, Lipari, Jun. 2006.
- [31] ADJIH, C.; BACCELLI, E.; JACQUET, P., "Link state routing in wireless ad-hoc networks". In: *IEEE Military Communications Conference*, pp. 1274–1279, Washington – DC, Nov. 2003
- [32] GLONASS project. Disponível em: <http://www.glonass-ianc.rsa.ru>. Acesso em Dezembro de 2011
- [33] GALILEO project. Disponível em: <http://www.esa.int/esaNA/galileo.html>. Acesso em Dezembro de 2011
- [34] KARP, B.; KUNG, H.T. "GPSR: greedy perimeter stateless routing for wireless networks". In: *ACM Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 243–254, New York, Aug. 2000.
- [35] ALVES, R.; CAMPBELL, I.; COUTO, R. et al., "Redes Veiculares: Princípios, Aplicações e Desafios", In: *Minicursos 27º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC 2009*, pp.199-254, Recife-PE, Brasil, Maio de 2009.
- [36] TOUSSAINT, G.T., "The relative neighbourhood graph of a finite planar set", In: *Pattern Recognition*, Vol. 12, pp. 261-268, 1980.
- [37] LOCHERT, C.; HARTENSTEIN, H.; TIAN, J. et al., "A routing strategy for vehicular ad hoc networks in city environments". In: *IEEE Intelligent Vehicles Symposium*, pp. 156–161. Columbus, Jun. 2003.
- [38] SEET, B.C.; LIU, G.; LEE, B.S. et al., "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications". In: *Proceedings of NETWORKING 2004*, pp. 989–999, Athens, May 2004
- [39] MOEZ, J.; RABAH, M.; SENOUCI, S-M. et al., "GyTAR: improved greedy traffic aware routing protocol for vehicular ad hoc networks in city environments". In: *Proc. 3rd ACM International Workshop on Vehicular Ad Hoc Networks*, Los Angeles, Sep. 2006.
- [40] KÄSEMANN, M.; FÜßLER, H.; HARTENSTEIN, H. et al., "A reactive location service for mobile ad hoc networks", 2002.
- [41] DIJKSTRA, E.W. "A note on two problems in connexion with graphs". In: *Numerische Mathematik*, Vol.1, no.1, pp.269–271, Dec. 1959.
- [42] LIU, L.; WANG, Z.; JEHNG, W.K., "A geographic source routing protocol for traffic sensing in urban environment". In: *IEEE International Conference on Automation Science and Engineering*, pp. 347–352., Arlington, Aug. 2008.
- [43] LI, J.; JANNOTTI, J.; COUTO, D. et al., "A scalable location service for geographic ad hoc routing", In: *ACM/IEEE MOBICOM'2000*, pp. 120-130, New York, 2000.

- [44] R. Pasquini, A flat routing proposal based on the XOR metric and Local Visibility, Ph.D. Thesis - University of Campinas, Campinas, Brazil, 2011.
- [45] BLOOM, B.H., "Space/time trade-offs in hash coding with allowable errors," In: Commun. ACM, vol. 13, no. 7, pp. 422–426, 1970
- [46] TARKOMA, S.;ROTHENBURG, C. E.; LAGERSPETZ, E., "Theory and practice of Bloom Filters for distributed systems". In: IEEE Communications Surveys and Tutorials, Vol. 14, no. 2, Mar. 2012.
- [47] BRODER, A. Z.; MITZENMACHER, M., "Network Applications of Bloom Filters: A Survey," In: Internet Mathematics, vol. 1, no. 4, pp. 636-646, 2003
- [48] CRUZ, E.P.F.; CAMPOS, C.A.V.; PASQUINI, R.; et al., "Performance Analysis of XOR-based routing in urban vehicular ad hoc networks", In: 2012 IEEE Wireless Communications and Networking Conference (WCNC 2012), pp. 2521-2525, Paris, April 2012.
- [49] TraNS, "open source tool for realistic simulations of VANET applications", Disponível em: <http://trans.epfl.ch/>. Acesso em: Setembro de 2011
- [50] KRAJZEWICZ, D.;HERTKORN, G.;WAGNER, P. et al., "Sumo (simulation of urban mobility), an open-source traffic simulation." In: 4th Middle East Symposium on Simulation and Modelling (MESM2002), pages 183–187,Sharjah, Oct. 2002
- [51] U.S. Census Bureau, "TIGER database/Line - Secondedition," Disponível em: <http://www.census.gov/geo/www/tiger>. Acesso em: Setembro de 2011
- [52] "Task 3 Final Report," Identify Intelligent Vehicle Safety Applications Enabled by DSRC, DOT HS 809 859, Mar. 2005. Disponível em: <http://www-nrd.nhtsa.dot.gov/pdf/nrd-12/1665CAMP3web/index.html>. Acesso em: Setembro de 2011
- [53] Information Sciences Institute, "NS-2 network simulator (version 2.33)" Disponível em: [http://nslam.isi.edu/nslam/index.php/Main\\_Page](http://nslam.isi.edu/nslam/index.php/Main_Page). Acesso em: Maio de 2011
- [54] T.S. Rappaport, "Wireless communications, principles and practice", Prentice Hall, 1996.