



UNIVERSIDADE ESTADUAL DO CEARÁ

ALISSON BARBOSA DE SOUZA

**MAV-AODV - UM PROTOCOLO *MULTICAST* BASEADO
EM COLÔNIAS DE FORMIGAS PARA REDES *AD HOC*
VEICULARES**

FORTALEZA - CEARÁ

2012

ALISSON BARBOSA DE SOUZA

**MAV-AODV - UM PROTOCOLO *MULTICAST* BASEADO EM COLÔNIAS DE
FORMIGAS PARA REDES *AD HOC* VEICULARES**

Dissertação apresentada no Curso de Mestrado Acadêmico em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. Joaquim Celestino Júnior

FORTALEZA - CEARÁ

2012

Dados Internacionais de Catalogação na Publicação
Universidade Estadual do Ceará
Biblioteca Central Prof. Antônio Martins Filho

S719m Souza, Alisson Barbosa de Souza
MAV-AODV- um protocolo multicast baseado em colônias
de formigas para redes AD HOC veiculares /Alisson Barbosa de
Souza. – 2012.
90f. : il. color., enc. ; 30 cm.

Dissertação (Mestrado) – Universidade Estadual do Ceará,
Centro de Ciências e Tecnologia, Mestrado Acadêmico em
Ciências da Computação, Fortaleza, 2012.

Orientação: Prof. Dr. Joaquim Celestino Júnior.

1. Vanets. 2. Colônias de formigas. 3. Multicast. I. Título.

CDD: 001.6

ALISSON BARBOSA DE SOUZA

MAV-AODV - UM PROTOCOLO *MULTICAST* BASEADO EM COLÔNIAS DE FORMIGAS PARA REDES *AD HOC* VEICULARES

Dissertação apresentada no Curso de Mestrado Acadêmico em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial para obtenção do grau de Mestre.

Área de Concentração: Ciência da Computação

Aprovada em: 26/03/2012.

BANCA EXAMINADORA

Prof. Dr. Joaquim Celestino Júnior (Orientador)
Universidade Estadual do Ceará – UECE

Prof. Dr. Marcial Porto Fernandez
Universidade Estadual do Ceará – UECE

Prof. Dr. Gustavo Augusto Lima de Campos
Universidade Estadual do Ceará – UECE

Prof. Dr. André Castelo Branco Soares
Universidade Federal do Piauí – UFPI

AGRADECIMENTOS

Primeiramente, agradeço ao Todo-Poderoso Deus Jeová, e a seu filho Jesus Cristo, pela vida, por todas as boas dádivas, pela esperança de vida futura, enfim, por tudo.

Agradeço ao meu pai, Valter, minha mãe, Ana, meu irmão, Gleidson, e minha cunhada, Carolina, pelo apoio contínuo em todos os momentos. Na vida, sempre passamos por momentos bons e ruins, mas sempre estaremos unidos e nos encorajando. Sou muito grato de estar rodeado de pessoas como vocês.

Agradeço também ao professor Joaquim Celestino Júnior, meu orientador desde 2006, pela orientação, confiança e respeito. Sem sua dedicação e esforço, este trabalho não poderia ter sido realizado.

Sou grato a todos os amigos que fiz, nos anos que estudei na UECE. Em especial, agradeço ao Sérgio Vieira, Filipe Maciel, Sérgio Luís, Felipe Xavier, Rudy Matela, Tiago Carneiro, Thiago Queiroz, Luís Ribeiro e Alysson Oliveira. A amizade e a ajuda de vocês foram essenciais nessa jornada.

A todas as pessoas que, de uma forma ou de outra, contribuíram para a realização deste trabalho.

“Manifesta-se um espírito nas leis do Universo — um espírito vastamente superior ao do homem, e diante do qual nós, com os nossos modestos poderes, devemos sentir-nos humildes.”
Albert Einstein (1879 - 1955)

RESUMO

Vehicle Ad Hoc Networks (VANETs) são redes que confiam nos próprios veículos para fornecerem sistemas de prevenção de acidentes e entretenimento para os passageiros dos veículos. Tais redes possuem características bem peculiares como restrições de mobilidade e alta velocidade dos nós. Neste trabalho, analisamos a eficiência do MAODV, um bem conhecido protocolo que constrói árvores sob demanda para conectar membros de grupos *multicast* em *Mobile Ad Hoc Networks* (MANETs). Sabendo que para manter uma estrutura em árvore em ambientes veiculares é um desafio devido às frequentes desconexões de enlaces, desenvolvemos o protocolo MAV-AODV que faz uso de informações de mobilidade veicular para construir árvores estáveis. O MAV-AODV usa mecanismos baseados em Colônias de Formigas para otimizar a construção e manutenção das árvores *multicast*. As mensagens do protocolo agem como agentes bioinspirados (formigas) ao depositar feromônio para avaliar a atratividade de cada rota. Foram realizados experimentos via simulação para analisar a eficiência de nosso protocolo.

Palavras-chave: VANETs. Colônia de Formigas. *Multicast*.

ABSTRACT

VANETs are networks that rely on their own vehicles to provide accident prevention systems and entertainment for passengers of vehicles. Such networks have clear distinguishing features such as restricted mobility and high velocity of nodes. In this thesis, we analyze the efficiency of MAODV, a well known protocol that builds trees on demand to connect multicast group members. Knowing that to maintain a tree structure in vehicular environments is a challenge due to frequent link disconnections, we developed the MAV-AODV protocol that makes use of vehicular mobility information to build stable trees. The MAV-AODV uses mechanisms based in Ant Colonies to optimize the construction and maintenance of multicast trees. The protocol messages act as bio-inspired agents (ants) depositing pheromone to evaluate the attractiveness of each route. Experiments were carried out via simulation to analyze the efficiency of our protocol.

Keywords: VANETs. Ant Colony. Multicast.

LISTA DE FIGURAS

Figura 1	Exemplo de uma VANET (KARIM, 2008)	22
Figura 2	Estrada com 3 veículos mostrando seus alcances de transmissão/recepção. O veículo B é vizinho de A e C, mas A e C não são vizinhos. (CORREIA; CELESTINO; CHERKAOUI, 2011)	24
Figura 3	Espectro DSRC (JIANG; DELGROSSI, 2008).	25
Figura 4	Comutação de Canais DSRC definida pela arquitetura WAVE (LI, 2010). ...	25
Figura 5	Arquitetura WAVE (ALVES et al., 2009).	26
Figura 6	Taxonomia dos Protocolos <i>Multicast</i> em Redes <i>Ad Hoc</i> (BADARNEH; KADOCH, 2009)	32
Figura 7	Estrutura <i>Multicast</i> em Árvore	33
Figura 8	Estrutura <i>Multicast</i> em Malha	33
Figura 9	Processo de Junção <i>Multicast</i> do MAODV (MUÑOZ, 2009)	34
Figura 10	Estrutura em Malha do PUMA e Propagação de Anúncios <i>Multicast</i> (VAISHAMPAYAN; GARCIA-LUNA-ACEVES, 2004)	35
Figura 11	Lista de Conectividade do Nó 6 (VAISHAMPAYAN; GARCIA-LUNA-ACEVES, 2004)	35
Figura 12	Criação de estrutura em malha do protocolo ODMRP (MUÑOZ, 2009)	36
Figura 13	Envio de RREQ pelo protocolo ROVER (KIHL et al., 2007)	38
Figura 14	Envio de RREP pelo protocolo ROVER (KIHL et al., 2007)	38

Figura 15	Protocolo DRG (LIN; CHEN; LEE, 2010)	39
Figura 16	Protocolo IVG (LIN; CHEN; LEE, 2010)	40
Figura 17	Protocolo <i>Mobicast</i> (LIN; CHEN; LEE, 2010)	41
Figura 18	Colônia de Formigas	42
Figura 19	Um Esquema de Roteamento <i>Multicast</i> (SEBASTIAN et al., 2010)	50
Figura 20	Alcance de Transmissão/Recepção do Veículo <i>x</i> .	55
Figura 21	Formato da Mensagem <i>Beacon</i> do Protocolo Implementado neste Trabalho.	56
Figura 22	Formato da Mensagem RREQ.	63
Figura 23	Formato da Mensagem RREP.	63
Figura 24	Exemplo de Requisição de Rota.	65
Figura 25	Exemplo de Resposta de Rota.	67
Figura 26	Exemplo do Processo de Escolha de RREPs por Roleta.	68
Figura 27	Exemplo de um Nó Interior numa <i>Árvore Multicast</i> .	69
Figura 28	Exemplo de Criação de Ramo de <i>Árvore</i> .	70
Figura 29	Cenário Avaliado.	72
Figura 30	Arquitetura Básica do ns-2 (ISSARIYAKUL; HOSSAIN, 2008).	73
Figura 31	Atraso Máximo Fim-a-Fim (Manhattan)	76

Figura 32	Atraso Máximo Fim-a-Fim (Auto-Estrada)	76
Figura 33	Variação do Atraso Fim-a-Fim (Manhattan)	77
Figura 34	Variação do Atraso Fim-a-Fim (Auto-Estrada)	77
Figura 35	<i>Overhead</i> de Roteamento (Manhattan)	78
Figura 36	<i>Overhead</i> de Roteamento (Auto-Estrada)	78
Figura 37	Taxa de Entrega de Pacotes (Manhattan)	79
Figura 38	Taxa de Entrega de Pacotes (Auto-Estrada)	79
Figura 39	Redundância de Pacotes (Manhattan)	80
Figura 40	Redundância de Pacotes (Auto-Estrada)	80

LISTA DE TABELAS

Tabela 1	Resumo dos Trabalhos Relacionados.	52
Tabela 2	Resumo dos Trabalhos Relacionados.	53
Tabela 3	Parâmetros Usados para Predição de Posicionamento.	59
Tabela 4	Parâmetros Usados para Calcular o Tempo de Vida de Enlaces.	60
Tabela 5	Tabela de Rotas do MAV-AODV/MAODV.	61
Tabela 6	Tabela de Rotas <i>Multicast</i> do MAV-AODV	62
Tabela 7	Parâmetros Usados nos Experimentos.	74

LISTA DE SIGLAS

ACO	<i>Ant Colony Optimization</i>
AODV	<i>Ad hoc On-demand Distance Vector routing protocol</i>
DRG	<i>Distributed Robust Geocast</i>
DSRC	<i>Dedicated Short Range Communications</i>
DYMO	<i>DYnamic Manet On-demand routing protocol</i>
GRPH	<i>GRouP Hello</i>
ITS	<i>Intelligent Transportation Systems</i>
IVG	<i>Inter-Vehicles Geocast</i>
MACT	<i>Multicast ACTivation</i>
MANETs	<i>Mobile Ad Hoc Networks</i>
MAODV	<i>Multicast Ad hoc On-demand Distance Vector routing protocol</i>
MAV-AODV	<i>Multicast with mechanisms of Ant colony optimization for Vanets based mA-ODV protocol</i>
NS-2	<i>Network Simulator - version 2</i>
OBU	<i>On-Board Unit</i>
ODMRP	<i>On-Demand Multicast Routing Protocol</i>
ROVER	<i>RObust VEhicular Routing</i>
RREP	<i>Route REPLY</i>
RREQ	<i>Route REQuest</i>
RSU	<i>Roadside Unit</i>
V2I	<i>Vehicle-to-Infrastructure</i>
V2V	<i>Vehicle-to-Vehicle</i>
VANETs	<i>Vehicular Ad Hoc Networks</i>
WAVE	<i>Wireless Access in the Vehicular Enviroments</i>

SUMÁRIO

1 INTRODUÇÃO	16
1.1 CONTEXTO E MOTIVAÇÃO	16
1.2 OBJETIVOS E CONTRIBUIÇÕES	18
1.3 METODOLOGIA	19
1.4 ORGANIZAÇÃO DO TRABALHO	20
2 FUNDAMENTAÇÃO TEÓRICA	21
2.1 <i>VEHICULAR AD HOC NETWORKS</i>	21
2.1.1 Introdução	21
2.1.2 Definição	21
2.1.3 Características	22
2.1.4 Padrões e Arquiteturas	24
2.1.4.1 <i>Dedicated Short Range Communications (DSRC)</i>	24
2.1.4.2 <i>Wireless Access in the Vehicular Enviroments (WAVE)</i>	25
2.1.5 Aplicações	28
2.1.5.1 Aplicações de Segurança	28
2.1.5.2 Aplicações de Conforto	29
2.1.6 Por que usar VANETs?	30
2.1.7 Desafios	30
2.2 <i>ROTEAMENTO MULTICAST</i>	31
2.2.1 Taxonomia dos Protocolos <i>Multicast</i>	32
2.2.2 Protocolos de Roteamento <i>Multicast</i> em MANETs	33
2.2.2.1 <i>Multicast Ad hoc On-demand Distance Vector routing protocol (MAODV)</i>	33
2.2.2.2 <i>Protocol for Unified Multicasting through Announcements (PUMA)</i>	34
2.2.2.3 <i>On-Demand Multicast Routing Protocol (ODMRP)</i>	36
2.2.3 <i>Multicast X Geocast</i>	37
2.2.4 Protocolos de Roteamento <i>Geocast</i> em VANETs	38
2.2.4.1 <i>RObust VEhicular Routing (ROVER)</i>	38
2.2.4.2 <i>Distributed Robust Geocast (DRG)</i>	39

2.2.4.3 <i>Inter-Vehicles Geocast (IVG)</i>	39
2.2.4.4 <i>Mobicast</i>	40
2.3 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS	40
2.3.1 Inspiração Biológica	41
2.3.2 A Metaheurística Otimização por Colônia de Formigas	42
2.3.3 Por que usar Otimização por Colônia de Formigas?	44
3 TRABALHOS RELACIONADOS	45
3.1 TRABALHOS QUE PROPUSERAM APRIMORAMENTOS AO MAODV	45
3.2 UTILIZAÇÃO DE OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS EM PROBLE- MAS DE ROTEAMENTO EM REDES <i>AD HOC</i> MÓVEIS	47
3.3 OUTROS TRABALHOS QUE CONTRIBUÍRAM PARA A PROPOSTA	49
3.4 CONSIDERAÇÕES FINAIS	51
4 O PROTOCOLO DE ROTEAMENTO <i>MULTICAST</i> MAV-AODV	54
4.1 MENSAGENS <i>BEACONS</i>	55
4.2 PREDIÇÃO DE MOBILIDADE	56
4.2.1 Distância entre Veículos e Modelos de Rádio Propagação	57
4.2.2 Predição de Posicionamento de Vizinhança em um Dado Momento	58
4.2.3 Estimativa de Tempo de Vida de Enlace	59
4.3 ESPECIFICAÇÕES DO PROTOCOLO DE ROTEAMENTO <i>MULTICAST</i> MAV- AODV	61
4.3.1 Alterações sobre o MAODV	61
4.3.2 Tabelas de Roteamento	61
4.3.3 Formato das Mensagens de Roteamento	62
4.3.3.1 RREQ - <i>Route REQuest</i>	62
4.3.3.2 RREP - <i>Route REPLY</i>	63
4.3.4 Geração de Mensagem de Requisição de Rota	64
4.3.5 Estabelecimento de Rota Reversa	66
4.3.6 Geração de Mensagem de Resposta de Rota	66
4.3.7 Mensagens <i>Group Hello</i>	68
4.3.8 Nó Interior da Árvore <i>Multicast</i> e Tratamento de Redundâncias	69
4.3.9 Ativação de Rota <i>Multicast</i>	70

5 SIMULAÇÃO E ANÁLISE DE RESULTADOS	72
5.1 CENÁRIOS	72
5.2 SIMULADOR	72
5.3 DETALHES DOS EXPERIMENTOS REALIZADOS	73
5.4 TRÁFEGO DE DADOS E MENSAGENS <i>BEACONS</i>	74
5.5 ANÁLISE DE RESULTADOS	75
5.5.1 Atraso Máximo Fim-a-Fim	75
5.5.2 Variação do Atraso Fim-a-Fim	76
5.5.3 <i>Overhead</i> de Roteamento	77
5.5.4 Taxa de Entrega de Pacotes	78
5.5.5 Redundância de Pacotes	79
6 CONCLUSÃO E TRABALHOS FUTUROS	81
REFERÊNCIAS	83

1 INTRODUÇÃO

Este capítulo contextualiza, motiva, esclarece os objetivos e contribuições do trabalho, explica a metodologia aplicada e mostra como está definida a organização do trabalho.

1.1 Contexto e Motivação

Vehicular Ad Hoc Networks (VANETs), uma subclasse das *Mobile Ad Hoc Networks* (MANETs), são uma abordagem promissora para os Sistemas de Transportes Inteligentes (ITS). Essas redes não têm uma estrutura fixa e confiam nos próprios veículos para fornecer as suas funcionalidades. Tais redes permitem dois tipos de comunicação: veículo-veículo e veículo-RSU (*Roadside Unit* - dispositivo fixo localizado nos acostamentos das pistas), também chamada de veículo-infraestrutura. Além disso, por ter um custo mais acessível, ter atrasos de roteamento menores e por poder trocar informações com outros nós, mesmo em locais sem infraestrutura, uma VANET é mais apropriada para distribuir informações em cenários veiculares (KARIM, 2008).

Há alguns anos, indústrias e governos têm investido em tecnologias para tornar o conceito de VANETs em realidade, focando em dois principais aspectos: segurança no trânsito e conforto para os passageiros dos veículos.

Nas VANETs, os veículos trocam, entre si, mensagens que informam as condições do ambiente em que os veículos estão. Tais informações podem ajudar a evitar acidentes e prover mais segurança aos veículos (YOUSEFI; MOUSAVI; FATHY, 2006). Por exemplo, um veículo pode informar aos outros que determinado trecho da estrada está deteriorado ou escorregadio. Uma freada brusca pode ser detectada pelo próprio veículo que está freando e pode ser transmitida para os veículos que vem logo atrás. Esse sistema de alerta de emergência pode evitar engavetamentos. Além disso, essas mensagens de alerta podem se propagar mais rapidamente do que os indicadores visuais percebidos pelo motorista, como a luz de freio do veículo seguinte (BISWAS; TATCHIKOU; DION, 2006). Também são possíveis assistência de mudança de faixa e auxílio a ultrapassagens. Baseado nas informações dos veículos vizinhos e do próprio veículo, esse sistema pode informar ao motorista se é recomendado, ou não, mudar de faixa ou tentar uma ultrapassagem.

Outros tipos de aplicações são as de conforto, por exemplo: otimização de rotas para um destino, sistema de informação de tráfego, informações climáticas, de preço, localização de restaurantes, dentre outras (YOUSEFI; MOUSAVI; FATHY, 2006). Um outro exemplo é o pagamento de pedágio de maneira automática, sem que o veículo pare e sem que haja qualquer intervenção do motorista. Além disso, como os veículos também se comunicam com os dispositivos fixos localizados nos acostamentos ou canteiros das pistas, também seria possível os

passageiros acessarem a Internet através desta rede.

Embora seja um tipo específico de MANETs, VANETs possuem características que são dramaticamente diferentes das MANETs tradicionais. Tais características implicam em novos desafios a serem enfrentados. Como os nós das VANETs são veículos, uma característica marcante é que a velocidade dos nós nas VANETs é muito maior do que nas MANETs. Essa diferença em termos de velocidade dos nós faz com que uma rede veicular seja muito mais instável do que as MANETs convencionais. É possível que num ambiente veicular, uma rede esteja formada por apenas alguns segundos.

Uma outra diferença está na forma de mobilidade dos nós. Nas redes *ad hoc* tradicionais, geralmente, os nós se movem de maneira aleatória. Já nas redes *ad hoc* veiculares, os nós se movimentam com um certo nível de previsibilidade, porque os nós são restritos aos limites físicos e legais (limites de velocidade, sentido da via etc) das pistas. Além disso, os nós se movimentam de acordo com o objetivo e comportamento dos motoristas. Dessa forma, um nó pode dobrar em uma rua próxima, acelerar ou desacelerar, frear bruscamente, fazer um retorno etc.

Também, em alguns tipos de MANETs, existe uma preocupação com o tempo de vida dos dispositivos que fornecem a comunicação *ad hoc*. Em VANETs, não há essa preocupação pois os dispositivos usufruem da energia das baterias dos veículos.

Essas características específicas das redes veiculares, fazem com que os protocolos de roteamento feitos para as MANETs não sirvam de maneira satisfatória para as VANETs. Então, existe uma necessidade de especificar um protocolo totalmente voltado para as VANETs ou adaptar os protocolos das MANETs para as VANETs. Härr, Bonnet e Filali (2008) mostra que, enquanto a primeira opção fornece soluções bem especializadas e eficientes, a segunda opção fornece interoperabilidade e universalidade, elementos fundamentais para os parceiros industriais que investem nas VANETs.

Um outro tópico abordado neste trabalho é o roteamento *multicast*. O roteamento *multicast* é a transmissão de pacotes para um grupo de nós, identificados por um único endereço de destino. Existe uma faixa de IPs reservada para *multicast*. Dessa maneira, os dispositivos de rede reconhecem os endereços IPs que são usados por aplicações *multicast*. Algumas características do roteamento *multicast* são: utiliza-se o protocolo UDP na camada de transporte, devido à natureza *unicast* do protocolo TCP; nós podem entrar ou deixar o grupo *multicast* quando quiserem; um nó pode ser membro de mais de um grupo ao mesmo tempo (JUNHAI; LIU; DANXIA, 2008; KUNZ, 2003).

Tal tipo de roteamento é o método mais eficiente de comunicação em grupo, superando o roteamento *unicast* e *broadcast*. Com o desenvolvimento de mais tecnologias em redes sem fio, os fornecedores de conteúdo e serviço estão cada vez mais interessados em formas de

comunicação em grupo (VARSHNEY, 2002).

Existem várias aplicações que requerem *multicast*. Um veículo que precisa transmitir informações urgentes para somente alguns nós da rede. Uma ambulância que precisa abrir caminho para agilizar seu atendimento. Jogos interativos entre passageiros de veículos. Anúncios de lojas para um determinado grupo de veículos.

O roteamento *multicast* em VANETs é muito desafiador devido às rápidas mudanças topológicas causadas pela mobilidade dos veículos (MOHAPATRA; LI; GUI, 2005; MUÑOZ, 2009; SEBASTIAN et al., 2010). Além disso, a alta velocidade dos nós, a mudança de densidade dos ambientes veiculares e outras características próprias das VANETs causam muitos problemas aos protocolos *multicast*. Por isso é tão importante o estudo de um protocolo *multicast* adaptável ao ambiente veicular.

Neste trabalho, adotamos a estrutura *multicast* baseada em árvore porque comparada com as outras estruturas, a baseada em árvore fornece uma considerável eficiência no encaminhamento dos dados (evitando redundância de dados como na estrutura baseada em malha). Porém, é uma estrutura que não fornece robustez porque quando os nós se movimentam, os caminhos podem facilmente ser quebrados sem ter caminhos alternativos (JUNHAI; LIU; DANXIA, 2008; YANG; WU, 2005). Por isso, tentamos usar a objetividade da estrutura baseada em árvore e dar robustez a essa estrutura por escolher caminhos com maiores tempos de vida, como mostrado nas seções seguintes.

Além disso, conforme apresentado nas próximas seções, existem poucos trabalhos na literatura que tratam do roteamento *multicast* em VANETs. Dessa forma, existe a necessidade de um protocolo *multicast* eficiente que se adapte às características das VANETS, mantendo baixos níveis de atraso e de *overhead* de roteamento e tendo uma alta taxa de entrega de pacotes, ou seja, um protocolo que mantenha um bom desempenho.

Para obter o desempenho desejado, utilizamos, em nosso protocolo, um mecanismo bioinspirado baseado em colônia de formigas. A ideia desse mecanismo vem da metaheurística Otimização por Colônia de Formigas (DORIGO, 1992; DORIGO; MANIEZZO; COLORNI, 1996; DORIGO et al., 1991). A fonte de inspiração dessa metaheurística foi o comportamento forrageiro das formigas reais. Tal metaheurística foi escolhida pelos seguintes motivos: é adaptável a topologias dinâmicas, permite avaliar a qualidade dos enlaces, tem suporte para múltiplos caminhos, pode encontrar soluções em tempo viável e não precisa de controle central (GUNES; SORGES; BOUAZIZI, 2002; GLOVER; KOCHENBERGER, 2003; TIRUCHEN-GODE, 2009).

1.2 Objetivos e Contribuições

O objetivo deste trabalho é propor um protocolo baseado em colônia de formigas para tornar os protocolos *multicast* mais adaptáveis e eficientes em ambientes veiculares. Comprovamos isso em simulações com nosso protocolo (MAV-AODV - Multicast with Ant Colony Optimization for Vanets based on MAODV) que utiliza os procedimentos reativos do protocolo AODV (PERKINS et al., 2003) e do MAODV (YANG; WU, 2005; ROYER; PERKINS, 1999) para construir uma árvore e conectar os membros de um grupo *multicast*. Além disso, alguns mecanismos da metaheurística Otimização por Colônia de Formigas são utilizados para construir e manter a árvore *multicast*.

O MAV-AODV pode ser aplicado aos cenários veiculares porque leva em conta a mobilidade dos nós e, portanto, pode escolher rotas mais estáveis de acordo com o tempo de vida de cada rota. O tempo de vida de cada rota é calculado pelo tempo de vida de cada enlace da rota. A estimativa do tempo de vida de cada enlace é feita por um protocolo de mensagens *beacons*, que desenvolvemos e que atua juntamente com o MAV-AODV.

A principal diferença do MAV-AODV para outros protocolos, encontrados na literatura, é que o MAV-AODV é um protocolo *multicast* tradicional (não *geocast*) adaptado aos cenários veiculares. O MAV-AODV pode formar grupos *multicast* precisando apenas do interesse dos nós, independente da posição geográfica dos veículos. Além do mais, os princípios baseados em colônia de formigas ajudam a otimizar o desempenho do protocolo e faz com que ele seja adaptável às características das VANETs, escolhendo rotas mais estáveis.

Com respeito às contribuições deste trabalho, as principais são:

- Método para fornecer eficiência, robustez, e adaptação de estruturas *multicast* em cenários veiculares.
- Implementação de um protocolo de roteamento *multicast* tradicional adaptado aos cenários veiculares, inclusive utilizando protocolos-padrões de VANETs (como o 802.11p). Na literatura de comunicação em grupo, e tirando protocolos *broadcast* e *geocast*, não existem protocolos *multicast* tradicionais feitos para redes *ad hoc* veiculares.
- Mecanismos bioinspirados. Utilizamos princípios da Otimização por Colônia de Formigas para construção da árvore *multicast*. Essa construção da árvore é feita utilizando as rotas mais estáveis (curtas e com suficientes tempos de vida) e que proporcionam boas métricas de desempenho.

1.3 Metodologia

Os experimentos foram realizados utilizando o simulador NS-2 (*Network Simulator - versão 2.34*) (BERKELEY; ISI, 1998). Os experimentos foram executados a partir de arquivos de simulação em linguagem OTcl (WETHERALL, 1995). O tráfego veicular foi gerado pelo *Vehicular Network Movement Generator* (NZOUONTA, 2008), que implementa os modelos de mobilidade propostos por Gipps (GIPPS, 1981, 1986).

As informações obtidas após a simulação foram tratadas para a análise de desempenho com respeito às seguintes métricas: atraso fim-a-fim mínimo e máximo (MAIA; CELESTINO, 2009); variação do atraso fim-a-fim; overhead de roteamento; e taxa de entrega de pacotes. Para avaliar o desempenho de nosso mecanismo baseado em colônia de formigas, testamos o MAV-AODV num cenário urbano e comparamos com um bem conhecido protocolo *multicast* para MANETs, o MAODV.

1.4 Organização do Trabalho

Este trabalho é organizado como descrito a seguir.

O capítulo 2 apresenta a fundamentação teórica sobre os principais temas abordados neste trabalho. O capítulo 3 faz um levantamento dos principais trabalhos relacionados. O capítulo 4 descreve o protocolo *multicast* MAV-AODV, o cálculo de estimativa de tempos de vida de enlaces para ajudar no processo de escolha de rotas e um protocolo de mensagens *beacons* que desenvolvemos. No capítulo 5, o simulador, o cenário de testes, os métodos experimentais e a análise de resultados são detalhados. O capítulo 6 mostra as conclusões e possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

A seguir, será apresentada a fundamentação teórica dos principais temas abordados neste trabalho.

2.1 *Vehicular Ad Hoc Networks*

2.1.1 Introdução

Os veículos automotores vêm incorporando diferentes avanços tecnológicos que melhoram a experiência do condutor e dos passageiros. A evolução tecnológica converge, agora, para sistemas de comunicação que possibilitem a interação entre diferentes veículos. O objetivo principal desses sistemas é prover a comunicação de usuários móveis e oferecer as condições necessárias para que aplicações com diferentes requisitos sejam atendidas satisfatoriamente. Os sistemas de comunicação que possuem veículos como transmissores e receptores de dados formam as chamadas redes *ad hoc* veiculares (ALVES et al., 2009).

As redes *ad hoc* veiculares são promissoras porque podem oferecer segurança no trânsito e vários tipos de serviços aos motoristas e passageiros. Isso pode ser percebido pelo crescente interesse da comunidade científica e pelo interesse das indústrias automotivas. Além disso, órgãos governamentais dos Estados Unidos, União Europeia e Japão, e órgãos de padronização de todo o mundo (IEEE (IEEE, 2011), por exemplo), demonstram interesse e empenho pelas redes veiculares por formarem grupos de estudo, fornecerem padrões, documentos, análises e produtos relacionados a esse tema.

As próximas seções discutem os principais aspectos das redes *ad hoc* veiculares.

2.1.2 Definição

Vehicular Ad Hoc Networks ou redes *ad hoc* Veiculares (VANETs), uma subclasse das *Mobile Ad Hoc Networks* (MANETs), são uma abordagem promissora para os Sistemas de Transportes Inteligentes (ITS). Essas redes não têm uma estrutura fixa e confiam nos próprios veículos para fornecer as suas funcionalidades.

Além disso, conforme a Figura 1, as VANETs possuem dois tipos de comunicação: *Vehicle-to-Vehicle* (V2V) e *Vehicle-to-Infrastructure* (V2I), também chamada de *Vehicle-to-RSU*. Elas também possuem os seguintes principais tipos de dispositivos: *On-Board Units* (OBUs) - unidades de bordo localizadas dentro do veículo que fazem com que eles se comportem como nós *ad hoc*; *Roadside Units* (RSUs) (não são obrigatórios para a existência de uma VANET) - unidades fixas que proveem conectividade à Internet e são localizadas em can-

teiros ou calçadas das vias de trânsito; sensores; *Global Positioning System* (GPS).

A comunicação V2V permite que uma rede VANET seja formada em qualquer lugar, bastando apenas que haja veículos equipados com OBUs e que eles estejam suficientemente próximos uns dos outros. Porém, a conectividade é seriamente afetada por causa de constantes desconexões da rede decorrentes de distanciamentos entre os veículos.

Já na comunicação V2I, as RSUs podem ser dispostas estrategicamente para evitar o problema de frequentes desconexões. Entretanto, fornecer a comunicação V2I pode ser oneroso devido ao elevado custo de instalação dessa infraestrutura. Também, ter uma comunicação V2I não é garantia de conectividade total porque uma RSU pode estar se comunicando com vários veículos. Portanto, a oportunidade de transmissão/recepção de cada veículo é limitada.

A Figura 1 mostra um exemplo clássico de uma VANET. Ela apresenta os dois tipos de comunicação possíveis: V2V e V2I (veículo se comunicando com a RSU). A Figura também apresenta um evento de emergência (colisão entre veículos) que as VANETs se encarregam de transmitir para outros veículos para evitar novas colisões.

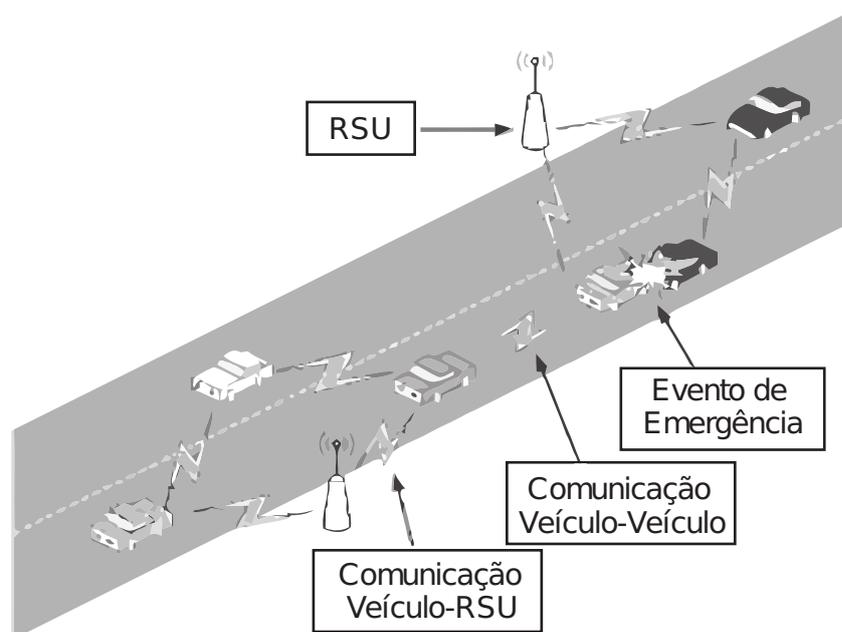


Figura 1: Exemplo de uma VANET (KARIM, 2008)

2.1.3 Características

Embora as VANETs sejam um tipo específico das MANETs, as VANETs possuem características bem distintas das MANETs genéricas. Conforme Yousefi, Mousavi e Fathy (2006) tais características têm importantes implicações para o projeto de um protocolo VANET. Por conta disso, muitos protocolos desenvolvidos para MANETs não têm um desempenho adequado em VANETs.

Conforme Yousefi, Mousavi e Fathy (2006), em VANETs, as principais diferenças das MANETs são:

1. Os nós se movem em altas velocidades pelo fato de serem, na verdade, os próprios veículos. Em alguns casos, os veículos podem se mover a velocidades superiores a 200 km/h. Isso ocasiona instabilidade na rede fazendo com que a rede permaneça conectada por apenas poucos segundos. Além disso, a topologia da rede muda rapidamente devido, entre outros fatores, à alta velocidade relativa entre os veículos.
2. A conectividade é altamente dependente do cenário, ou seja, para se manter conectada, uma rede precisa que os seus nós estejam uns sob o alcance de transmissão/recepção dos outros (Figura 2). Nas VANETs, como os nós podem dobrar numa rua, mudar de direção, efetuar outras manobras bruscas ou, por um outro motivo, saírem do alcance de transmissão/recepção de outros nós, a frequente fragmentação da rede é uma outra característica importante.
3. VANETs possuem um pequeno diâmetro efetivo da rede. Esse parâmetro é proporcional à quantidade de nós conectados e à quantidade de tempo que eles ficam conectados. De fato, nas VANETs, poucos nós permanecem muito tempo conectados. A maioria dos nós se desconectam por se afastarem dos outros nós e, assim, permanecem pouco tempo conectados à rede. Isso faz com que o alcance geral de transmissão/recepção da rede (diâmetro efetivo da rede), na maioria das vezes, seja pequeno.
4. Não há significativas restrições de energia, ao contrário de redes de sensores e de outros tipos de redes móveis em que o tempo de vida das baterias são motivo de séria preocupação. Em VANETs, não há essa preocupação pois os dispositivos usufruem da energia das baterias dos veículos.
5. Alta capacidade computacional das OBUs.
6. Em ambientes urbanos ou em entradas/saídas de rodovias, a quantidade de nós conectados a uma mesma rede pode ser muito grande. Portanto, há uma necessidade em termos de escalabilidade para os protocolos VANETs.
7. Densidade variável da rede. Em um dado momento, a rede pode estar esparsa, com poucos nós conectados. Por exemplo, numa estrada rural do interior do Brasil. Em outro momento, a rede pode estar muito densa, com uma grande quantidade de nós conectados. Isso pode acontecer num congestionamento quilométrico numa grande avenida de uma metrópole.
8. A topologia da rede pode ser afetada pelo comportamento dos motoristas. Além disso, a divulgação de uma mensagem, entre os veículos, também pode fazer com que os motoristas tenham uma determinada reação e a topologia da rede mude outra vez.

9. Nas MANETs, os nós se movem de maneira aleatória, sem nenhum padrão determinado. Porém, nas VANETs, há uma certa previsibilidade do movimento dos nós porque eles tem que seguir os padrões das vias de trânsito, como: sentido da via, semáforos, limites de velocidade, limites físicos de largura da via etc.

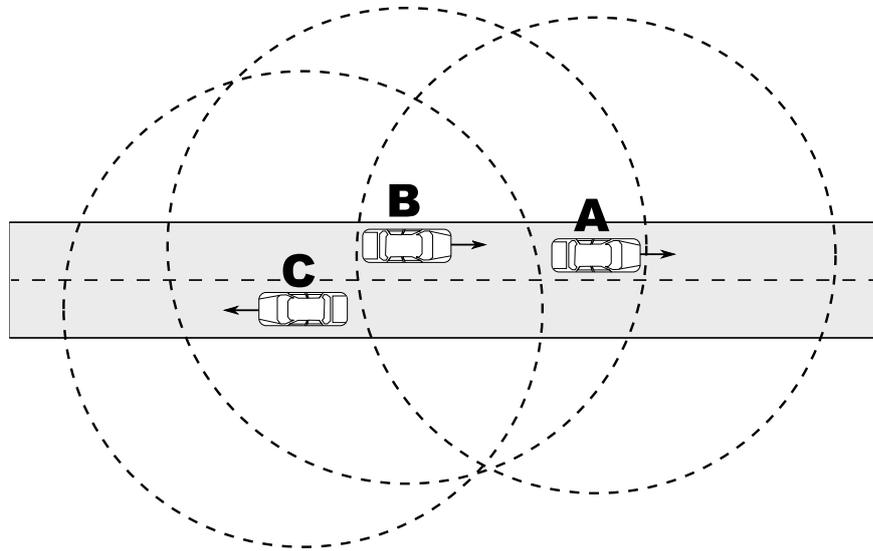


Figura 2: Estrada com 3 veículos mostrando seus alcances de transmissão/recepção. O veículo B é vizinho de A e C, mas A e C não são vizinhos. (CORREIA; CELESTINO; CHERKAOUI, 2011)

2.1.4 Padrões e Arquiteturas

A seguir e de acordo com Alves et al. (2009), serão descritos os padrões mais adotados e reconhecidos nas redes veiculares.

2.1.4.1 *Dedicated Short Range Communications (DSRC)*

Dedicated Short Range Communications (DSRC) é uma padronização de um espectro de banda nos Estados Unidos (JIANG; DELGROSSI, 2008). Em 1991, o congresso americano ordenou a criação de um programa para sistemas veiculares que tinha como principais objetivos aumentar a segurança, reduzir a poluição, melhorar o problema de congestionamentos etc, que seriam usados para a infraestrutura de transportes da nação. Em 1996, o Departamento de Transportes do Estados Unidos da América (DOT) desenvolveu um framework conhecido como NITSA (*National Intelligent Transportation Systems Architecture*) que planejaria, definiria e integraria o sistemas de transportes americanos. O NITSA reconheceu os dispositivos sem fio como fundamentais para um bom funcionamento de transportes inteligentes. Ao mesmo tempo, algumas aplicações de pedágio foram feitas usando um espectro entre 902 MHz e 928 MHz. Entretanto, outro espectro acabou se tornando o padrão. Em 1999, a Comissão de Comunicação

Federal (FCC) dos Estados Unidos alocou 75MHz de espectro da faixa de 5.85 - 5.925 GHz para ser usado exclusivamente para comunicações veículo-veículo ou veículo-infraestrutura (UZCA-TEGUI; ACOSTA-MARUM, 2009). Esse espectro tinha o objetivo principal de fornecer comunicações dedicadas de curto alcance ou DSRC, que acabou se tornando o padrão mais aceito em VANETs.

Como mostrado na Figura 3, o espectro DSRC é estruturado em 7 canais de 10 MHz. O canal 178 é o canal de controle (CCH - *Control CHannel*), que é restrito somente para comunicações de segurança. Só passam, por esse canal, mensagens curtas de alta prioridade (mensagens do protocolo WSMP - *Wave Short Message Protocol* - ver Figura 5) e dados de gerenciamento. Os dois canais das extremidades são reservados para usos especiais. Os outros são canais de serviço (SCH - *Service CHannels*), disponíveis para uso em comunicações de segurança e de conforto.

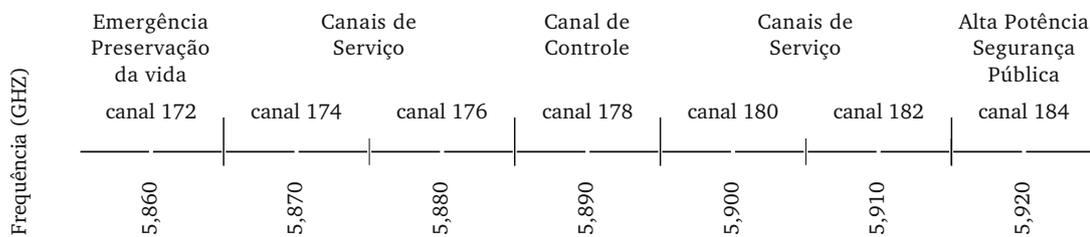


Figura 3: Espectro DSRC (JIANG; DELGROSSI, 2008).

Além disso, a comutação de canais, definida nos padrões da arquitetura WAVE (*Wireless Access in the Vehicular Enviroments*) (IEEE, 2011), é feita por divisão de frequência e de tempo. Períodos repetitivos de 100 ms são dados. Conforme a Figura 4, a cada 100 ms, 50 ms são alocados para o canal de controle (CCH) e outros 50 ms são alocados para os canais de serviços (SCH), incluindo 4 ms de intervalo de guarda para a comutação entre os canais CCH e SCHs (LI, 2010).

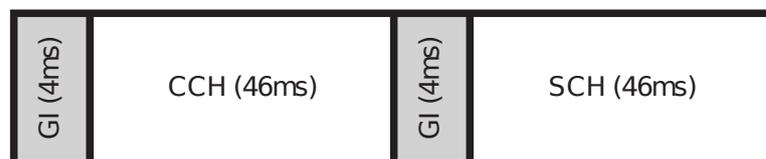


Figura 4: Comutação de Canais DSRC definida pela arquitetura WAVE (LI, 2010).

Vale ressaltar ainda, que a arquitetura WAVE usa o padrão DSRC.

2.1.4.2 *Wireless Access in the Vehicular Enviroments* (WAVE)

Em 2004, o IEEE (*Institute of Electrical and Electronics Engineers*) iniciou a padronização das comunicações em redes veiculares. Uma dessas padronizações, a arquitetura WAVE

(*Wireless Access in the Vehicular Environment*) é definida em cinco documentos: IEEE 1609.1, IEEE 1609.2, IEEE 1609.3, IEEE 1609.4 e IEEE 802.11p (IEEE, 2011; LI, 2010; UZCATEGUI; ACOSTA-MARUM, 2009; HARTENSTEIN; LABERTEAUX, 2008). O padrão IEEE 802.11p define as camadas físicas e de controle de acesso ao meio (MAC) para redes veiculares. Além disso, a arquitetura WAVE designa uma família de padrões que não se restringe às camadas MAC e física, como apresentado na Figura 5. Os padrões da família IEEE 1609 definem outras camadas da pilha de protocolos, incluindo uma camada de rede alternativa à camada IP, características de segurança para aplicações DSRC e operação em múltiplos canais de comunicação (UZCATEGUI; ACOSTA-MARUM, 2009; ALVES et al., 2009).

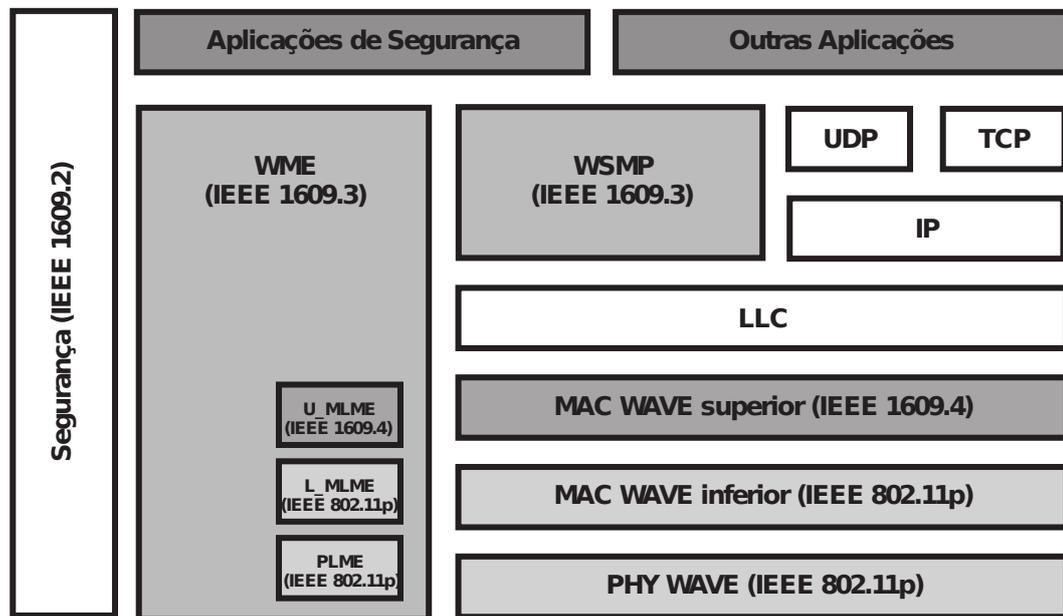


Figura 5: Arquitetura WAVE (ALVES et al., 2009).

O padrão IEEE 1609.1 especifica serviços e interfaces da aplicação de Gerenciamento de Recursos da arquitetura WAVE. Esse padrão sincroniza as OBUs e as RSUs, de uma rede *ad hoc* veicular, para se manterem conscientes do uso de recursos como memória e processamento. Esse gerenciamento de recursos, que o IEEE 1609.1 faz, visa um melhor escalonamento de tarefas e um melhor desempenho para VANETs.

O IEEE 1609.2 define formatos e processamento seguros de mensagens. Define também quando e como as mensagens de segurança devem ser processadas. A utilização de emissão e revogação de certificados de segurança e ferramentas de segurança tradicionais, como a Infraestrutura de Chaves Públicas (PKI) também são definidas por esse padrão. O IEEE 1609.2 também define um tipo especial de OBU: as PSOBUs - *Public Safety OBUs*. Essas OBUs de segurança pública são usadas por veículos especiais do governo como viaturas da polícia e do corpo de bombeiros.

O IEEE 1609.3 especifica os serviços das camadas de rede e de transporte, incluindo

o endereçamento e o roteamento, por definir que pilha usar na camada de enlace lógico (LLC - *Logical Link Control*). A LLC pode escolher usar o protocolo WSMP (*WAVE Short Message Protocol* - usado para dar mais eficiência) ou a pilha TCP/IP ou UDP/IP. Além disso, o padrão 1609.3 define a MIB (*Management Information Base*) para a pilha WAVE.

O padrão IEEE 1609.3 também é responsável pela configuração e manutenção do sistema através do plano de gerenciamento. Uma entidade definida por esse padrão (WME - *WAVE Management Entity*) é responsável por reunir as informações das entidades de gerenciamento de outras camadas como a MLME (*MAC Layer Management Entity*) e a PLME (*Physical Layer Management Entity*). A WME implementa os seguintes serviços: registro de aplicações, gerenciamento de WBSSs (*WAVE Basic Service Sets*), monitoramento da utilização de canais, manutenção da base de dados de gerenciamento etc.

Além de gerenciar as camadas de rede (IP - versão 6) e transporte (TCP e UDP), o IEEE 1609.3 também oferece uma alternativa à utilização dessas camadas por definir o WSMP (*WAVE Short Message Protocol*). A justificativa para utilizar esse novo protocolo é a maior eficiência no ambiente WAVE, onde se espera que a maioria das aplicações exija, dentre outros requisitos, latência muito baixa.

O padrão IEEE 1609.4 define modificações no padrão IEEE 802.11, para a operação em múltiplos canais. Para isso, esse padrão define como vai ser feita a comutação entre os canais. A classificação de pacotes indicando se eles vão ser comutados para o canal de controle ou para um dos canais de serviço e a prioridade de encaminhamento que esses pacotes terão são especificados pelo padrão IEEE 1609.4. A divisão de tempo entre os canais e a sincronização desses tempos em todos os dispositivos da rede também é feita pelo IEEE 1609.4 (ALVES et al., 2009).

Finalmente, o IEEE 802.11p define as diferenças específicas do controle de acesso ao meio em ambientes de comunicação WAVE com relação ao IEEE 802.11 tradicional. As principais diferenças são: alcances de operação de até 1000 metros, tipo especial de quadro, canais de 10 MHz descritos pelo DSRC, endereço MAC aleatório, controle de prioridade e potência de sinal (UZCATEGUI; ACOSTA-MARUM, 2009).

Uma outra diferença do IEEE 802.11p para o IEEE 802.11 tradicional diz respeito às BSSs (*Basic Service Sets*). BSS é um conjunto de estações se comunicando entre si através de um ponto de acesso (AP). O conjunto BSS serve para controlar o acesso a recursos e serviços do AP. Os usuários de redes IEEE 802.11 identificam um BSS pela SSID (*Service Set Identification*) divulgada periodicamente. No IEEE 802.11 tradicional, o processo de associação e autenticação de um usuário, que queira entrar numa BSS, demanda muito tempo. O IEEE 802.11p elimina esses processos, deixando que as camadas superiores cuidem dessas tarefas. O IEEE 802.11p cria um novo tipo de BSS, a WBSS (*WAVE BSS*). A WBSS utiliza SSIDs coringas. Dessa forma, um usuário pode receber e transmitir quadros de dados com o SSID

coringa, sem a necessidade de estar associado a um BSS (ALVES et al., 2009).

2.1.5 Aplicações

Agora podemos analisar as aplicações que utilizam as arquiteturas e os padrões descritos na seção anterior. A seguir, veremos as duas áreas de aplicações mais relevantes, que são as voltadas para a segurança do trânsito e as voltadas para o conforto dos passageiros dos veículos (YOUSEFI; MOUSAVI; FATHY, 2006).

2.1.5.1 Aplicações de Segurança

As aplicações de segurança visam aumentar a segurança dos passageiros por fazer com que mensagens de alerta ou emergência sejam transmitidas entre os veículos. Tais informações são ainda apresentadas aos motoristas para que ele tenha tempo suficiente para reagir a determinado evento. Essas informações também podem ser passadas para os sistemas de segurança ativos que podem, automaticamente, inflar um *air bag*, frear o veículo, transmitir outras mensagens de alerta etc. As aplicações de segurança visam também fornecer informações úteis sobre o ambiente físico, fazendo com que o motorista escolha um caminho mais seguro. Portanto, as aplicações de segurança pretendem minimizar os acidentes, oferecendo informações úteis aos motoristas e aos sistemas automáticos dos veículos, como avisos de colisões e alertas de condições de estradas.

Para auxiliar as aplicações de segurança, existem dois tipos de mensagens que usam o canal de controle (178) e possivelmente pode usar alguns canais de serviço: periódicas e orientadas a eventos. As periódicas são consideradas mensagens preventivas. Tais mensagens são também conhecidas como mensagens *beacon*. Essas mensagens fazem com que os veículos se tornem conscientes de seu ambiente. Elas contém informações sobre velocidade, posição e direção dos veículos e evitam situações de emergência antes que elas aconteçam.

Outro tipo de mensagem que usa o canal 178 e ajuda as aplicações de segurança são as mensagens orientadas a eventos. Essas mensagens são resultados de detecção de situações inseguras, tais como: uma batida de veículos, a proximidade de dois veículos em alta velocidade etc. Além disso, as mensagens orientadas a eventos têm a mais alta prioridade na fila de pacotes do canal 178 (YOUSEFI; MOUSAVI; FATHY, 2006).

Exemplos de aplicações dessa categoria são:

- Alerta de colisão - auxiliam os motoristas a evitar batidas entre veículos. Podem usar as informações de posição, direção e velocidade dos veículos e calcular que, se as trajetórias dos veículos estiverem para se cruzar, poderá haver uma colisão.

- Colisão iminente - geram informações para os mecanismos automáticos do veículo serem acionados (*air bags*, cintos de segurança, freios etc) para tentar diminuir os danos do acidente.
- Alertas sobre locais perigosos - através de sensores dos veículos, essas aplicações são capazes de detectar pistas escorregadias, buracos nas estradas e outras situações de risco e informá-las aos outros veículos para que eles evitem tais locais.
- Assistente de mudança de faixa e de ultrapassagem - através de informações cinemáticas dos outros veículos, tais aplicações permitem que o motorista saiba quando é possível trocar de faixa ou ultrapassar com segurança.
- Alerta de situação de semáforo - tais aplicações permitem que o motorista se mantenha consciente da situação do semáforo e, assim, possa ajustar sua velocidade.
- Veículo de emergência passando - essas aplicações são usadas quando veículos dos bombeiros, ou da polícia, ambulâncias ou outros, estão trafegando em situação de emergência e precisam avisar aos veículos que estão na frente para eles abrirem caminho.
- Informações restritas de segurança - são utilizadas quando veículos de emergência precisam trocar mensagens somente entre eles. Por exemplo, veículos da polícia informando a outros veículos da polícia sobre uma perseguição a um veículo em fuga.

2.1.5.2 Aplicações de Conforto

Esse tipo de aplicação visa melhorar o conforto dos passageiros e a eficiência do tráfego, levando os veículos a escolherem as rotas menos congestionadas. As mensagens geradas por esse tipo de aplicação utilizam os canais de serviço, não podendo usar o canal de controle (178) que é destinado às aplicações de segurança.

Exemplos de aplicações dessa categoria são:

- Aplicações de acesso à Internet - um veículo que quer se conectar à Internet envia uma requisição para encontrar um nó que tenha tal acesso (uma RSU, ou outros dispositivos de outras tecnologias como WiMAX ou 3G). O nó que tem tal acesso responde à requisição e, assim, começa a interação para o acesso à Internet.
- Diagnóstico - permitem que oficinas mecânicas, concessionárias de veículos, postos de combustível, avaliem a situação do veículo e sugiram serviços. Por exemplo, se o nível de combustível do veículo estiver baixo, um posto pode indicar um abastecimento já informando os preços.

- Jogos *multiplayers* entre os passageiros de alguns veículos - passageiros de diferentes veículos podem jogar determinados tipos de jogos que contém interatividade entre vários jogadores.
- Troca de músicas entre passageiros de veículos.
- Informações climáticas.
- Aplicações de pontos de interesse - permite obter informações sobre determinadas empresas ao longo do percurso, preços de restaurantes, bares, atrações turísticas etc.
- Informações sobre a rota menos congestionada até determinado destino.
- Pedágio automático - permite pagar o pedágio sem precisar parar o veículo ou sem precisar de qualquer intervenção do motorista ou dos passageiros.

2.1.6 Por que usar VANETs?

Diferentemente das redes 3G/4G e de qualquer rede que use estação-base, as VANETs podem ser formadas em qualquer lugar sem custo adicional. As redes 3G/4G têm um alto custo de instalação e manutenção, sendo necessário gastar centenas de milhões de dólares com infraestrutura (KARIM, 2008). Já VANETs podem ser formadas em estradas sem infraestrutura nenhuma, bastando que haja veículos habilitados para trocarem informações entre si.

As VANETs também têm um melhor desempenho se comparadas com qualquer rede com estação-base (KARIM, 2008). Algumas aplicações são sensíveis ao atraso. Por exemplo, um veículo freou e quer transmitir essa informação para o veículo que vem logo atrás para evitar que ele colida. Nas VANETs, essa informação pode ser passada diretamente de um veículo para o outro. Numa rede que usa estação-base, a informação sai do veículo que freou para a estação-base e da estação-base para o veículo que pode colidir. O atraso é maior. Além disso, a estação-base pode estar sobrecarregada de informações e pode aumentar o atraso mais ainda. Pode ser, também, que uma estação-base não esteja presente em todas as estradas, restringindo a área de cobertura.

Dessa forma, por ter um custo mais acessível, ter atrasos de roteamento menores e por poder trocar informações com outros nós mesmo em locais sem infraestrutura, uma VANET é mais apropriada para distribuir informações em cenários veiculares (KARIM, 2008).

2.1.7 Desafios

As características peculiares das VANETs criam alguns desafios que precisam ser solucionados para haver uma comunicação eficiente e para que os objetivos, inicialmente projetados, das redes veiculares sejam atingidos. Alguns desafios são listados a seguir:

- Escalabilidade - por exemplo, os protocolos de roteamento especificados para VANETs devem ser capazes de manter um bom desempenho independentemente da quantidade de nós da rede. Nas redes veiculares, essa quantidade pode variar consideravelmente.
- Interoperabilidade - os sistemas desenvolvidos para as VANETs devem ser capazes de interagir com dispositivos de outras tecnologias sem fio.
- Mudanças topológicas da rede - a velocidade dos nós contribui para que haja constantes mudanças na topologia da rede. Isso faz com que as conexões tenham pouco tempo de duração. Além disso, a dispersão rápida de veículos acarreta constantes fragmentações de caminhos e rotas.
- Densidade variável de nós - variações de velocidade, sinalizações de trânsito, modelos de vias automotivas e outros fatores contribuem significativamente para as mudanças de densidade em uma rede. Quando a densidade é alta, a taxa de perda de pacotes é maior devido aos problemas de contenção de meio e colisão de pacotes. Quando a densidade é baixa, há também consideráveis perdas de pacotes porque a probabilidade de conectividade é baixa (SHANKAR et al., 2008; PANICHPAPIBOON; PATTARA-ATIKOM, 2008; YOUSEFI; ALTMAN; EL-AZOUZI, 2007; TONGUZ et al., 2007).
- Inundação da rede por pacotes *broadcast* - muitas aplicações VANETs realizam comunicações utilizando *broadcast*. Mas, esse tipo de transmissão reduz a vazão útil da rede e aumenta as redundâncias de dados, contenções e colisões de pacotes. Além disso, deve-se tomar cuidado para que mensagens de segurança não sejam perdidas devido às inundações de pacotes por outras aplicações que usam *broadcast* (WISITPONGPHAN et al., 2007).
- Falta de um ponto de coordenação central - os projetistas de sistemas, aplicativos e protocolos para VANETs precisam desenvolver seus projetos de uma maneira que eles sejam distribuídos (HARTENSTEIN; LABERTEAUX, 2008).

2.2 Roteamento *Multicast*

O roteamento *multicast* é a transmissão de pacotes para um grupo de zero ou mais nós identificados por um único endereço de destino. Além disso, tal tipo de roteamento é o método mais eficiente de comunicação em grupo, superando o roteamento *unicast* e *broadcast*. Com o desenvolvimento de mais tecnologias em redes sem fio, os fornecedores de conteúdo e serviço estão cada vez mais interessados em formas de comunicação em grupo. Porém, existem várias dificuldades no processo de roteamento *multicast* em redes sem fio (VARSHNEY, 2002).

Mais especificamente em VANETs, o roteamento *multicast* torna-se ainda mais desafiador por causa das rápidas mudanças topológicas causadas pela mobilidade dos veículos

(MOHAPATRA; LI; GUI, 2005; MUÑOZ, 2009; SEBASTIAN et al., 2010). Além disso, a alta velocidade dos nós e a mudança de densidade dos ambientes veiculares causam grandes problemas aos protocolos *multicast*. Por isso é tão importante o estudo de um protocolo *multicast* adaptável ao mundo veicular.

2.2.1 Taxonomia dos Protocolos *Multicast*

Existem diversas classificações para o roteamento *multicast*. A mais usada é baseada na estrutura dos caminhos. Em geral, usa-se três tipos de estrutura: baseada em árvore, baseada em malha e híbrida (ver Figura 6). Os nós usados nessas estruturas formam um subgrafo que inclui o nó fonte e todos os nós receptores. Tais estruturas são fortemente dependentes do processo de criação de caminhos implementados pelo protocolo de roteamento específico (RUIZ; GOMEZ-SKARMETA, 2006).

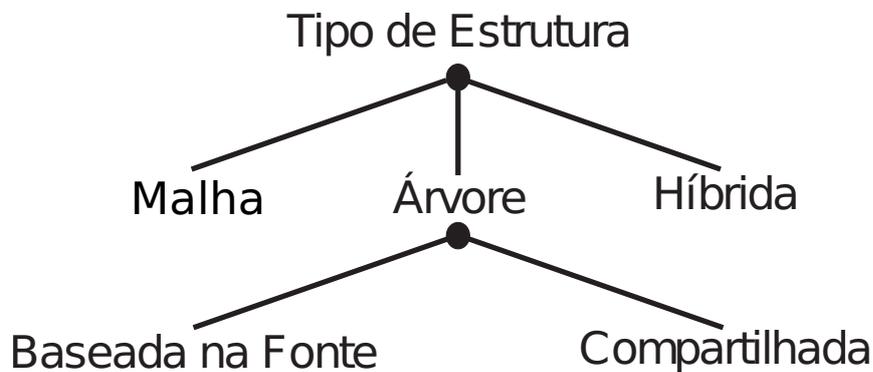


Figura 6: Taxonomia dos Protocolos *Multicast* em Redes *Ad Hoc* (BADARNEH; KADOCH, 2009)

A estrutura baseada em árvore pode ser dividida em duas: árvore baseada na fonte ou SBT (*Source-Based multicast Tree*) - uma árvore é formada para cada nó fonte (ou emissor de pacotes multicast); árvore compartilhada ou CBT (*Core-Based multicast Tree*), utiliza uma única árvore mesmo que exista mais de um nó fonte. A estrutura em árvore compartilhada tem menos *overhead* de controle, mas pode ter mais atraso de roteamento porque os caminhos não são otimizados. A estrutura em árvore baseada na fonte tem mais *overhead* de controle para manter mais árvores, mas tem menos atraso de roteamento porque os caminhos são otimizados de acordo com cada nó fonte. Entretanto, comparado com as outras estruturas, a baseada em árvore fornece objetividade no encaminhamento dos dados. Porém, é uma estrutura que não fornece robustez porque quando os nós se movimentam, os caminhos podem facilmente ser quebrados sem ter caminhos alternativos (JUNHAI; LIU; DANXIA, 2008; YANG; WU, 2005).

A estrutura baseada em malha sacrifica a eficiência do *multicast* por fornecer caminhos redundantes para um mesmo destino. Entretanto, a robustez aumenta porque se um nó se movimentar e quebrar um caminho, outros caminhos estarão disponíveis para se fazer o roteamento.

Já as estruturas híbridas tentam mesclar as estruturas em árvore e em malha para fornecer eficiência e robustez. Entretanto, pode-se criar árvores não otimizadas por causa da mobilidade dos nós (JUNHAI; LIU; DANXIA, 2008).

As Figuras 7 e 8 mostram duas estruturas *multicast*: baseada em árvore e baseada em malha. Em termos de definição, o nó S representa o nó fonte; os nós R1, R2, R3 e R4 representam os nós receptores; e os nós 1, 2, 3 e 4 com círculos duplicados representam os nós encaminhadores (não fazem parte do grupo *multicast*, mas fazem parte da estrutura *multicast* para manter a conectividade).

Nota-se que a estrutura baseada em malha tem caminhos redundantes (um nó receptor pode ter mais de um nó-pai - na Figura 8, o nó R3 tem como nó-pai os nós 3 e 4) para dar robustez ao roteamento. Por exemplo, se o nó 3 se mover e sair do alcance de transmissão do nó R3, haverá um caminho alternativo (S, 1, 2, 4, R3) para manter a conectividade do grupo *multicast*. Já na estrutura baseada em árvore, cada nó só tem um nó pai no caminho até o nó fonte (RUIZ; GOMEZ-SKARMETA, 2006).

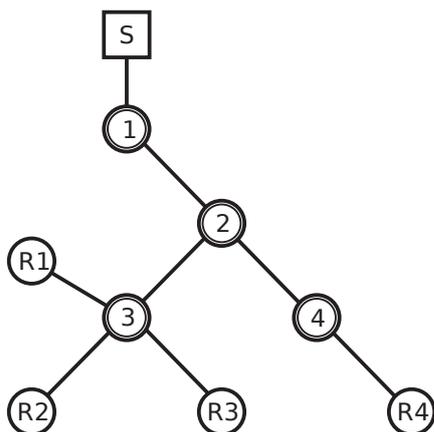


Figura 7: Estrutura *Multicast* em Árvore

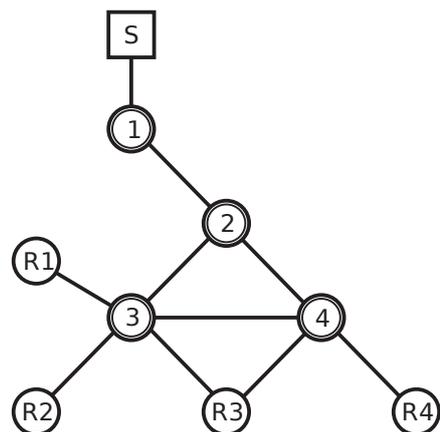


Figura 8: Estrutura *Multicast* em Malha

2.2.2 Protocolos de Roteamento *Multicast* em MANETs

Conforme explicado na Seção 2.2.3 e de acordo com nosso conhecimento da literatura, não existem protocolos *multicast* convencionais específicos para VANETs. Além disso, adaptar um protocolo específico de MANETs para VANETs torna a solução universal e interoperável, elementos fundamentais para os parceiros industriais que investem nas VANETs. Por isso, nesta Seção, analisamos alguns dos protocolos *multicast* convencionais para MANETs.

Na literatura, os principais tipos de protocolos de roteamento *multicast* são os baseados em árvore e em estruturas em malha (JUNHAI; LIU; DANXIA, 2008; YANG; WU, 2005). Alguns protocolos que exemplificam esses tipos de estruturas são explicados brevemente a seguir.

2.2.2.1 Multicast Ad hoc On-demand Distance Vector routing protocol (MAODV)

O protocolo MAODV (ROYER; PERKINS, 1999; YANG; WU, 2005) é uma extensão do protocolo AODV (PERKINS et al., 2003). Ele mantém uma árvore compartilhada para cada grupo *multicast*. Ele utiliza *broadcast* para encontrar uma rota, sob demanda, e construir uma árvore de roteamento. O primeiro nó de grupo *multicast* se torna o líder do grupo. O líder do grupo *multicast* é responsável por manter o número de sequência do grupo e enviar esse número para todos os membros. De acordo com a Figura 9, um nó que queira se juntar ao grupo *multicast* manda uma requisição de rota (RREQ) em *broadcast*. Essa mensagem será retransmitida em *broadcast* por todos os nós intermediários até encontrar um nó que esteja na árvore. Esse nó que está na árvore envia então uma resposta à requisição (RREP) em *unicast* pelo caminho reverso até o nó requisitante. O nó requisitante pode receber mais de uma mensagem RREP. Então, ele selecionará a rota mais curta (baseada na quantidade de saltos) e mais atual (baseada no número de sequência) enviando uma mensagem de ativação (MACT) em *unicast* através da rota escolhida. Todo o nó intermediário será um nó encaminhador.

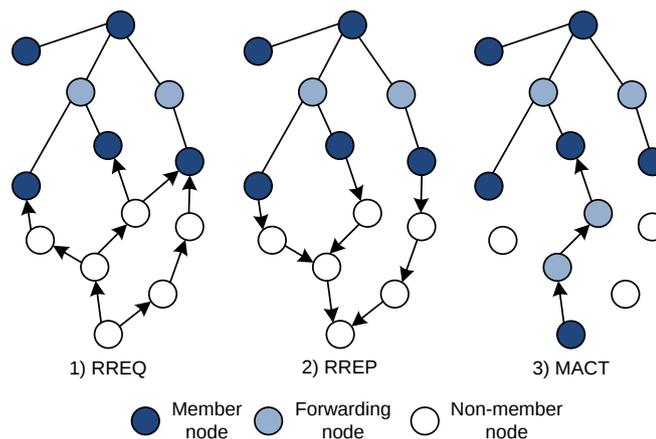


Figura 9: Processo de Junção *Multicast* do MAODV (MUÑOZ, 2009)

As tabelas de roteamento serão atualizadas para conter a informação de que existe mais um membro do grupo *multicast* e outros nós encaminhadores. Dessa forma, só existirá um caminho entre qualquer par de nós da árvore. Quando uma falha ocorre, um processo de descoberta de rota terá que ser feito novamente. Se um nó quiser sair do grupo *multicast*, ele pode sair se ele for um nó que não tem filhos. Se ele tiver filhos, poderá sair do grupo *multicast*, mas não da árvore *multicast* porque terá que encaminhar as mensagens para seus nós-filhos (BADARNEH; KADOCH, 2009).

2.2.2.2 Protocol for Unified Multicasting through Announcements (PUMA)

PUMA (VAISHAMPAYAN; GARCIA-LUNA-ACEVES, 2004; RANGARAJAN, 2011) é um protocolo de roteamento baseado em malha. Ele não precisa de protocolos *unicast* para

ajudá-lo no roteamento e os líderes do grupo *multicast* (*cores*) são eleitos dinamicamente. Só há uma mensagem de controle para todas as funções do protocolo: *multicast announcement* ou anúncio. Os nós receptores escolhem o *core* para servir de ponto de contato entre os nós que são membros do grupo *multicast* e os que são apenas membros da malha. Os receptores *multicast* se conectam ao *core* através dos caminhos mais curtos. Os nós que estão no caminho entre os receptores e o *core* coletivamente formam a estrutura de malha do protocolo.

As mensagens de anúncio fornecem detalhes como o número de sequência, ID (identificador) do grupo *multicast*, ID do *core*, distância para o *core* e detalhes sobre os pais dos nós. O pai de um nó é o vizinho preferido para o nó alcançar o *core*. O *core* do grupo transmite esses anúncios periodicamente a cada 3 segundos. Os outros nós quando recebem um anúncio, esperam um tempo e depois criam seu próprio anúncio. Como uma mensagem de anúncio atravessa a rede, ela estabelece uma lista de conectividade em cada nó na rede. Usando as listas de conectividade, os nós estão habilitados a estabelecer uma estrutura em malha e rotear pacotes de dados dos nós emissores para os nós receptores. Essa lista de conectividade guarda várias rotas para o *core* e as ordena segundo o número de sequência, distância para o *core* e o tempo de chegada do anúncio *multicast*. Quando há uma mudança de *core*, o nó apaga sua lista de conectividade antiga e constrói uma nova. Se uma rota ativa falhar, o nó usará as informações do anúncio seguinte, da lista de conectividade, para usar como rota.

A figura 10 apresenta uma estrutura em malha formada pelo PUMA e como os anúncios *multicast* são propagados. Já a figura 11 mostra a lista de conectividade no nó 6 do exemplo da figura 10.

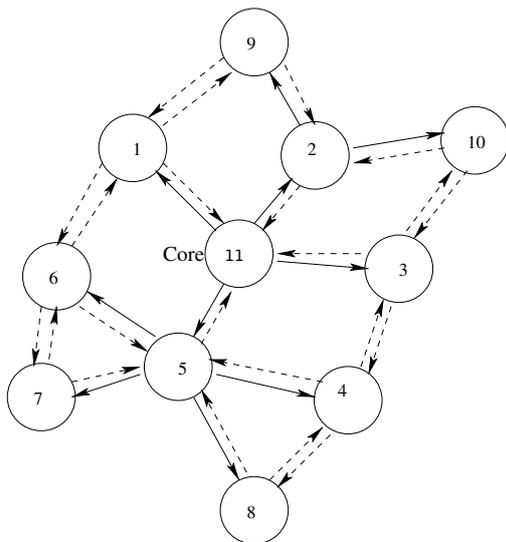


Figura 10: Estrutura em Malha do PUMA e Propagação de Anúncios *Multicast* (VAISHAMPAYAN; GARCIA-LUNA-ACEVES, 2004)

Lista de Conectividade no Nó 6
ID do Core = 11, ID do grupo = 224.0.0.1,
Núm. de Seq. = 79

Vizinho	Anúncio Multicast		Tempo (ms)
	Dist. para o Core	Pai	
5	1	11	12152
1	1	11	12180
7	2	5	12260

Figura 11: Lista de Conectividade do Nó 6 (VAISHAMPAYAN; GARCIA-LUNA-ACEVES, 2004)

Conforme as figuras 10 e 11, o nó 6 tem 3 entradas na sua lista de conectividade: o

anúncio vindo do nó 5, 1 e 7. Entretanto, ele escolhe a entrada do nó 5 como sua melhor entrada por causa da sua distância para o *core* e porque ele recebeu o anúncio do nó 5 antes do anúncio vindo do nó 1. O nó 6 usa essa entrada para gerar seu próprio anúncio *multicast*, que especifica o ID do *core* = 11, ID do grupo = 224.0.0.1, número de sequência = 79, distância para o *core* = 2 e o nó pai = 5.

Se um receptor quer se juntar a um grupo *multicast*, ele primeiro verifica se ele recebeu um anúncio para aquele grupo ou não. Se ele já tiver recebido, então o *core* especificado no anúncio também será o seu *core*. Se ele não tiver recebido o anúncio, ele passa a ser *core* e começa a distribuir anúncios para seus vizinhos. Se diversos receptores tentaram se juntar ao grupo ao mesmo tempo, e eles não tiverem recebido anúncio, o receptor que tiver o maior ID será eleito como o *core*.

No PUMA, os pacotes *multicast* se movem salto a salto até alcançarem um membro da malha. Um nó só encaminha um pacote *multicast* se ele for o pai do nó que transmitiu o pacote. Uma vez que o pacote de dados alcança a estrutura em malha, ele é inundado na malha (transmitido em *broadcast* para toda a malha). Para evitar redundâncias, o ID do pacote é usado para detectar e descartar pacotes duplicados.

2.2.2.3 On-Demand Multicast Routing Protocol (ODMRP)

O protocolo ODMRP (LEE; GERLA; CHIANG, 1999; YANG; WU, 2005) é baseado na estrutura em malha. Esse protocolo usa o conceito de nó encaminhador para fazer o roteamento *multicast*. Ele busca nós (membros ou não do grupo *multicast*) na rede para serem encaminhadores e só eles podem encaminhar mensagens *multicast*. O nó fonte, sob demanda, estabelece as rotas por mandar uma mensagem JOINDATA em *broadcast*. Essa mensagem é periodicamente transmitida para refrescar as informações do grupo *multicast* e de suas rotas. Enquanto a mensagem trafega pela rede, os nós vão adicionando informações às suas tabelas de roteamento. Essa mensagem será encaminhada até um membro do grupo *multicast* ser alcançado.

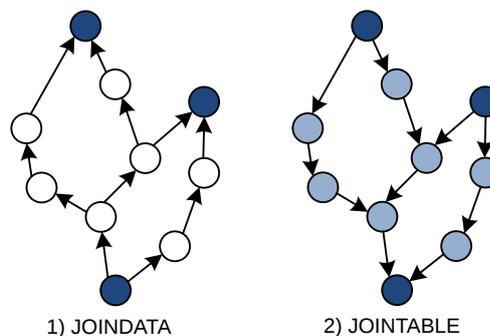


Figura 12: Criação de estrutura em malha do protocolo ODMRP (MUÑOZ, 2009)

Assim, o membro do grupo cria uma mensagem JOINTABLE e envia essa mensagem em *broadcast* para seus vizinhos. Todo nó vizinho saberá que está no caminho entre o nó fonte e um membro de grupo se o identificador do próximo salto em uma das entradas na mensagem JOINTABLE é seu próprio identificador. Então, esse nó estabelece ele mesmo como sendo um nó encaminhador. Depois, ele constrói sua própria mensagem JOINTABLE e propaga a mensagem até alcançar o nó fonte via caminho mais curto. Dessa forma, o nó fonte pode enviar dados para todos os membros do grupo com a ajuda da estrutura em malha. Além disso, os membros do grupo *multicast* podem deixar de ser membros a qualquer momento e a mudança pode ser detectada quando as mensagens para refrescar o grupo são enviadas (BADARNEH; KADOCH, 2009).

O processo de envio de JOINDATAs e JOINTABLEs para formar uma estrutura em malha para o protocolo ODMRP é descrito na figura 12. No momento 1, um nó fonte (círculo escuro mais abaixo) envia mensagens JOINDATA para encontrar receptores *multicast*. No momento 2, os nós receptores (círculos escuros mais acima) enviam mensagens JOINTABLE em *broadcast* para informar, ao nó fonte, as rotas disponíveis para o roteamento *multicast*. Dessa forma, uma estrutura em malha é formada com os nós encaminhadores (círculos levemente escuros) roteando as mensagens *multicast* entre o nó fonte e os nós receptores.

2.2.3 *Multicast X Geocast*

Em termos de definição, é necessário esclarecer o que é *multicast* e *geocast*. Para fazer *multicast*, de uma maneira geral, é necessário definir grupos *multicast*. Em algoritmos de *multicast* convencionais, um grupo *multicast* é considerado como uma coleção de nós que se registraram àquele grupo. Isso quer dizer, que se um nó quer receber mensagens de um grupo, ele tem que primeiro se juntar ao grupo. Quando um nó quer enviar dados, ele transmite em *multicast* com o endereço do grupo e todos os membros desse grupo recebem a mensagem. Os nós pertencentes ao grupo *multicast* não possuem qualquer limitação geográfica (KO; VAIDYA, 1999; HASSAN; AHMED; RAHMAN, 2011).

Já o termo *geocast* é uma variante do termo *multicast*. *Geocast* é a distribuição de mensagens para um conjunto de nós especificados em uma região geográfica determinada previamente. Diferentemente do *multicast* tradicional, aqui um grupo *multicast* (ou grupo *geocast*) é implicitamente definido pelo fato de um nó estar em determinado local. Se o nó estiver numa determinada área em qualquer momento, ele automaticamente se tornará um membro do grupo *geocast* correspondente em qualquer momento. Um nó que faz parte de um roteamento *geocast* não precisa solicitar sua entrada e nem requerer um endereço IP específico como no *multicast* tradicional (KO; VAIDYA, 1999; HASSAN; AHMED; RAHMAN, 2011).

Na próxima seção, apresentamos os principais protocolos *geocast* para contextualizar

o leitor e exemplificar que, embora um protocolo *geocast* seja um tipo específico de protocolo *multicast* convencional, tal como o MAODV, existe diferença entre esses tipos de protocolos. Além disso, serve como revisão de literatura, visto que não existem protocolos *multicast* convencionais projetados para VANETs.

2.2.4 Protocolos de Roteamento *Geocast* em VANETs

2.2.4.1 *ROBust VEhicular Routing (ROVER)*

O ROVER (KIHLE et al., 2007) é um protocolo de roteamento *geocast* confiável. Ele utiliza procedimentos semelhantes aos do protocolo AODV por utilizar mensagens de requisição e confirmação de rotas. Tais mensagens de controle são inundadas na rede e as mensagens de dados são enviadas via *unicast*. Cada mensagem precisa conter três informações: definição da mensagem (M), definição da aplicação (A) e definição da zona geográfica relevante (ZOR). O protocolo também define uma Zona de Encaminhamento (ZOF). Todos os veículos numa ZOF fazem parte do processo de roteamento.

O processo de montagem da árvore *multicast* começa quando a camada de roteamento recebe um pacote (M,A,ZOR) da camada de aplicação. Então o nó espalha uma mensagem ZRREQ (*Zone Route Request*), conforme a Figura 13, contendo um VIN (número de identificação do veículo), SS (número de sequência), localização e a ZOR atual. Os veículos que estão na ZOF não podem enviar uma ZRREP (*Zone Route Reply*), a não ser que eles já tenham recebido uma ZRREQ e precisem encaminhá-la. Mas, conforme a Figura 14, só quem pode gerar uma ZRREP é quem está na ZOR.

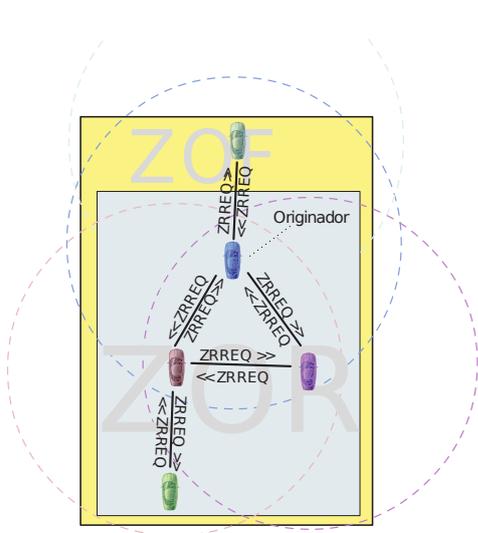


Figura 13: Envio de ZRREQ pelo protocolo ROVER (KIHLE et al., 2007)

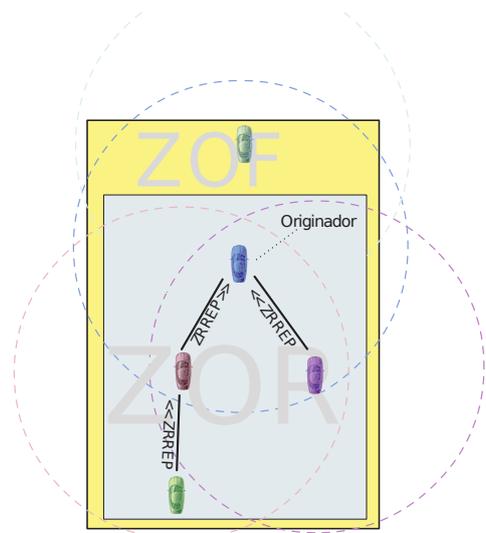


Figura 14: Envio de ZRREP pelo protocolo ROVER (KIHLE et al., 2007)

Diferentemente do AODV, uma ZRREP não é enviada de volta para o nó fonte, mas

somente para o nó de quem ele recebeu uma ZRREQ. Dessa forma, o nó fonte só envia uma mensagem de dados (via *unicast*) para os nós de quem ele recebeu uma ZRREP. Assim, a árvore *multicast* é formada usando várias ligações *unicast* entre os nós.

2.2.4.2 *Distributed Robust Geocast (DRG)*

O protocolo DRG (KIHLE; SICHITIU; JOSHI, 2008; JOSHI; SICHITIU; KIHLE, 2007) é um protocolo *geocast* que também usa os conceitos de ZOF e ZOR. Tal protocolo utiliza alguns nós como encaminhadores ou *relays* para transmitir uma mensagem para uma ZOR. A escolha desses nós encaminhadores (fazem parte da ZOF) é feita com base na distância para o destino, na taxa de cobertura (de acordo com o alcance de transmissão de cada nó) e do ângulo que se forma entre os nós. No percurso desses nós encaminhadores, acontecem retransmissões das mensagens até que um nó *relay* transmita a mensagem. Isso é tratado como um reconhecimento implícito. A Figura 15 mostra que o problema de fragmentação da rede é superado com a assistência dos veículos da ZOF, a saber, os veículos V_G V_F .

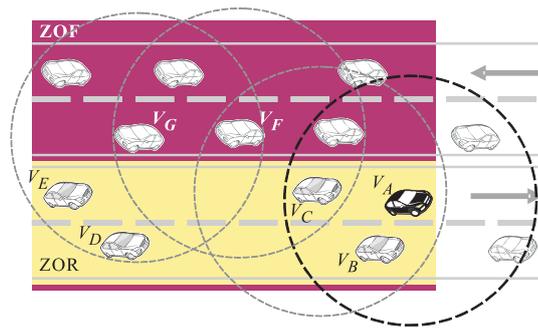


Figura 15: Protocolo DRG (LIN; CHEN; LEE, 2010)

Dessa forma, o DRG é um protocolo completamente distribuído, sem *overhead* de controle e sem informações de estado. Ele usa um flooding restrito, em que só alguns nós encaminham as mensagens para a ZOR. Por não ter mensagens de controle, a carga da rede é diminuída.

2.2.4.3 *Inter-Vehicles Geocast (IVG)*

O IVG (BACHIR; BENSLIMANE, 2003) é um protocolo *geocast* usado para informar todos os veículos numa estrada de qualquer perigo que aconteça, tal como um acidente. A área de risco é definida com base na direção e no posicionamento dos veículos. Veículos localizados na área de risco formam um grupo. O grupo é definido temporariamente e dinamicamente pela localização, velocidade e direção do veículo. O protocolo IVG usa *broadcasts* periódicos para superar possíveis fragmentações da rede e entregar a mensagem para todos os membros do grupo *multicast*.

O protocolo IVG também reduz a quantidade de saltos por fazer com que os nós mais distantes tenham prioridade no acesso ao meio e possam entregar a mensagem mais rapidamente. A Figura 16 mostra um exemplo de como funciona o protocolo IVG. O veículo V_A encontra um problema e informa isso para todos os veículos na área de risco. Os veículos V_B , V_C e V_D formam um grupo porque estão na área de risco. O responsável por encaminhar as mensagens de V_A é o V_C porque ele está mais longe do que o V_B . O veículo V_B recebe as mensagens de V_A , mas não as encaminha. Isso reduz a quantidade de saltos na transmissão de mensagens para a área de risco.

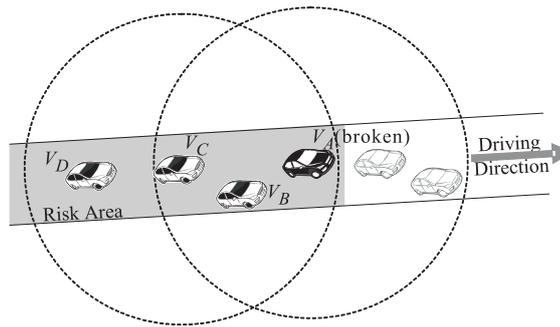


Figura 16: Protocolo IVG (LIN; CHEN; LEE, 2010)

2.2.4.4 Mobicast

O *Mobicast* (CHEN; LIN; LEE, 2010) é também um protocolo *geocast* que leva em conta o fator tempo e o espaço para estabelecer a Zona de Relevância. Esse protocolo transmite mensagens para uma zona geográfica no tempo t denotada por ZOR_t . Para manter a conectividade da ZOR_t , uma zona de encaminhamento de mensagens (ZOF_t) é criada para disseminar as mensagens para a ZOR_t .

Além disso, uma zona $Z_t^{V_i}$ é criada para estimar mais ainda uma ZOF_t . Uma $Z_t^{V_i}$ é uma zona elíptica para encaminhar mensagens do protocolo *Mobicast* mais próxima do veículo-destino. Se um veículo não pode enviar mensagens para qualquer vizinho, uma nova $Z_t^{V_i}$ precisa ser criada para manter a conectividade. Assim, uma ZOF_t é gerada por todas as $Z_t^{V_i}$ iniciadas. A Figura 17 ilustra isso. No tempo t , os veículos V_1 , V_2 , V_3 e V_4 estão localizados na ZOR_t e recebem as mensagens do veículo V_e . No tempo $t + 1$, V_2 e V_4 não podem receber mensagens diretamente devido a fragmentação da rede. No tempo $t + 1$, V_e , V_5 e V_1 iniciam $Z_{t+1}^{V_e}$, $Z_{t+1}^{V_5}$ e $Z_{t+1}^{V_1}$ para encaminhar mensagens para V_4 e V_2 . Nesse caso, $ZOF_{t+1} = ZOR_{t+1} \cup Z_{t+1}^{V_e} \cup Z_{t+1}^{V_5} \cup Z_{t+1}^{V_1}$.

2.3 Otimização por Colônia de Formigas

A otimização por colônia de formigas (ACO - *Ant Colony Optimization*) surgiu na década de 90 como uma nova técnica bioinspirada para resolver problemas difíceis de otimi-

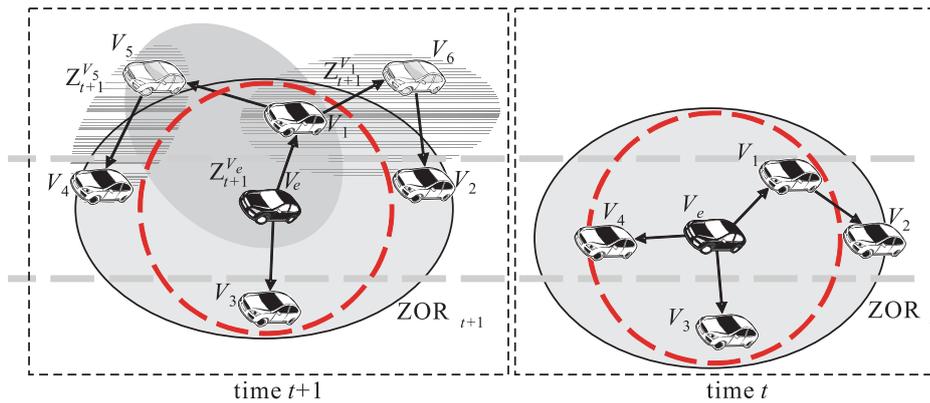


Figura 17: Protocolo *Mobicast* (LIN; CHEN; LEE, 2010)

zação combinatória (DORIGO, 1992; DORIGO; MANIEZZO; COLORNI, 1996; DORIGO et al., 1991). ACO é uma metaheurística ¹ usada para encontrar boas soluções em problemas computacionais num tempo razoavelmente viável (GLOVER; KOCHENBERGER, 2003).

2.3.1 Inspiração Biológica

A fonte de inspiração da metaheurística ACO foi o comportamento forrageiro das formigas reais. Inicialmente, quando as formigas saem de suas tocas, elas procuram por comida de maneira aleatória. Assim que uma formiga acha uma fonte de comida, ela avalia quão boa é essa fonte e volta para a toca carregando um pouco dessa comida. No caminho de volta para a toca, as formigas deixam um rastro de feromônio - substância química volátil (evapora com o tempo) capaz de modificar o comportamento e a reação de outros membros da mesma espécie que estejam em um local próximo ao ocupado pelo membro da espécie que excretou a substância (KARLSON; LÜSCHER, 1959). A quantidade de feromônio depositado depende da qualidade e da quantidade da fonte de comida encontrada pela formiga. Essa quantidade de feromônio vai atrair outras formigas para tal fonte de alimento. Isso não quer dizer que todas as formigas seguirão pelo caminho que tem mais feromônio. Mas, quer dizer que haverá uma maior probabilidade de o movimento aleatório das formigas seguir por esse caminho. Quando as formigas seguem por esse caminho, elas vão depositar ainda mais feromônio e o nível dessa substância vai aumentar. Essa comunicação indireta entre as formigas através de feromônios fazem com que elas encontrem o caminho mais curto entre o formigueiro e a fonte de alimento (TIRUCHENGODE, 2009). A metaheurística ACO utiliza esses princípios de formigas reais para modelar as formigas artificiais.

A Figura 18 mostra esses princípios. Em (i), observa-se uma formiga indo em busca

¹O termo metaheurística se refere a um conjunto de conceitos algorítmicos que podem ser aplicados a uma diversidade de problemas computacionais (BLUM; ROLI, 2003). Uma metaheurística é uma espécie de framework algorítmico aplicável à uma ampla gama de problemas de otimização e que requer relativamente poucas modificações para torná-la adaptada a um problema específico.

de alimento no momento 'a' e voltando para o formigueiro deixando feromônio por onde passa no momento 'b'. (ii) mostra várias formigas indo em busca de comida e voltando para o formigueiro depositando mais feromônio. As formigas que escolhem o menor caminho retornam primeiro ao formigueiro. Dessa forma, o menor caminho vai se tornando o que tem maior quantidade de feromônio. Isso faz com que a probabilidade de outras formigas escolherem esse caminho aumente. Em (iii), percebemos a convergência de caminhos. Após algum tempo, a maioria das formigas passa a utilizar o caminho mais curto.

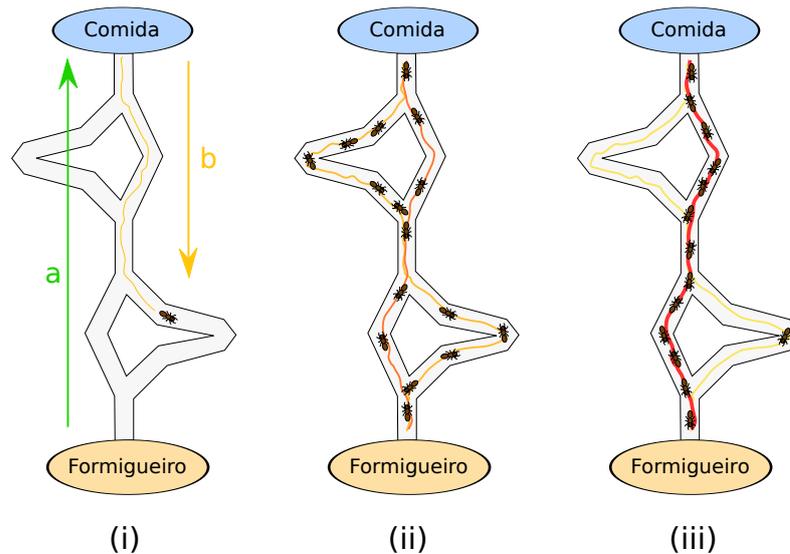


Figura 18: Colônia de Formigas

2.3.2 A Metaheurística Otimização por Colônia de Formigas

A metaheurística ACO é baseada em inteligência coletiva por usar informações de agentes (formigas) e possui as seguintes características (COELHO; NETO, 2004):

- É um algoritmo não determinístico que usa mecanismos presentes na natureza. Utiliza o comportamento das formigas para a escolha de caminhos eficientes até uma fonte de comida.
- É um algoritmo paralelo e adaptativo. Os agentes se movem de maneira independente, simultânea e sem um controle central.
- É um algoritmo cooperativo. Cada agente (formiga) escolhe um caminho com base na informação depositada por outros agentes (rastros de feromônios) que tenham percorrido um mesmo caminho.

Para tentar provar as propriedades teóricas da metaheurística ACO, é interessante entender sua generalização (DORIGO; BLUM, 2005). O algoritmo 1 apresenta uma generalização

simplista e resumida do pseudo-código da metaheurística ACO. Como ele é um modelo geral, ele requer algumas modificações para se adaptar a um problema específico. Mas, um ingrediente básico da metaheurística ACO é o procedimento de construção da solução (MANGAI; TAMILARASI; VENKATESH, 2008). A cada iteração, n formigas probabilisticamente constroem soluções para um dado problema de otimização combinatória. Tal procedimento monta soluções como sequências de elementos do conjunto finito de possibilidades de solução. Esse processo de construção de soluções pode ser visto também como um caminhamento no grafo $G = (V, A)$, cujos vértices são elementos do conjunto V e as arestas são elementos do conjunto A . Dessa forma, à medida que as formigas vão percorrendo o grafo G , uma solução vai sendo montada.

Algoritmo 1: Otimização por Colônia de Formigas (MANGAI; TAMILARASI; VENKATESH, 2008)

Entrada: Uma instância X de um problema de otimização combinatória

início

enquanto condições de saída não forem satisfeitas **faça**

Construa_Solução();

Atualize_Feromônio();

Melhor_Solução \leftarrow Melhor_solução_na_população_de_soluções;

fim

fim

Saída: Candidato a melhor solução da instância X

Para uma formiga ir caminhando no grafo, ela tem que ir escolhendo os próximos caminhos. Conforme Huang, Han e Hou (2007), essas escolhas são feitas com base na probabilidade de transição mostrada abaixo:

$$p(c_i^j | s_{parcial}) = \frac{[\tau_i^j]^\alpha * [\eta(c_i^j)]^\beta}{\sum_{c_k^l \in R(s_{parcial})} [\tau_i^l]^\alpha * [\eta(c_k^l)]^\beta}, \forall c_i^j \in R(s_{parcial}) \quad (2.1)$$

Nessa fórmula (2.1), $s_{parcial}$ é a solução que está sendo construída, c_i^j representa a escolha de um componente de $s_{parcial}$, R é o conjunto de elementos para se construir o próximo componente da solução parcial, $[\tau_i^j]$ é o nível de feromônio armazenado entre i e j , $[\eta(c_i^j)]$ é um valor heurístico associado ao componente c_i^j e α e β ($0 < \alpha, \beta < 1$) são valores de ponderação para medir a importância de τ e η .

Entretanto, existem diversas formas de se representar as probabilidades de transição. A ilustrada em 2.1 é vista como tradicional por ter sido utilizada nos primeiros algoritmos da metaheurística ACO (DORIGO; MANIEZZO; COLORNI, 1996; DORIGO et al., 1991).

Além disso, existe a função de atualização de feromônio. Essa função tem como objetivo aumentar o nível de feromônio das melhores soluções. Segundo Dorigo e Blum (2005),

grande parte dos algoritmos de ACO utilizam uma variação da regra de atualização de feromônio mostrada abaixo:

$$\tau_i^j \leftarrow (1 - \rho) * \tau_i^j + \rho * \sum_{s \in S} F(s) \quad (2.2)$$

Em 2.2, temos que τ_i^j é o nível de feromônio que está sendo atualizado, $\rho \in (0, 1]$ é a taxa de evaporação. A taxa de evaporação tem a função de decrementar todos os valores de feromônio de maneira uniforme. A evaporação serve para evitar uma convergência muito rápida do algoritmo em direção a uma região subótima, isto é, serve para não convergir para um ótimo local. Dessa maneira, essa taxa de evaporação serve para que o algoritmo explore novas regiões do espaço de busca. O s representa uma solução no conjunto S de soluções. Já $F(s)$ é a função que avalia a qualidade de uma solução.

Dessa forma, no fim do algoritmo, a melhor solução encontrada tem maior probabilidade de ser escolhida pelas próximas formigas que irão percorrer o grafo G .

2.3.3 Por que usar Otimização por Colônia de Formigas?

A metaheurística Otimização por Colônia de Formigas possui vantagens em ser aplicada aos problemas de roteamento em Redes *Ad Hoc* Móveis (GUNES; SORGES; BOUAZIZI, 2002; GLOVER; KOCHENBERGER, 2003; TIRUCHENGODE, 2009). Tais vantagens são consideradas a seguir:

- Adaptação a topologias dinâmicas. Uma topologia dinâmica é um grande problema para a maioria dos protocolos de roteamento em MANETs. A metaheurística ACO é baseada em sistemas de agentes e funciona com formigas individuais. Isso permite uma alta adaptação à topologia atual da rede.
- Avaliação da qualidade dos enlaces. É possível levar em conta as informações de qualidade do enlace ou da conexão ao se depositar ou evaporar as concentrações de feromônios. Isso melhora o processo de decisão do protocolo de roteamento ao escolher boas rotas e descartar rotas ruins. Para isso, é necessário modelar uma forma de monitorar a qualidade dos enlaces.
- Suporte para múltiplos caminhos. Cada tabela de roteamento de um nó pode armazenar a quantidade de feromônio. Assim, quando uma formiga vai escolher seu próximo salto ou caminho, ela pode checar o nível de concentração de feromônio do nó atual. Portanto, há suporte para múltiplos caminhos.
- Tempo. A metaheurística ACO é usada para encontrar boas soluções (não necessariamente a solução ótima) em tempo computacionalmente viável.

- Aprendizado contínuo e cooperativismo. As soluções vão sendo melhoradas ao passo que as formigas continuam explorando novas regiões do espaço de busca. As formigas continuam aprendendo sobre a qualidade de determinados caminhos com base nos feromônios deixados por outras formigas.
- Não há um controle central. Em VANETs, essa característica é muito importante visto que também não existe um nó central que controle todas as informações da rede. Um algoritmo distribuído como os da metaheurística ACO são de muita valia em VANETs.

3 TRABALHOS RELACIONADOS

3.1 Trabalhos que Propuseram Aprimoramentos ao MAODV

Em Chang, Lin e Liang (2008), foi feito um aprimoramento no protocolo *multicast* MAODV. Em redes *ad hoc* móveis, enlaces sem fio podem ser desfeitos ocasionalmente devido à mobilidade dos nós, fazendo com que um nó na ponta de um enlace saia do alcance do nó que está na outra ponta do enlace. Esse problema em árvores *multicast* ocasiona falha de ramos inteiros de uma árvore, baixa confiabilidade de rede e baixa taxa de entrega de pacotes. Para superar esse problema, o protocolo MAODV transmite uma mensagem RREQ em *broadcast* para encontrar uma nova rota e reparar aquele enlace com erro. Mas, isso resulta em mensagens de controle excessivas e baixo sucesso na taxa de reparo do enlace. No mecanismo proposto neste trabalho, armazena-se uma lista de vizinhança em cada nó. Dessa forma, quando há uma falha de enlace, um nó já pode solicitar uma nova rota para seus vizinhos com roteamento *unicast* e não *broadcast* do MAODV. Para tanto, novas mensagens tiveram que ser criadas para fazer o esquema de reparo local baseado em *unicast* funcionar. Na lista de vizinhança, podem ficar um número razoável de enlaces ou caminhos para recuperar o enlace desconectado. Quando um enlace falha, procura-se o caminho mais próximo (distância física) ou caminho que tenha o menor número de saltos. Isso aumenta a confiabilidade da rede, minimiza o número de mensagens de controle e reduz o atraso de reparo dos enlaces desfeitos. As simulações foram feitas no NS-2 e observou-se que o esquema de reparo local, usando *unicast*, foi, de fato, melhor do que o MAODV na taxa de reparo e no *overhead* de controle.

Em Yan, Wan e Fan (2010), os esforços são concentrados em resolver o problema das frequentes mudanças de topologia em redes *ad hoc* móveis (MANETs). Se o problema da estabilidade de topologia não for considerado, as falhas de enlaces serão inevitáveis e, assim, árvores *multicast* terão que ser reconstruídas continuamente. Isso acarreta maior *overhead* de controle porque mais mensagens de protocolo terão que ser enviadas para que a árvore seja reconstruída. Dessa forma, para melhorar a estabilidade do roteamento, um novo protocolo foi proposto e chamado de PPC-MAODV. Tal protocolo leva em conta a estabilidade dos enlaces que pertencem a um caminho para se construir uma árvore *multicast*. As simulações mostram que o PPC-MAODV pode reduzir o número de pacotes descartados e o atraso fim-a-fim.

Em Zhu e Lian (2008), a preocupação maior é com a predição de mobilidade porque em redes *ad hoc* móveis os nós se movem aleatoriamente e a topologia da rede muda rapidamente. Nesse trabalho, foi proposto um protocolo chamado de MPMRPQ (*QoS Multicast Routing Protocol with Mobile Prediction based on MAODV*) que é baseado no protocolo MAODV e utiliza predição de mobilidade para melhorar o roteamento *multicast* mesmo com topologias rapidamente mutáveis. Ainda leva-se em conta parâmetros de qualidade de serviço (QoS - Qua-

lity of Service) em decisões de roteamento. No momento da construção da árvore *multicast*, os seguintes parâmetros dos enlaces são avaliados: atraso, variação do atraso, taxa de perdas de pacotes, largura de banda e tempo de vida do enlace. O tempo de vida do enlace está de acordo com informações de mobilidade. Se o tempo de vida de um enlace, estiver abaixo do permitido, o enlace já pode ser considerado como desconectado e será necessário um re-roteamento do caminho. O MPMRPQ é testado no NS-2, com um modelo de mobilidade aleatório e comparado ao MAODV.

Em Jie et al. (2009), o problema de ter que reconstruir uma árvore *multicast* frequentemente devido à mobilidade dos nós em MANETs, fazendo com que o custo de roteamento e o atraso de transmissão aumentem consideravelmente, é tratado. Um método de escolha de caminho para a árvore foi desenvolvido de acordo com a taxa de mudança de nós vizinhos para se evitar que árvore *multicast* permaneça mais tempo sem a necessidade de se reconstruir. Baseado nesse método, um novo protocolo chamado de SP-MAODV (*Stable Path based MAODV*) foi desenvolvido para escolher caminhos mais estáveis e com menos saltos. Além disso, com o método proposto, não é necessário enviar mensagens *Hello* do protocolo MAODV periodicamente. Os resultados da simulação mostram que a taxa de transmissão de pacotes, o *overhead* de roteamento, o atraso fim-a-fim médio e a variação do atraso mostraram-se melhor no SP-MAODV. Dessa forma, o SP-MAODV reduziu a probabilidade de interrupção de caminhos e melhorou o desempenho do MAODV.

Em Vaishampayan (2004), mudanças foram propostas ao MAODV e o protocolo resultante foi chamado de ROMANT (*RObust Multicasting in Ad hoc Networks using Trees*). Uma mudança proposta no ROMANT foi quanto ao mecanismo de *Merge*. No MAODV, um nó que quisesse mudar de árvore teria que pedir permissão ao seu respectivo líder de grupo. Nesse trabalho, a proposta é não precisar mais dessa permissão. Se um nó receber duas mensagens GRPH (*GRouP Hello*) com o mesmo endereço IP *multicast*, ele simplesmente descarta a GRPH do líder de grupo que tenha o menor endereço IP. Se o líder de grupo que tem o menor endereço IP receber uma GRPH com o mesmo endereço IP *multicast*, ele para de enviar mensagens GRPH. Além disso, uma nova mensagem chamada *Join Annoucement* é proposta. Essa mensagem contém o endereço do melhor próximo salto até o líder do grupo *multicast*. Existe a possibilidade de usar vários próximos saltos porque um nó pode receber várias GRPHs de diferentes nós. Assim, uma lista de conectividade é mantida com todas as informações das diferentes GRPHs. Assim, depois que o próximo salto recebe uma *Join Annoucement*, ele próprio se considera parte da árvore e inicia o envio de *Join Annoucements* periódicos até o líder do grupo. Em contraste com o MAODV, onde todos os nós periodicamente transmitem mensagens *Hello*, somente membros da árvore transmitem *Join Annoucements* no protocolo ROMANT. Além disso, no ROMANT, como sempre os membros da árvore ficam escolhendo os melhores próximos saltos através das *Join Annoucements* e das listas de conectividade, muitos enlaces são reparados antes de falharem. Isso ainda evita o *overhead* de mensagens de controle do

MAODV, tais como um novo processo de RREQ-RREP-MACT. Finalmente, o ROMANT foi testado no simulador Qualnet e comparado com os protocolos MAODV e ODMRP. A partir dos resultados, observa-se que o ROMANT supera as fragilidades do MAODV e mantém um desempenho melhor ou igual ao do ODMRP.

Em Chen, Zhong e Liu (2009), um novo protocolo foi proposto usando como base o protocolo MAODV. Tal protocolo é chamado de MMPS (*Multicast protocol based on Mobility Prediction and Self pruning flooding*). Duas desvantagens do protocolo MAODV são tratadas: transmissão de RREQ e GRPH em inundação cega e reparo passivo de falha de enlaces. O método de inundação cega do MAODV faz com que as mensagens RREQ e GRPH sejam espalhadas por toda a rede. Isso consome recursos de rede (largura de banda, por exemplo), pode causar congestionamento na rede, aumentar a probabilidade de colisão de pacotes e as perdas de pacotes. Para sanar esse problema, o MMPS usa uma inundação auto-ajustável. Esse tipo de inundação faz com que os nós verifiquem uma lista de nós adjacentes e calculem se é necessário encaminhar uma mensagem *Hello* ou uma GRPH. Dessa maneira, só alguns nós irão repassar tais mensagens e os recursos da rede serão poupados. Uma outra desvantagem identificada do protocolo MAODV é no processo de reparo de falha de enlaces. Esse processo é passivo, ou seja, o processo só será iniciado quando o enlace já falhou. Isso resulta em perdas de pacotes para o nó membro do grupo *multicast* que está desconectado da árvore *multicast*. Além disso, o processo de recuperação pode ser lento. O MMPS mantém uma tabela de estado de enlace em cada nó. A cada pacote recebido, a potência do sinal é calculada. Se a potência estiver abaixo de um limite estabelecido, significa que o enlace irá falhar em breve. Assim, o processo de recuperação do enlace é ativado antes de o enlace falhar por completo. Isso evita mais perdas de pacotes e mais *overhead* de controle. Usando o *Network Simulator*, o MMPS foi avaliado e comparado ao MAODV. Percebeu-se que o MMPS teve melhor desempenho na taxa de entrega de pacotes e no *overhead* de controle do que o MAODV.

3.2 Utilização de Otimização por Colônia de Formigas em Problemas de Roteamento em Redes *Ad Hoc* Móveis

Em Correia, Celestino e Cherkaoui (2011), um protocolo reativo (MAR-DYMO) aplicado somente ao roteamento *unicast* foi proposto para aprimorar o protocolo DYMO (CHAKERES; PERKINS, 2010; SOMMER; DRESSLER, 2007). O MAR-DYMO usou os procedimentos de roteamento do DYMO para modelar o comportamento de formigas que ajudarão na escolha de melhores rotas e na prevenção de falhas. Essa escolha é feita com base em informações de posição e velocidade que são utilizadas pelas formigas para a evaporação e o depósito de feromônios. Esses feromônios indicam quão atrativa é uma rota. A atratividade de uma rota é calculada de acordo com os seguintes parâmetros: distância entre os nós e tempo de vida do caminho. Já a prevenção de falhas é feita pela evaporação do feromônio, ou seja, quando

um enlace vai se tornando defeituoso, o nível de feromônio diminui mais rapidamente. Dessa forma, um enlace defeituoso terá uma probabilidade menor de ser escolhido. Além disso, as informações de mobilidade são usadas para deixar as mensagens *Hello* aperiódicas. As mensagens *Hello* só serão transmitidas quando for necessário. Dessa forma, o MAR-DYMO tenta ser um protocolo adaptável aos cenários veiculares. Simulações foram feitas em cenários urbanos e o MAR-DYMO foi comparado com outros protocolos.

Em Shen et al. (2008), a metaheurística colônia de formigas foi utilizada para o problema de roteamento *multicast* em redes *ad hoc*. Nessas redes, a energia para um nó se manter ativo na rede é uma preocupação. Os protocolos devem ser desenvolvidos de um modo a poupar esse recurso. Algumas propostas tentam diminuir a potência de transmissão dos nós intermediários para poupar a energia deles e assim aumentar seus tempos de vida. Entretanto, ao se fazer isso, a quantidade de saltos, para se chegar nos destinos, pode aumentar. Se a quantidade de saltos aumentar, o atraso fim-a-fim dos pacotes de dados também irá aumentar e prejudicar aplicações sensíveis ao atraso. Dessa forma, o objetivo dessa proposta é diminuir as perdas de energia dos nós *ad hoc* e respeitar os limites de atraso estabelecidos. Para isso, uma árvore de Steiner (HWANG; RICHARDS, 1992) restrita é construída com o intuito de diminuir as potências de transmissão dos nós e limitar a quantidade de saltos dos caminhos *multicast*. Depois que a árvore é construída, formigas são enviadas pela rede para procurar rotas que diminuam o custo de encaminhamento (relação energia-atraso ou potência-quantidade de saltos). As formigas depositam uma maior quantidade de feromônio nas rotas de menor custo e o feromônio vai evaporando nas rotas de maiores custos. Assim, alguns caminhos podem ser trocados na árvore, de acordo com o grau de feromônio de cada rota e a árvore de Steiner restrita vai mantendo a melhor relação entre o consumo de energia e o atraso fim-a-fim.

Em Tiruchengode (2009), a metaheurística colônia de formigas é utilizada para aperfeiçoar o roteamento *multicast* em redes MANETs. Nesse trabalho, foi proposto o algoritmo AMR (*Ant based Multicast Routing*) para otimizar diversas métricas de roteamento *multicast* simultaneamente. Uma árvore *multicast* é construída usando formigas, que avaliam as rotas depositando feromônio de acordo com a atratividade de cada rota. O problema que o AMR resolve é encontrar uma árvore *multicast* que minimize o custo da árvore conforme os níveis de atraso fim-a-fim e conforme a altura máxima da árvore. O AMR foi comparado ao MAODV com mobilidade aleatória no simulador NS-2 e mostrou ter um desempenho melhor do que o do MAODV.

Em Huang, Han e Hou (2007), um algoritmo de roteamento *multicast* chamado AS-DLMA (*Ant System for Delay-constrained Low-cost Multicast routing Algorithm*) foi proposto para redes infraestruturadas. O algoritmo começa buscando uma rota de um nó fonte para todos os nós receptores usando o algoritmo de caminho mais curto de Dijkstra. Então, a árvore *multicast* é transformada num grafo para as formigas percorrerem e otimizarem parâmetros de

qualidade de serviço. O principal objetivo do ASDLMA é encontrar um árvore que minimize o atraso fim-a-fim. Para tanto, as formigas depositam mais feromônio nas rotas que tiverem menores atrasos. Com o tempo, tais rotas serão escolhidas para ficarem na árvore. Finalmente, o ASDLMA foi avaliado e comparado com outros protocolos e obteve resultados interessantes.

Em Mangai, Tamilarasi e Venkatesh (2008), foi proposto um protocolo de roteamento *multicast* baseado em otimização por colônia de formigas para redes *ad hoc* sem fio. Esse protocolo é um aprimoramento do DCMP (*Dynamic Core based Multicast routing Protocol*), um protocolo *multicast* baseado em estrutura em malha. A proposta desse trabalho é utilizar sistemas de colônias de formigas para otimizar a estrutura em malha por, frequentemente, buscar rotas mais curtas (que terá mais quantidade de feromônio) e assim adaptar a estrutura em malha com essas novas rotas. Os experimentos foram realizados no simulador NS-2 comparando o DCMP com e sem colônia de formigas. Observou-se que o DCMP com colônia de formigas teve uma maior taxa de entrega de pacotes e menor *overhead* de controle.

3.3 Outros Trabalhos que Contribuíram para a Proposta

Outros trabalhos, de temas variados, também foram importantes para a concepção do MAV-AODV. A seguir, listamos alguns desses trabalhos.

Em Biradar, Manvi e Reddy (2010), foi desenvolvido um melhoramento no roteamento *multicast* baseado em estrutura em malha para redes *ad hoc* móveis. Esse melhoramento foi proposto para deixar o roteamento *multicast* mais estável e confiável. Uma estrutura em malha é construída por usar RREQs (*Route REQuests*) e RREPs (*Route REPlies*). Os RREPs fazem uso de um banco de dados que fica em cada nó da rede e que contém informações sobre a estabilidade dos enlaces. Os caminhos estáveis são baseados na escolha de nós encaminhadores que tenham alta estabilidade na conectividade de seus enlaces. A estabilidade de enlace é calculada de acordo com parâmetros dos nós, tais como a potência de recepção, distância entre os nós vizinhos e um histórico de BER (*Bit Error Rate*). Esse histórico de BER não precisa ser grande. Para se obter uma medida precisa, basta utilizar intervalos de confiança sobre os históricos. Dessa maneira, o protocolo desenvolvido e chamado de LSMRM (*Link Stability based Multicast Routing scheme in Manet*) foi comparado com outros protocolos de roteamento *multicast* baseado em estrutura em malha como o ODMRP e o EODMRP. O cenário de mobilidade MANET foi o RWP (*Random WayPoint*). Observou-se que o esquema proposto produziu melhores taxa de entrega de pacotes, reduziu o atraso de pacotes e reduziu vários tipos de *overhead*. De fato, esse trabalho conseguiu deixar o roteamento *multicast* baseado em estrutura em malha mais estável e confiável.

O artigo em Sebastian et al. (2010) apresenta um esquema de roteamento *multicast* que é usado para transmissão de mensagens de emergência para evitar acidentes de trânsito.

Esse esquema de roteamento tem um mecanismo que detecta quais veículos estão ameaçados e mantém uma árvore *multicast* para tais veículos. Desse modo, quando ocorrer qualquer anomalia, os veículos serão avisados por mensagens *multicast*. É mencionado que os recursos da rede serão poupados porque nem todos os veículos retransmitirão a mensagem e haverá menos redundância de pacotes. Para modelar a identificação e escolha dos nós *multicast* utiliza-se o problema da árvore de Steiner com o atraso e o alcance de transmissão sendo os parâmetros de otimização. A Figura 19 ilustra o conceito desse esquema de roteamento. O veículo A forma um grupo *multicast* com os veículos G, B e C para transmitir mensagens de emergência. O grupo *multicast* é formado com base no grau de ameaça e influência que um nó exerce sobre outros. Esse grau de influência e ameaça é calculado com base em cinemática, dependendo da distância, velocidade e direção dos veículos. Tais informações cinemáticas são transmitidas periodicamente entre os veículos através de mensagens *beacon* (ou de sinalização).

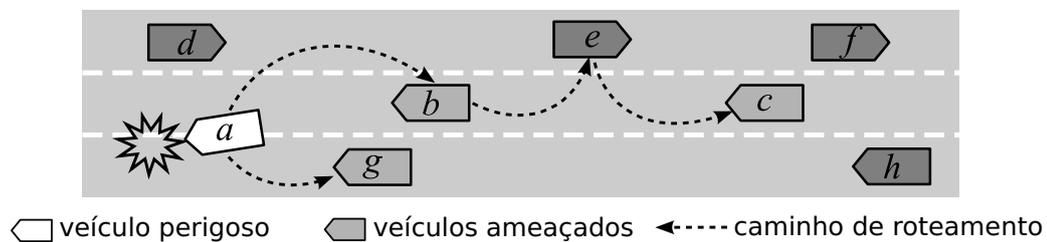


Figura 19: Um Esquema de Roteamento *Multicast* (SEBASTIAN et al., 2010)

Em Souza et al. (2011), estuda-se a influência da densidade de veículos em VANETs e uma possível adaptação às mudanças topológicas da rede. De fato, a análise da densidade em VANETs contribui para um melhor entendimento dos protocolos de roteamento em cenários veiculares. Quando um ambiente está densamente povoado de veículos, a probabilidade de colisão de pacotes será maior. Principalmente, se o tempo de backoff dos nós forem pequenos, significando que os nós terão um menor intervalo para se transmitir um pacote. Em ambientes esparsamente povoados de veículos, se o tempo de backoff for muito grande, os recursos da rede não serão bem aproveitados. Dessa forma, um mecanismo foi proposto para adaptar a janela de contenção e o tempo de backoff de cada veículo conforme o cenário veicular. A lógica *fuzzy* é usada para capturar informações de contexto e mensurar como mudar os parâmetros de camada de enlace para otimizar o acesso ao meio sem fio. Os experimentos foram feitos no simulador NCTUns e observou-se que o mecanismo proposto diminuiu a taxa de perdas de pacotes, aumentou a taxa de sucesso na entrega dos pacotes e melhor utilizou os recursos da rede.

Em Huang et al. (2007), o problema das mudanças topológicas da rede devido à mobilidade dos nós em redes *ad hoc* veiculares é tratado. Nesse trabalho, um mecanismo de melhoramento de enlace é proposto. Duas técnicas de aprendizado de máquinas (Otimização por Enxame de Partículas e Lógica *Fuzzy*) são incorporadas ao esquema proposto para garantir a precisão da predição da falha de enlaces da rede. Através da análise do padrão de mobilidade

dos veículos pode-se prever quando um enlace irá falhar. Essa análise é feita por um sistema *Fuzzy* que utiliza a distância entre dois veículos, a idade do motorista e a velocidade do veículo para estimar quando um veículo estará fora do alcance de transmissão de outros veículos. Se o sistema *Fuzzy* detectar que uma falha de enlace está perto de acontecer, ele faz com que o nó envie uma mensagem informando aos outros nós para que se inicie uma construção de rota alternativa àquele enlace que está para falhar. Além disso, um esquema de Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*) é usado para configurar os parâmetros do sistema *Fuzzy*. O esquema foi comparado com protocolos *unicast* e *multicast* e observou-se que o esquema proposto obteve melhores resultados.

Em Wang, Yang e An (2009), a preocupação principal é com a estabilidade das rotas em protocolos *unicast* em redes *ad hoc* veiculares. Um sistema de lógica *Fuzzy* é usado para ajudar na escolha de rotas. A proposta é levar em conta dois parâmetros para melhorar a estabilidade do protocolo AODV em VANETs: tempo de vida da rota e sentido para onde veículos se dirigem. Quando um nó deseja se conectar a outro, ele manda uma requisição de rota (RREQ) que vai armazenando, na própria mensagem, o menor tempo de vida de um enlace da rota. Além disso, a RREQ também manda informações de mobilidade dos veículos e o ângulo entre os veículos. Esse ângulo é necessário para o cálculo para saber em que sentido e direção estão indo os veículos. As informações de mobilidade servem para estimar o tempo de vida de cada enlace, e, por consequência, o tempo de vida da rota. Dessa forma, quando várias RREQs chegam ao destino, o destino roda um sistema *Fuzzy* para, probabilisticamente, decidir para qual rota ele vai mandar uma replicação de rota (RREP). A rota que tiver mais veículos indo no mesmo sentido e que tiver o maior tempo de vida terá maior probabilidade de ser escolhida. A proposta desse trabalho, chamada de Fcar (*Fuzzy control based AODV routing*), foi testada e comparada com o protocolo AODV. As simulações foram feitas em cenário urbano e o Fcar obteve melhor desempenho do que o AODV.

Em Maia e Celestino (2009), foi proposto um modelo analítico para analisar a qualidade de serviço em árvores *multicast/geocast* em redes *ad hoc* veiculares. Uma árvore é montada quando um nó quer transmitir uma mensagem. Ele busca por ERCs (*Emitter and Receiver Centers*), que são uma espécie de RSUs (dispositivos VANETs de acostamentos) e que irão encaminhar essa mensagem para outros veículos usando inundação. Em termos de estrutura da árvore, os ERCs são os vértices da árvore e o nó emissor é a raiz. Os principais parâmetros de qualidade de serviço analisados são diferentes tipos de atrasos fim-a-fim. Para modelar o atraso fim-a-fim das árvores *multicast*, foi utilizada uma distribuição tipo-fase e cadeias de Markov. Alguns importantes e conhecidos tipos de atrasos apresentados são: Δ_{max} - o atraso máximo de uma árvore *multicast*; Δ_{min} - o atraso mínimo de uma árvore *multicast*; e o $\delta = \Delta_{max} - \Delta_{min}$ - variação entre o atraso máximo e o mínimo da árvore *multicast*. Finalmente, mostra-se como aproximar as distribuições de atraso fim-a-fim em roteamento *multicast* baseado em árvore para redes *ad hoc* veiculares.

3.4 Considerações Finais

A seguir, as Tabelas 1 e 2 resumem os trabalhos relacionados.

	(CHANG; LIN; LIANG, 2008)	(YAN; WAN; FAN, 2010)	(ZHU; LIAN, 2008)
Roteamento	<i>Multicast</i>	<i>Multicast</i>	<i>Multicast</i>
Estrutura	Árvore	Árvore	Árvore
Protocolo-Base	MAODV	MAODV	MAODV
Tipo de Rede	MANET	MANET	MANET
Resumo	Reparo de Rotas por Listas de Vizinhança	Construção de Árvores com Rotas Estáveis	QoS com Predição Veicular
	(CHEN; ZHONG; LIU, 2009)	(VAISHAMPAYAN, 2004)	(JIE et al., 2009)
Roteamento	<i>Multicast</i>	<i>Multicast</i>	<i>Multicast</i>
Estrutura	Árvore	Árvore	Árvore
Protocolo-Base	MAODV	MAODV	MAODV
Tipo de Rede	MANET	MANET	MANET
Resumo	Reparo de Rotas e Melhoria na Inundação de RREQ-J	Merge de Árvores e Atualização de Rotas	Construção de Árvores Estáveis
	(HUANG; HAN; HOU, 2007)	(TIRUCHENGODE, 2009)	(SHEN et al., 2008)
Roteamento	<i>Multicast</i>	<i>Multicast</i>	<i>Multicast</i>
Estrutura	Árvore	Árvore	Árvore
Protocolo-Base	-	-	-
Tipo de Rede	Infraestruturada	MANET	MANET
Resumo	Otimização de Árvore usando ACO	Otimização de Árvore usando ACO	Economia de Energia dos Nós usando ACO
	(SEBASTIAN et al., 2010)	(SOUZA et al., 2011)	(HUANG et al., 2007)
Roteamento	<i>Multicast</i>	<i>Broadcast</i>	<i>Unicast</i>
Estrutura	Árvore	-	-
Protocolo-Base	-	-	-
Tipo de Rede	VANET	VANET	VANET
Resumo	Transmissão de Mensagens de Emergência	Ajuste de <i>Backoff</i> por Densidade	Predição de Falha usando Mobilidade, <i>Fuzzy</i> e PSO
	(WANG; YANG; AN, 2009)		
Roteamento	<i>Unicast</i>		
Estrutura	-		
Protocolo-Base	AODV		
Tipo de Rede	VANET		
Resumo	Escolha de Melhor Rota usando <i>Fuzzy</i> e Predição de Mobilidade		

Tabela 1: Resumo dos Trabalhos Relacionados.

	(MANGAI; TAMILARASI; VENKATESH, 2008)	(MAIA; CELESTINO, 2009)
Roteamento	<i>Multicast</i>	<i>Multicast</i>
Estrutura	Malha	Árvore
Protocolo-Base	DCMP	-
Tipo de Rede	MANET	VANET
Resumo	Uso de ACO para Otimização da Malha	Modelo Analítico para QoS
	(CORREIA; CELESTINO; CHERKAOUI, 2011)	(BIRADAR; MANVI; REDDY, 2010)
Roteamento	<i>Unicast</i>	<i>Multicast</i>
Estrutura	-	Malha
Protocolo-Base	-	-
Tipo de Rede	VANET	MANET
Resumo	Uso de ACO para Escolha de Rota	Monitoramento de Enlaces
Resumo	usando Predição de Mobilidade	para Otimização da Malha

Tabela 2: Resumo dos Trabalhos Relacionados.

De acordo com as tabelas apresentadas, esses trabalhos possuem elementos que estão presentes na proposta do MAV-AODV. Porém, segundo nosso conhecimento da literatura, não existe uma proposta para construir árvores de protocolos *multicast* tradicionais duráveis juntando princípios da metaheurística Otimização por Colônias de Formigas e de predição de tempo de vida de rotas *multicast* com mensagens *beacons* e cálculos cinemáticos em redes *ad hoc* veiculares.

4 O PROTOCOLO DE ROTEAMENTO *MULTICAST* MAV-AODV

O roteamento *multicast* é o método mais eficiente de comunicação em grupo, superando o roteamento *unicast* e *broadcast*. Além disso, com o desenvolvimento de mais tecnologias em redes sem fio, os fornecedores de conteúdo e serviço estão cada vez mais interessados em formas de comunicação em grupo. Porém, existem várias dificuldades no processo de roteamento *multicast* em redes sem fio (VARSHNEY, 2002). Mais especificamente em VANETs, o roteamento *multicast* torna-se ainda mais desafiador por causa das rápidas mudanças topológicas causadas pela mobilidade dos veículos (MOHAPATRA; LI; GUI, 2005; MUÑOZ, 2009; SEBASTIAN et al., 2010). Além disso, a alta velocidade dos nós e a mudança de densidade dos ambientes veiculares causam grandes problemas aos protocolos *multicast*. Um desses problemas é a constante quebra de enlaces das estruturas *multicast*. Quando um enlace quebra, vários membros da estrutura *multicast* podem deixar de receber os dados do grupo *multicast*. Além disso, para tentar reparar tais danos, os protocolos de roteamento *multicast* gastam muitos recursos da rede trocando mensagens de controle.

Neste trabalho, utilizamos uma estrutura que, comparada com as outras estruturas, fornece objetividade no encaminhamento dos dados (evitando redundância de dados como na estrutura baseada em malha). Tal estrutura é chamada de estrutura *multicast* baseada em árvore. Porém, ela, de uma forma geral, é uma estrutura que não fornece robustez. Isso acontece porque quando os nós se movimentam, os caminhos podem quebrar facilmente sem ter caminhos alternativos (JUNHAI; LIU; DANXIA, 2008; YANG; WU, 2005). Por isso, tentamos usar a eficiência da estrutura baseada em árvore e dar estabilidade a essa estrutura por desenvolvermos um protocolo *multicast* chamado MAV-AODV.

O MAV-AODV (*Multicast with Mechanisms of Ant Colony Optimization for Vanets based MAODV protocol*) é um protocolo que usa os princípios do MAODV (YANG; WU, 2005; ROYER; PERKINS, 1999) para fazer o roteamento *multicast* em redes ad hoc veiculares. O protocolo MAV-AODV utiliza informações de mobilidade veicular para tentar dar estabilidade ao roteamento *multicast* em VANETs. Além disso, o MAV-AODV usa mecanismos baseados em Otimização por Colônias de Formigas para otimizar a construção e manutenção das árvores *multicast*. As mensagens do protocolo agem como agentes bioinspirados (formigas) ao depositar feromônio para avaliar a atratividade de cada rota.

A seguir, descrevemos o protocolo que implementamos para transmissão de mensagens *beacons* e alguns cálculos de predição de mobilidade. Esses dois tópicos são necessários ao funcionamento do nosso protocolo MAV-AODV e atuam juntamente com ele.

Logo após, partimos para a descrição do MAV-AODV. Descrevemos o que foi necessário alterar no protocolo MAODV e explicamos os principais procedimentos do MAV-AODV.

4.1 Mensagens *Beacons*

Para o MAV-AODV poder escolher rotas estáveis com base no tempo de vida de cada rota/enlace é necessário haver uma predição de mobilidade. No ambiente veicular, existem mensagens chamadas *beacons*. Tais mensagens são periódicas e servem para manter um nó consciente de sua vizinhança e de sua movimentação. As mensagens *beacons* são usadas para evitar eventos como uma ultrapassagem errada, uma troca de faixa não permitida, colisão de veículos etc. Além disso, essas mensagens podem ser usadas para outras aplicações, como por exemplo, protocolos de roteamento. Essencialmente, as mensagens *beacons* contém informações sobre o estado do nó que as enviam. Supondo que os nós/veículos possuam GPS (*Global Positioning System*), eles podem divulgar suas posições, direções, velocidades e outras informações agregadas sobre o estado de alguns vizinhos. Além do mais, geralmente, essas mensagens são enviadas via *broadcast* de no máximo um salto, ou seja, um veículo envia suas mensagens *beacons* para, no máximo, todos os veículos que estão dentro de seu alcance de transmissão (ver Figura 20). O intervalo de transmissão entre mensagens *beacons* é, comumente, em torno de 1 segundo. Também o tempo de transmissão *interbeacon* é uniformemente distribuído num intervalo fechado entre 0.5 e 1.5 segundos para evitar sincronização e colisão de *beacons* (FRACCHIA; MEO; ROSSI, 2006; YOUSEFI; FATHY; BENSLIMANE, 2007; YOUSEFI; MOUSAVI; FATHY, 2006).

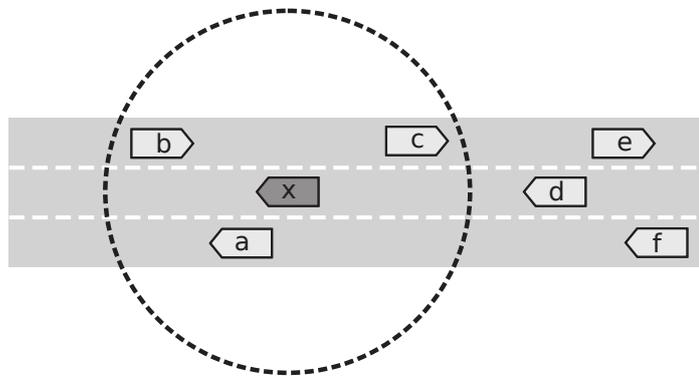


Figura 20: Alcance de Transmissão/Recepção do Veículo *x*.

Na Figura 20, podemos observar que o nó *x* só pode transmitir sua mensagem *beacon* para os nós que estão dentro de seu alcance de transmissão, ou seja, a um salto de distância. Então, o nó *x* pode transmitir somente para os nós *a*, *b* e *c*.

Neste trabalho, as mensagens *beacons* têm os campos mostrados na Figura 21, que possibilitam que um veículo possa fazer previsões de mobilidade acerca dos outros veículos. Consideramos o ambiente veicular como sendo um plano de eixos X e Y. Além disso, o protocolo de mensagens *beacons*, desenvolvido neste trabalho, envia as mensagens em *broadcast* de, no máximo, um salto a cada *y* segundos, onde $y \in [0.5, 1.5]$. Uma outra ação que o protocolo de mensagens *beacons* toma é calcular a estimativa do tempo de vida dos enlaces da rede. Esse

cálculo será explicado em detalhes na Seção 4.2.3.

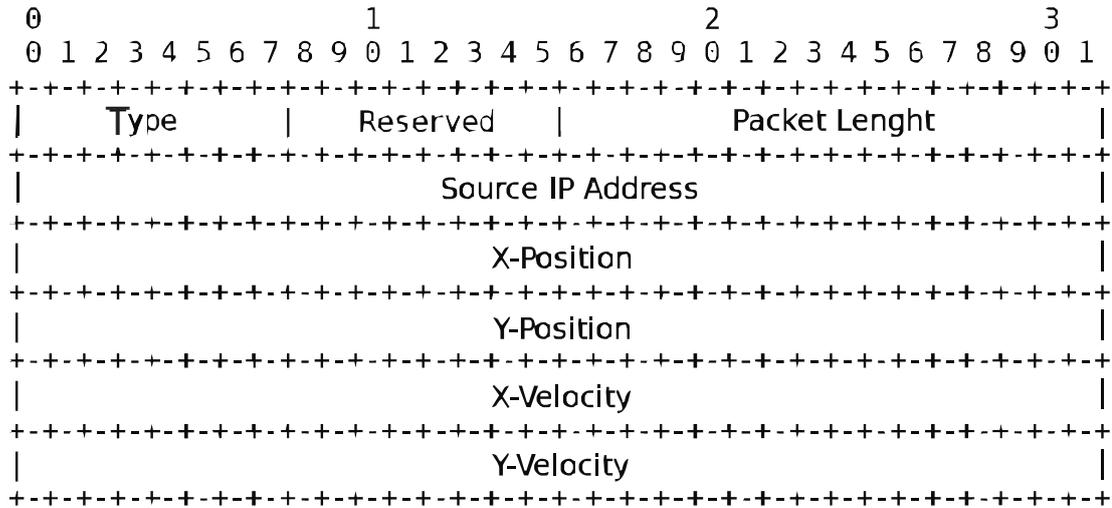


Figura 21: Formato da Mensagem *Beacon* do Protocolo Implementado neste Trabalho.

Os campos da mensagem *beacon* são os seguintes:

- *Type* - Tipo de mensagem. Identifica a mensagem *beacon*.
- *Reserved* - Enviado como 0. Ignorado na recepção.
- *Packet Length* - Comprimento da mensagem.
- *Source IP Address* - O endereço IP do nó que está enviando a mensagem.
- *X-Position* - Posição no eixo X do nó que está enviando a mensagem.
- *Y-Position* - Posição no eixo Y do nó que está enviando a mensagem.
- *X-Velocity* - Vetor velocidade no eixo X do nó que está enviando a mensagem.
- *Y-Velocity* - Vetor velocidade no eixo Y do nó que está enviando a mensagem.

Essas informações capacitam os veículos a fazer cálculos de predição de mobilidade de outros veículos, conforme explicado na seção seguinte.

4.2 Predição de Mobilidade

Através da disseminação de informações de mobilidade pelas mensagens *beacons*, é possível obter informações de contexto e prever a mobilidade de determinados veículos. A seguir, veremos como podemos obter tais informações.

4.2.1 Distância entre Veículos e Modelos de Rádio Propagação

A distância entre veículos é uma informação importante em redes veiculares. Essa informação é fundamental para tomadas de decisão de aplicações que fornecem segurança para o condutor e os passageiros do veículo. Com a informação da distância entre veículos é possível saber se existe risco de colisão iminente. Nesse caso, um veículo munido de sensores e equipamentos de segurança poderia acionar mecanismos para evitar a colisão (freios automáticos) ou diminuir o impacto da colisão (acionando *air bags* e regulando os cintos de segurança).

A distância entre veículos também é importante para saber se haverá recepção de mensagens. Por exemplo, se a distância entre dois veículos for muito grande, a probabilidade de comunicação entre esses veículos será muito pequena. Por isso, essa informação é muito importante para protocolos de roteamento em redes ad hoc.

A distância entre os veículos, junto com um modelo de rádio propagação, fornece uma boa medida para a probabilidade de recepção de mensagens. Por exemplo, usando um modelo de rádio propagação determinístico, dá para saber se uma mensagem será entregue com sucesso ou não.

O modelo de rádio propagação determinístico *Two-Ray Ground* utiliza a distância entre veículos para saber se uma mensagem será entregue. O *Two-Ray Ground* funciona de uma tal maneira que, se dois nós estiverem em qualquer posição, de modo que a distância entre eles seja menor que uma distância d , a força do sinal recebido será a mesma. Esse modelo é muitas vezes usado em processos de avaliação ou projeto de protocolos de comunicação sem fio. A vantagem está em sua fácil compreensão e interpretabilidade nas simulações (TORRENT-MORENO et al., 2006; STEPANOV; HERRSCHER; ROTHERMEL, 2005). A fórmula para o modelo de propagação *Two-Ray Ground* é a seguinte:

$$P_r = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad (4.1)$$

Conforme a equação 4.1, P_r é a potência do sinal recebido (em W), P_t é a potência do sinal transmitido, G_t e G_r são, respectivamente, os ganhos de transmissão e recepção das antenas, h_t e h_r são, respectivamente, as alturas das antenas de transmissão e recepção, d é a distância entre o transmissor e o receptor e L é a perda do sistema.

Entretanto, o modelo *Two-Ray Ground* é muito otimista para distâncias curtas. Então, muitos trabalhos usam uma combinação de modelos. Usam o *Two-Ray Ground* para distâncias maiores entre os nós e usam o modelo *Free Space* (ver equação 4.2) para distâncias menores (STEPANOV; HERRSCHER; ROTHERMEL, 2005).

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^4 L} \quad (4.2)$$

Conforme a equação 4.2, do modelo de propagação *Free Space*, λ é o comprimento de onda.

Entretanto, um outro tipo de predição ainda pode ser feito.

4.2.2 Predição de Posicionamento de Vizinhança em um Dado Momento

A partir das informações de mobilidade adquiridas das mensagens *beacons*, é possível também estimar a posição de vizinhos em um dado momento futuro por meio de conceitos cinemáticos simples. Bastando para isso que os parâmetros de mobilidade, tais como a velocidade, a distância entre os veículos e a direção, não variem drasticamente.

Para calcular a posição futura de nós vizinhos, Härrri, Bonnet e Filali (2008) propuseram um *framework* de grafos cinéticos. Tal *framework* definiu quatro parâmetros para descrever a mobilidade de um veículo: x (posição do nó no eixo x), y (posição do nó no eixo y), dx (vetor velocidade do nó no eixo x) e dy (vetor velocidade do nó no eixo y). Esses parâmetros correspondem, respectivamente, aos campos *X-Position*, *Y-Position*, *X-Velocity* e *Y-Velocity* da mensagem *beacon* utilizada em nossos cenários veiculares. Assume-se que os nós seguem uma trajetória linear num período de tempo relativamente curto. A posição desse nó/veículo em função do tempo é dada pela equação 4.3, descrita abaixo:

$$Pos_i(t) = \begin{bmatrix} x_i + dx_i \cdot t \\ y_i + dy_i \cdot t \end{bmatrix}, \quad (4.3)$$

onde $Pos_i(t)$ é a posição do nó/veículo i no tempo t .

Agora vamos considerar um veículo j como sendo vizinho do veículo i . Para extrair a distância relativa futura do veículo j e calcular sua trajetória, precisa-se definir a distância ao quadrado entre os veículos i e j , como mostrado, pela equação 4.4, abaixo:

$$D_{ij}^2 = D_{ji}^2 = \|Pos_j(t) - Pos_i(t)\|^2 = \left(\begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} + \begin{bmatrix} dx_j - dx_i \\ dy_j - dy_i \end{bmatrix} \cdot t \right)^2 \quad (4.4)$$

Após efetuarmos os cálculos da equação 4.4, obtemos a seguinte fórmula:

$$D_{ij}^2 = a_{ij}t^2 + b_{ij}t + c_{ij}, \quad (4.5)$$

onde $a_{ij} \geq 0$, $c_{ij} \geq 0$. Dessa forma, os três parâmetros que descrevem a trajetória mútua dos veículos i e j são: a_{ij} , b_{ij} e c_{ij} . Além disso, a trajetória relativa de i para j , representando a distância relativa entre i e j é $D_{ij}^2(t) = a_{ij}t^2 + b_{ij}t + c_{ij}$.

A tabela 3 resume todos os parâmetros utilizados nesta subseção.

Parâmetros	Significados
x_i	Posição do nó i no eixo X
y_i	Posição do nó i no eixo Y
dx_i	Vetor velocidade do nó no eixo X
dy_i	Vetor velocidade do nó no eixo Y
t	Tempo em um dado momento
$Pos_i(t)$	Posição do nó i no tempo t
D_{ij}	Distância entre os nós i e j
a_{ij}	Valor associado à t^2 (fórmula 4.5), obtido depois de se calcular a equação 4.4 e colocar t^2 em evidência
b_{ij}	Valor associado à t (fórmula 4.5), obtido depois de se calcular a equação 4.4 e colocar t em evidência
c_{ij}	Valor não associado à t e nem à t^2 (fórmula 4.5), obtido depois de se calcular a equação 4.4 e colocar t e t^2 em evidência

Tabela 3: Parâmetros Usados para Predição de Posicionamento.

4.2.3 Estimativa de Tempo de Vida de Enlace

As informações de mobilidade fornecidas pelas mensagens *beacons* ainda nos possibilitam calcular outro parâmetro de mobilidade: o tempo de vida dos enlaces. Conforme Härri, Bonnet e Filali (2008) e Menouar, Lenardi e Filali (2007), o tempo de vida do enlace (i, j) corresponde ao tempo estimado $t_{link} = t_1 - t_0$, onde t_1 é o tempo quando a distância d entre os dois nós se torna maior que o alcance de comunicação R (isto é, o tempo quando j sai do alcance de comunicação de i) e t_0 é o tempo inicial dos nós.

Assim, para calcular t_{link} , precisamos resolver a seguinte inequação: $D_{ij}^2(t) \leq R^2$. Isso significa que os nós serão vizinhos até quando a distância entre eles ao quadrado for menor ou igual ao alcance de comunicação máximo deles ao quadrado. Dessa forma, para saber o valor de t_{link} , obtemos a equação 4.6 a seguir:

$$D_{ij}^2(t) - R^2 = 0 \quad (4.6)$$

Como já sabemos o valor de $D_{ij}^2(t)$, obtemos a equação 4.7 abaixo:

$$a_{ij}t^2 + b_{ij}t + c_{ij} - R^2 = 0, \quad (4.7)$$

onde o t da equação 4.7, corresponde ao t_{link} que queremos encontrar. Dessa maneira, para calcular o t_{link} , basta resolver a equação do segundo grau mostrada na equação 4.7.

Entretanto, essa maneira de calcular t_{link} ainda precisa de um pequeno reparo. Se dois veículos tiverem mobilidade muito semelhante (por exemplo, dois veículos indo na mesma direção, próximos um do outro e com velocidades parecidas), o tempo de vida do enlace tende a ser infinito. Para solucionar essa questão, Namboodiri e Gao (2007) propôs um limite superior, uma constante limitante, para valores muito grandes de tempos de vida dos enlaces estimados. No nosso trabalho, esse limite superior é dado por $t_{maxlifetime}$. Por exemplo, se o valor estimado de um tempo de vida de um enlace for maior do que $t_{maxlifetime}$ segundos, esse valor passa a ser $t_{maxlifetime}$ segundos.

Vale ressaltar ainda que, para saber o tempo de vida de um caminho, basta buscar pelo menor valor de um enlace desse caminho, como mostrado na equação 4.8 a seguir:

$$t_{path} = \min_{\forall link \in path} \{t_{link}\} \quad (4.8)$$

A tabela 4 resume os parâmetros mencionados nesta subseção.

Parâmetros	Significados
d	Distância entre dois nós
R	Alcance de comunicação de um nó
t_1	Tempo quando a distância d se torna maior que R
t_0	Tempo inicial de um nó
t_{link}	Tempo de vida de um enlace
$t_{maxlifetime}$	Valor máximo que o tempo de vida de um enlace pode ter
t_{path}	Tempo de vida de um caminho

Tabela 4: Parâmetros Usados para Calcular o Tempo de Vida de Enlaces.

Todas essas informações podem ser usadas em avaliação e projeto de protocolos de roteamento em redes ad hoc móveis. Muitos trabalhos usaram essas informações para aprimorar o desempenho de determinados protocolos. No nosso trabalho, utilizamos a estimativa de tempo de vida de enlace para projetar o protocolo de roteamento *multicast* MAV-AODV. Quando uma mensagem *beacon* gerada por um nó x chega em um nó y , o nó y , através do protocolo de mensagens *beacons*, usa o cálculo de estimativa de tempo de vida de enlace para prever quando o enlace ($x \leftrightarrow y$) irá quebrar. Essa estimativa fica armazenada em cada um dos nós da rede, sendo constantemente atualizadas. Dessa forma, quando uma mensagem RREQ-J do protocolo MAV-AODV chegar em um determinado nó, ela contará com a estimativa de tempo de vida de enlace atualizada. Isso será melhor descrito na seção 4.3.4.

4.3 Especificações do Protocolo de Roteamento *Multicast* MAV-AODV

4.3.1 Alterações sobre o MAODV

Foi necessário fazer algumas alterações no MAODV para que o MAV-AODV fosse concebido. As principais alterações sobre o MAODV foram feitas:

- Nas Tabelas de Roteamento;
- Nas Mensagens de Roteamento;
- Na Função de Requisição de Rotas;
- Na Função de Resposta de Rotas;
- Na Função de Ativação de Rota.

4.3.2 Tabelas de Roteamento

Cada nó que tem MAV-AODV funcionando mantém, obrigatoriamente, duas tabelas de roteamento. A primeira delas é a Tabela de Rotas. Essa tabela não foi alterada e permaneceu conforme a descrição do protocolo MAODV. A tabela de rotas é usada para registrar os próximos saltos para determinada rota na rede. Conforme a Tabela 5, os campos da tabela de rotas são os seguintes:

Endereço de IP do destino
Número de sequência do destino
Contador de saltos para o destino
Próximo salto
Tempo de vida

Tabela 5: Tabela de Rotas do MAV-AODV/MAODV.

Novas entradas são colocadas na tabela quando RREQs e RREPs são recebidos. Associado com cada entrada está o tempo de vida, indicando a quantidade de tempo que aquela entrada é válida. Rotas são excluídas da tabela se não houver atualizações ou se a rota não for usada com o tempo de vida indicado.

A segunda tabela de roteamento que um nó mantém é a Tabela de Rotas *Multicast*. Essa tabela contém entradas para o grupo *multicast* de que o nó é membro. No MAV-AODV, essa tabela foi alterada para armazenar um campo chamado de Feromônio. De acordo com a Tabela 6, cada entrada na tabela de rotas *multicast* contém as seguintes informações:

Endereço IP do grupo <i>multicast</i>
Endereço IP do líder do grupo <i>multicast</i>
Número de sequência do grupo <i>multicast</i>
Contador de saltos para o líder do grupo <i>multicast</i>
Contador de saltos para um membro da árvore <i>multicast</i>
Tempo de vida
Próximos saltos
— Endereço IP do próximo salto
— Direção do enlace
— Flag ativada
Feromônio

Tabela 6: Tabela de Rotas *Multicast* do MAV-AODV

Novas entradas são colocadas nessa tabela depois de um nó tornar-se um roteador (ou membro da árvore *multicast*) para um grupo *multicast*. Associado com cada entrada de Próximo Salto está um *Flag* Ativado. Esse *flag* é usado para indicar se o enlace tem sido oficialmente adicionado à árvore *multicast*. O *Flag* Ativado de uma entrada de Próximo Salto é acionado somente depois da ativação de uma rota pela recepção de uma mensagem de Ativação *Multicast* (MACT), como descrito nas seções seguintes. Para entradas de rotas *multicast*, podem existir mais do que uma entrada de Próximo Salto. Tal caso, acontece quando um nó, numa árvore, tem mais de um nó-filho.

O campo Feromônio baseia-se no conceito de Colônias de Formigas e indica a atratividade de uma rota. Conforme veremos nas seções seguintes, quanto maior essa quantidade, maior será a chance de a rota, com esse nível de feromônio, ser escolhida.

4.3.3 Formato das Mensagens de Roteamento

Das quatro mensagens de controle de roteamento do MAODV, só duas foram alteradas para o desenvolvimento do MAV-AODV: as mensagens RREQ e RREP. As mensagens MACT e GRPH continuaram inalteradas.

4.3.3.1 RREQ - *Route REQuest*

O formato da mensagem de Requisição de Rota é ilustrado na Figura 22. Tal formato segue todas as especificações do protocolo AODV. Mas, para o MAODV foram acrescentados dois novos campos:

- *J - Join Flag*; ativada quando um nó quer se juntar a um grupo *multicast*.

- R - *Repair Flag*; ativada quando um nó quer iniciar um reparo para conectar duas porções previamente desconectadas da árvore *multicast*.

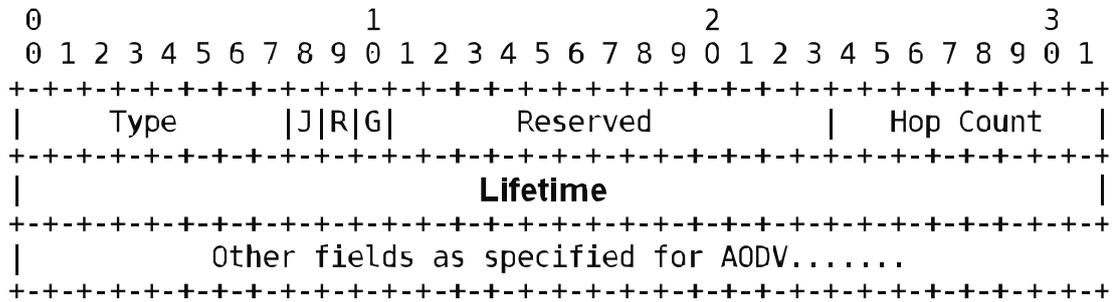


Figura 22: Formato da Mensagem RREQ.

Do MAODV para o MAV-AODV, a única alteração foi o acréscimo de um novo campo chamado *Lifetime*. Como veremos nas seções seguintes, esse campo serve para armazenar o menor tempo de vida de um enlace de uma determinada rota.

4.3.3.2 RREP - *Route REPLY*

O formato da mensagem de Resposta de Rota é ilustrado na Figura 23. Tal formato segue todas as especificações do protocolo AODV. Mas, para o MAODV foi acrescentado um novo campo:

- R - *Repair flag*; ativado quando um nó está respondendo a uma requisição de reparo para conectar duas porções previamente desconectadas da árvore *multicast*.

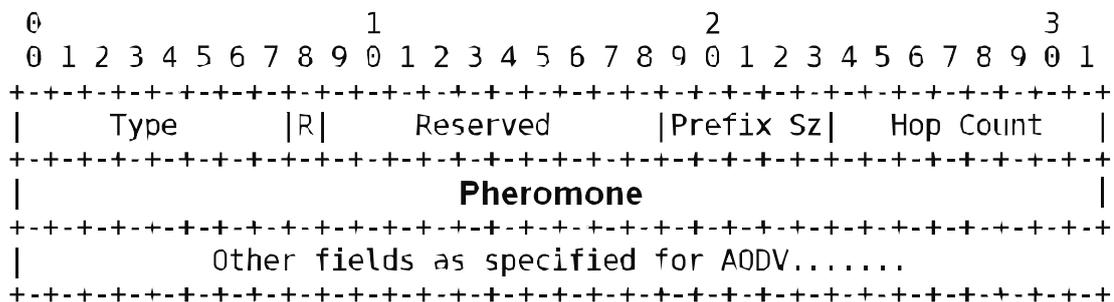


Figura 23: Formato da Mensagem RREP.

Do MAODV para o MAV-AODV, foi acrescentado um novo campo chamado de *Pheromone*. Como veremos nas seções seguintes, tal campo serve para avaliar e registrar a atratividade/qualidade de uma rota em um dado momento.

4.3.4 Geração de Mensagem de Requisição de Rota

Um nó envia uma mensagem *Join* de requisição de rota (RREQ-J) quando ele quer se juntar a um grupo *multicast*, ou quando ele tem dados para enviar para o grupo *multicast* e não tem uma rota para tal grupo. O Dest-Addr dessa RREQ é o endereço IP do grupo *multicast* desejado, e o número de sequência do destino é o último número de sequência conhecido para aquele grupo. Se o nó quer se juntar a um grupo *multicast*, ele marca um campo na mensagem RREQ chamado de *Join Flag*, senão, ele não marca esse campo.

No MAODV, a RREQ-J pode ser enviada em *unicast* ou em *broadcast* dependendo da informação disponível no nó origem. Se o nó tem um registro do líder do grupo *multicast* antes de requisitar uma rota para o grupo *multicast*, e se o nó solicitante tem uma rota válida para tal líder, ele insere na mensagem RREQ-J, o endereço IP do líder do grupo e envia a RREQ-J em *unicast* através do caminho conhecido para o líder do grupo. Mas, se o nó fonte não sabe quem é o líder do grupo ou se ele não tem uma rota para o líder do grupo, ele envia a RREQ-J em *broadcast*.

Entretanto, no MAV-AODV, a RREQ-J só pode ser enviada em *broadcast*, mesmo que um nó saiba o endereço IP do líder do grupo *multicast* e tenha uma rota válida para ele. Isso foi feito para permitir a diversificação no espaço de busca. Mais rotas poderão ser avaliadas e existirá mais opções no momento da escolha da melhor rota para um grupo *multicast*. Além disso, com a mobilidade dos nós, uma rota válida para o líder, que antes era uma rota boa (em termos de estabilidade), em outro momento, pode não ser mais boa. Os nós daquela rota podem ter se movido, modificando a qualidade da rota.

A Figura 24 ilustra a propagação de RREQ-J (representada como formiga) em *broadcast* utilizando o protocolo MAV-AODV. Nessa figura, o nó *a* está solicitando entrar no grupo *multicast* que o nó *i* é membro.

Conforme a Figura 24, no processo de envio da RREQ-J até o nó destino, cada RREQ-J carrega o tempo de vida da rota, no campo *Lifetime* da mensagem. O tempo de vida da rota é o menor tempo de vida de um enlace da rota (em valores estimados). A estimativa de tempo de vida de enlace é feita pelo protocolo de mensagens *beacons* (ver Seção 4.1 e 4.2.3), que atualiza constantemente essa estimativa para torná-la disponível ao MAV-AODV. Dessa forma, quando uma mensagem RREQ-J chega em um nó, uma checagem é feita para saber se o tempo de vida da rota é menor ou maior do que o tempo de vida do último enlace. Se o tempo de vida do enlace for menor do que o da rota, o novo tempo de vida da rota que a RREQ-J vai carregar é o tempo de vida desse último enlace. Senão, o tempo de vida da rota permanecerá o mesmo.

Somente um membro da árvore *multicast* desejada (isto é, um roteador para o grupo *multicast*) pode responder a uma mensagem RREQ-J. Se não é uma RREQ-J, qualquer nó com uma rota válida para o grupo *multicast* pode responder. Se um nó recebe uma RREQ-J para

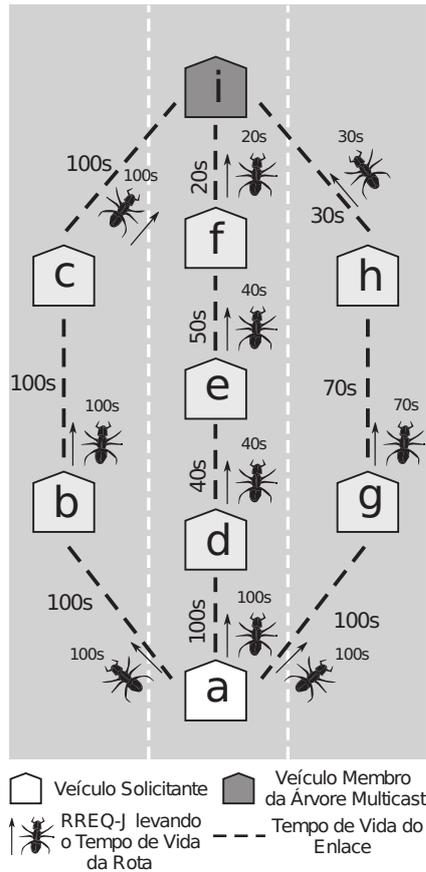


Figura 24: Exemplo de Requisição de Rota.

um grupo *multicast* do qual ele não é membro, ou se ele recebe uma RREQ-J e ele não tem uma rota para aquele grupo, ele novamente envia a mensagem RREQ-J em *broadcast* para seus vizinhos.

Se o nó origem não recebe uma RREP num tempo determinado, ele envia outra RREQ-J em *broadcast* com o campo *Broadcast-ID* incrementado em 1. Se ele não recebe uma RREP para essa RREQ-J, ele continua enviando *Join-RREQs* em *broadcast* até um limite de RREQ-RETRIES. Depois desse número de tentativas, ele assume que o grupo *multicast* é inalcançável, ou não existem outros membros daquele grupo *multicast* na sua porção conectada da rede. Nesse caso, o nó se torna o líder do grupo *multicast* e inicializa o número de sequência do grupo com o valor 1.

No MAV-AODV, as mensagens RREQ-J agem como formigas, utilizadas em problemas de otimização combinatória pela metaheurística Otimização por Colônia de Formigas, explorando o espaço de busca (caminhos da rede) à procura de alimento (no nosso caso, algum nó membro do grupo *multicast* buscado).

4.3.5 Estabelecimento de Rota Reversa

Um nó que recebe uma RREQ-J mantém uma entrada de rota correspondente na sua tabela de rotas *multicast* em adição à sua tabela de rotas (*unicast*). A *Flag* Ativada para essa entrada tem o valor FALSO e, somente depois de a rota ser selecionada para fazer parte da árvore *multicast*, essa *Flag* passar a ter o valor VERDADEIRO. Um nó só pode responder a uma RREQ-J se ele é um membro da árvore *multicast*, ou seja, se na sua tabela de rotas *multicast*, a *Flag* Ativada tiver o valor VERDADEIRO. A geração de RREPs é descrita a seguir.

4.3.6 Geração de Mensagem de Resposta de Rota

Se um nó recebeu uma RREQ-J para um grupo *multicast*, ele pode responder se ele é um roteador para a árvore do grupo *multicast* e se seu número de sequência registrado para o grupo *multicast* é pelo menos tão grande quanto o contido na RREQ-J. O nó que responde a uma RREQ-J atualiza sua tabela de rotas e de rotas *multicast* por colocar as informações de próximo salto do nó solicitante nas tabelas e gerar uma RREP.

Mas, antes de gerar a RREP, o nó que responde à mensagem RREQ-J, analisa dois campos da RREQ-J: *Lifetime* (ver Seção 4.2.3) e *Hop Count* (contador de saltos da rota percorrida pela RREQ-J). Quando uma RREQ-J chega no destino, ela chega com o menor tempo de vida de um enlace da rota, que é o valor *Lifetime*. Com base nesses valores, o nó aplica a equação 4.9 para calcular o feromônio da rota percorrida pela RREQ-J, como mostrado a seguir:

$$Pheromone(p) = \frac{Lifetime(p)}{HopCount(p)}, \quad (4.9)$$

onde p é o caminho percorrido pela RREQ-J.

O nó, então, coloca o valor de $Pheromone(p)$ no campo *Pheromone* da mensagem RREP. Logo em seguida, o nó envia a RREP em *unicast* de volta para o nó indicado pelo campo *Source-Addr* da RREQ-J recebida. A Figura 25 ilustra o caminho das RREPs (representadas como formigas) para o nó origem. As RREPs depositam o valor $Pheromone(p)$ nas tabelas de rotas *multicast* dos nós (campo Feromônio) durante todo o percurso de volta ao nó origem.

A RREP contém o último número de sequência conhecido para o grupo *multicast* e o endereço IP do líder do grupo *multicast*. Em adição, a RREP inclui um campo especial de extensão chamado MGROU-P-HOP. Esse campo é inicializado com zero e é incrementado a cada vez que o pacote é encaminhado. Quando uma RREP é recebida pelo nó origem, o MGROU-P-HOP indica a distância (em saltos) do nó origem para o membro mais próximo da árvore *multicast*. O endereço IP do líder do grupo é também colocado num campo de extensão chamado GROUP-LEADER-ADDR.

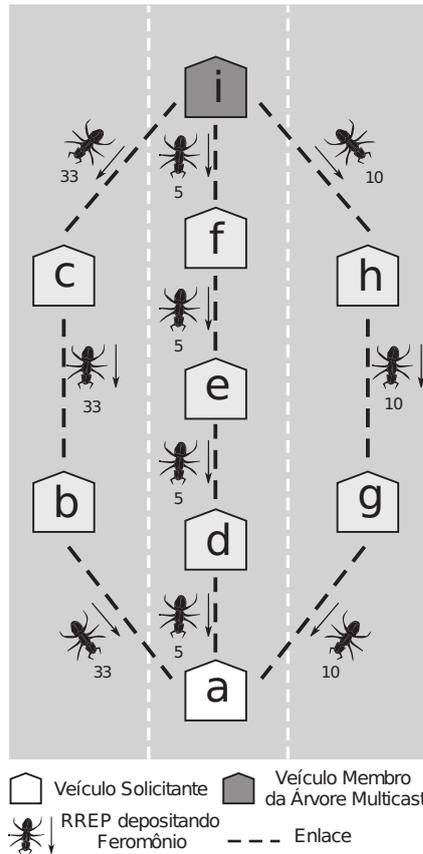


Figura 25: Exemplo de Resposta de Rota.

Quando os nós, ao longo do caminho para o nó origem, recebem uma RREP pela primeira vez, eles adicionam uma entrada para o nó de que eles receberam a RREP na tabela de rotas *multicast*, criando assim um percurso de encaminhamento.

Entretanto, mesmo esses nós intermediários podem receber mais de uma RREP para determinado par (origem-destino). Nesse caso, o MAV-AODV analisará alguns parâmetros para checar se a rota, da RREP, é válida e atualizada. Caso seja, o MAV-AODV escolherá, probabilisticamente, entre a RREP que já estava armazenada na tabela de rotas *multicast* e a RREP que acaba de chegar ao nó intermediário.

Esse processo de escolha probabilística se dá através de um mecanismo de roleta. Cada RREP tem uma quantidade de feromônio armazenada. A RREP que já estava na tabela de rotas *multicast* (chamaremos essa RREP de $RREP_{older}$) tem a quantidade de feromônio armazenada nessa própria tabela, que chamaremos de $Pheromone_{table}$. A RREP que acaba de chegar (chamaremos essa RREP de $RREP_{newer}$) tem a quantidade de feromônio armazenada na própria mensagem, no campo *Pheromone* (chamado de $Pheromone_{packet}$).

A Figura 26 ilustra como o processo de escolha é feito. A probabilidade de escolha de $RREP_{older}$ é dada por $P(older)$ e a probabilidade de escolha de $RREP_{newer}$ é dada por $P(newer)$.

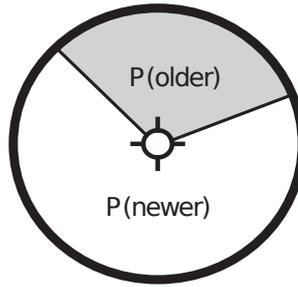


Figura 26: Exemplo do Processo de Escolha de RREPs por Roleta.

De acordo com a equação 4.10, $P(older)$ é calculada da seguinte maneira:

$$P(older) = \frac{Pheromone_{table}}{Pheromone_{table} + Pheromone_{packet}}. \quad (4.10)$$

De acordo com a equação 4.11, $P(newer)$ é calculada da seguinte forma:

$$P(newer) = \frac{Pheromone_{packet}}{Pheromone_{table} + Pheromone_{packet}}. \quad (4.11)$$

Conforme a Figura 26, a RREP que chegou depois tem uma quantidade maior de feromônio e uma maior probabilidade de ser escolhida. Se tal RREP, for escolhida, a tabela de rotas *multicast* terá que ser atualizada e essa RREP continuará a ser encaminhada até o nó origem. Senão, a tabela de rotas permanecerá inalterada e a RREP que chegou depois será descartada.

Além disso, se acontecer de chegar mais de uma RREP para o nó origem, esse processo de escolha será repetido para cada nova RREP que chegar até que um limite de tempo acabe. Após o limite de tempo acabar, um mecanismo de ativação de rota será acionado, conforme veremos em seções posteriores.

Dessa maneira, no protocolo MAV-AODV, as mensagens RREP atuam como formigas ao saírem depositando feromônio ao longo da rota de volta para o nó origem (ninho), após terem encontrado uma fonte de comida (no nosso caso, um membro da árvore *multicast*).

4.3.7 Mensagens *Group Hello*

O primeiro membro do grupo *multicast* se torna o líder para aquele grupo. Esse nó permanece sendo o líder do grupo até ele decidir deixar o grupo ou até haver uma junção de duas árvores separadas. O líder do grupo *multicast* é responsável por manter o número de sequência do grupo *multicast* e por disseminar esse número para o grupo *multicast*. Periodicamente (a cada GROUP-HELLO-INTERVAL segundos), o líder do grupo envia uma mensagem *Group Hello* em *broadcast*. A mensagem *Group Hello* é uma RREP não solicitada com um TTL maior

do que o diâmetro da rede para que seja propagada através da rede inteira. De acordo com Royer e Perkins (1999), um dos principais objetivos da mensagem *Group Hello* é manter todos os membros da árvore *multicast* com conhecimento atualizado a respeito do grupo *multicast* e de seu líder.

A *Group Hello* contém extensões que indicam os endereços IPs dos grupos *multicast* e seus números de sequência correspondentes de todos os grupos *multicast* de que o nó é líder de grupo. O número de sequência para o grupo é incrementado para cada envio de *Group Hello* em *broadcast* pelo líder do grupo. O HOP-CNT da *Group Hello* é inicializado com zero e é incrementado por cada nó que recebe a mensagem, indicando assim, a distância em saltos até o líder do grupo. Nós que são membros da árvore *multicast* usam a *Group Hello* para atualizar suas distâncias para o líder do grupo. A *Group Hello* também é usada para juntar árvores particionadas. Essa mensagem é usada nos protocolos MAODV e MAV-AODV.

4.3.8 Nó Interior da Árvore *Multicast* e Tratamento de Redundâncias

Em qualquer nó interior em uma árvore *multicast*, a entrada de rota na tabela de rotas *multicast*, para determinado grupo, pode ter múltiplos próximos saltos (Ver Figura 27). Quando um pacote de dados endereçado para o grupo *multicast* é recebido por um membro desse tipo, o endereço IP de origem e campos de identificação do cabeçalho IP do pacote são registrados. O pacote é então enviado em *multicast* para seus próximos saltos. Se o nó, além de ser membro da árvore, é também um membro do grupo, o pacote é processado. Na Figura 27, podemos observar que o nó interior X possui múltiplos saltos, no caso, dois saltos: os nós Y e Z.

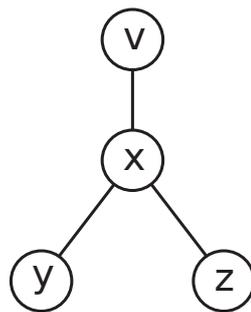


Figura 27: Exemplo de um Nó Interior numa Árvore *Multicast*.

Um nó numa árvore *multicast* pode receber o mesmo pacote de dados várias vezes. Por exemplo, se ele recebe um pacote de dados, ele retransmite o pacote para seus próximos saltos, e então recebe o mesmo pacote de dados quando seus próximos saltos retransmitem o pacote para os próximos saltos deles. O nó detectará essa redundância por checar o endereço IP de origem e os campos de identificação do cabeçalho IP, e então ele descartará o pacote (ROYER; PERKINS, 1999).

4.3.9 Ativação de Rota *Multicast*

A Figura 28 ilustra a ativação de um novo ramo de árvore *multicast* pelo protocolo MAV-AODV. Esta subseção explica como é feita essa ativação.

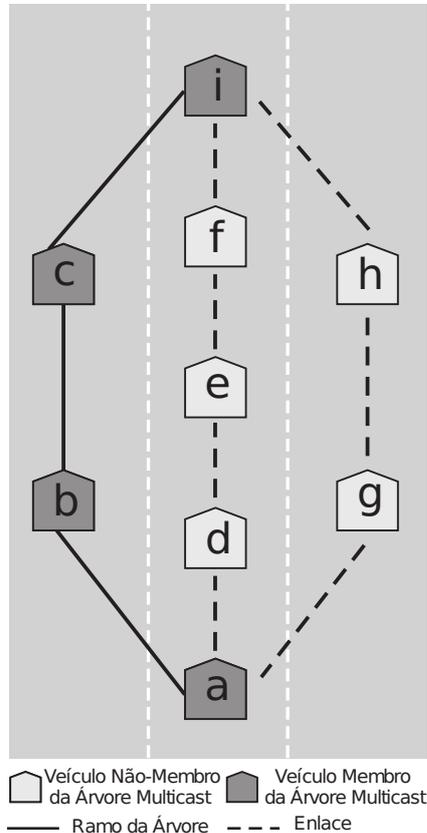


Figura 28: Exemplo de Criação de Ramo de Árvore.

Quando um nó origem envia uma RREQ-J em *broadcast* para um grupo *multicast*, ele espera receber mais do que uma resposta. Porém, um e somente um dos RREPs (em resposta às *Join-RREQs*) precisa ser selecionado como o próximo salto. Nesse sentido, somente um ramo é adicionado para a árvore e os *loops* são evitados. Isso é realizado como descrito a seguir.

O nó origem espera RTE-DISCOVERY-TIMEOUT milisegundos depois de enviar uma RREQ-J antes de selecionar uma rota. RTE-DISCOVERY-TIMEOUT é um parâmetro configurável que pode ser alterado de acordo com o tamanho da rede. Durante esse período de tempo, o nó terá maior probabilidade de escolher a rota recebida com o maior valor de feromônio, conforme descrito na Seção 4.3.6. No fim desse período, o nó habilita o próximo salto selecionado na sua tabela de rotas *multicast*, e então envia em *unicast* uma mensagem MACT (*Multicast Activation*) para seu próximo salto selecionado. Cada mensagem MACT contém os seguintes campos:

<P-FLAG, GL-FLAG, SOURCE-ADDR, SOURCE-SEQ-NUMBER, DEST-ADDR >

O DEST-ADDR é colocado como o endereço IP do grupo *multicast*. Os campos P-FLAG e GL-FLAG são usados para poda e escolha de novo líder de grupo (ROYER; PERKINS, 1999).

O próximo salto, quando recebe uma mensagem MACT, também habilita a rota para o nó origem na sua tabela de rotas *multicast*. Se esse nó é um membro da árvore *multicast*, ele não propaga mais a MACT, pois o novo ramo da árvore já foi inserido completamente. Entretanto, se esse nó não é um membro da árvore *multicast*, ele envia a MACT em *unicast* para o próximo salto, e habilita a rota correspondente na sua tabela de rotas *multicast*. Esse processo continua até se chegar ao nó que originou a RREP (porque tal nó era um membro da árvore *multicast*). Nós que geraram ou encaminharam RREPs e que não receberam uma MACT para ativar suas rotas depois de MTREE-BUILD milisegundos, excluem a rota para o nó solicitante.

A mensagem MACT garante que a árvore *multicast* não tenha múltiplos caminhos para qualquer nó da árvore (e, assim de fato, seja uma árvore). Nós somente encaminham pacotes de dados através de suas rotas ativadas nas suas tabelas de rotas *multicast* (*Flag* Ativada). Isso previne a possibilidade de pacotes de dados serem distribuídos para um nó origem por múltiplos caminhos, antes de uma MACT ser recebida.

5 SIMULAÇÃO E ANÁLISE DE RESULTADOS

Este capítulo apresenta os cenários em que os experimentos foram executados, mostra todos os detalhes dos experimentos realizados para a prova de conceitos do protocolo MAV-AODV e apresenta os resultados e suas respectivas análises.

5.1 Cenários

O cenário Manhattan é uma área urbana de 1600 x 1500 metros, feito de grids, como mostrado na figura 29. Cada avenida tem duas faixas e o tráfego veicular flui nos dois sentidos.

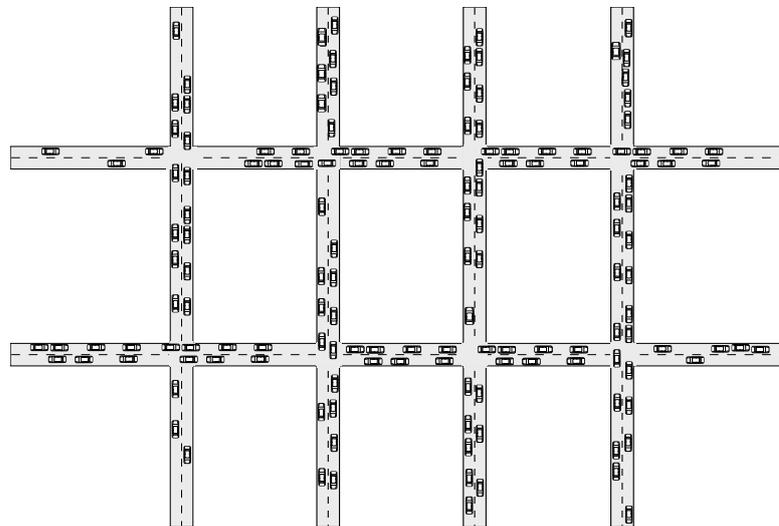


Figura 29: Cenário Avaliado.

O cenário de auto-estrada é uma pista de 1 quilômetro de comprimento com duas faixas nas duas direções. Quando um veículo chega ao final da pista, ele retorna pela direção contrária.

5.2 Simulador

O simulador utilizado nos experimentos foi o *Network Simulator* (ns-2). O ns-2 é um simulador de eventos discreto voltado especialmente para a pesquisa em redes de computadores (BERKELEY; ISI, 1998; ISSARIYAKUL; HOSSAIN, 2008; COUTINHO, 2003). Uma grande vantagem do ns-2 é que ele tem o código fonte aberto. Dessa forma, os usuários podem fazer os ajustes que quiserem para suas simulações particulares. Devido à sua natureza flexível e modular, o ns-2 ganhou popularidade constante desde seu nascimento em meados de 1989. Grandes instituições já contribuíram ou contribuem para o ns-2, tais como a *University of California*, *Cornell University*, *DARPA* (*Defense Advanced Research Projects Agency*), *NSF*

(*National Science Foundation*). Além disso, grupos de pesquisadores e desenvolvedores estão sempre trabalhando para deixar o ns-2 forte e versátil.

O ns-2 é composto, essencialmente, por duas linguagens: C++ e OTcl (*Object-oriented Tool command language*). C++ para a estrutura básica (protocolos, agentes etc) e OTcl para escrever os parâmetros da simulação (configuração de cenário, agendamento de eventos etc). A ligação entre o código C++ e as configurações de simulação do OTcl é feita pelo TclCL.

A arquitetura básica do ns-2 pode ser observada na figura 30.

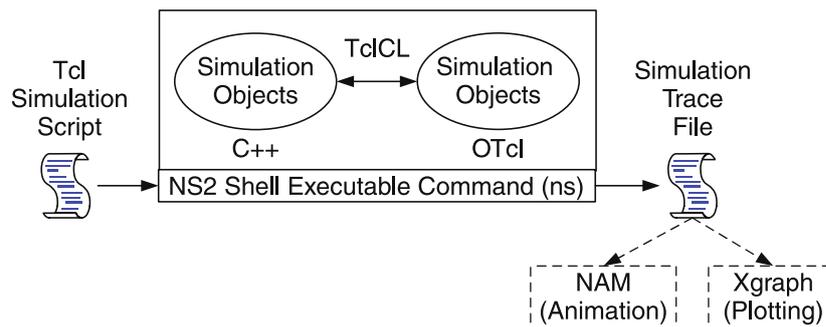


Figura 30: Arquitetura Básica do ns-2 (ISSARIYAKUL; HOSSAIN, 2008).

Depois que a simulação termina de ser executada, um arquivo de trace é criado. Nesse arquivo estão todas as informações da simulação em nível de redes de computadores. Nos arquivos de trace, estão registrados todos os pacotes que trafegaram pela rede durante a execução da simulação, mesmo os pacotes que foram descartados ou perdidos. A análise dos arquivos de trace nos permite avaliar o desempenho da rede, algum protocolo específico ou algum outro tópico de comunicação que o ns-2 possa oferecer.

No nosso trabalho, as simulações têm o objetivo de fornecer meios para a avaliação do protocolo MAV-AODV.

5.3 Detalhes dos Experimentos Realizados

No *Network Simulator* (ns-2.34) (BERKELEY; ISI, 1998), usamos módulos adicionais de camada de enlace e de camada física: *Mac802_11Ext* e *WirelessPhy-Ext*. Tais módulos possuem uma implementação mais cuidadosa das camadas MAC e PHY do padrão IEEE 802.11p. Esses módulos usaram, como base, os módulos originais do próprio ns-2 e aprimoraram para fornecerem um maior nível de precisão nos resultados da simulação (CHEN et al., 2007). O modelo de rádio propagação utilizado foi o *Free Space* para distâncias curtas e o *Two-Ray Ground* para distâncias maiores (STEPANOV; HERRSCHER; ROTHERMEL, 2005). O tráfego veicular foi gerado pelo *Vehicular Network Movement Generator* (NZOUONTA, 2008), que implementa os modelos de mobilidade propostos por Gipps (GIPPS, 1981, 1986). Nos modelos propostos por Gipps, os veículos trafegam em velocidade máxima, de modo que não

aconteça qualquer tipo de colisão. As velocidades variam de maneira a obedecer essa restrição. Os protocolos comparados foram: o MAV-AODV, o MAODV (ROYER; PERKINS, 1999) e o PUMA (VAISHAMPAYAN; GARCIA-LUNA-ACEVES, 2004). Além disso, todos os veículos são equipados com GPS (*Global Positioning System*). Segundo a literatura, pode-se ter boas amostras aleatórias com um tamanho de amostra maior ou igual a 30, o que caracteriza uma distribuição normal. Por isso, os experimentos foram executados 30 vezes para cada cenário.

A tabela 7 apresenta os principais parâmetros utilizados nos experimentos, como mostrado a seguir:

Parâmetros	Valores
Área do Cenário	1600m x 1500m
Alcance de Transmissão	350m, 500m
Modelo de Rádio Propagação	<i>Two-Ray Ground</i> e <i>Free Space</i> (distâncias curtas)
Modelo de Mobilidade	Gipps
Aplicação	CBR
Transporte	UDP
Camadas MAC e PHY	IEEE 802.11p
Tamanho do Pacote	256 bytes
Taxa de Transmissão	4 pacotes/segundo
Tamanho da Fila na Interface	50 pacotes
Tempo de Simulação	150 segundos
Número de Veículos	25, 50, 75 e 100
Protocolos de Roteamento	MAV-AODV, MAODV e PUMA
Número de Execuções de Cada Cenário	30 vezes
Nível de Confiança	95%

Tabela 7: Parâmetros Usados nos Experimentos.

5.4 Tráfego de Dados e Mensagens *Beacons*

O tráfego de dados usado nos experimentos é gerado pela aplicação CBR (*Constant Bit Rate*). Tal tráfego utiliza o UDP como protocolo de camada de transporte. Durante toda a simulação, apenas um nó transmite os dados *multicast* gerados pela aplicação CBR.

Para evitar que todos os nós receptores enviem requisições de rotas ao mesmo tempo e, assim, cada nó tenha um grande período de espera devido à contenção do meio sem fio e para se analisar mais claramente os *traces* da simulação, foi desenvolvido um escalonador de solicitações. A partir de 6 segundos da simulação, os nós receptores vão se juntando ao grupo *multicast*. O primeiro nó receptor tenta se juntar ao grupo *multicast* (faz uma solicitação para entrar no grupo) com 6 segundos de simulação e receber os dados da aplicação CBR. O segundo nó tenta se juntar com 7 segundos de simulação. O terceiro nó tenta se juntar com 8 segundos de

simulação. A fórmula 5.1 ilustra esse processo de os nós tentarem se juntar ao grupo *multicast*, como mostrado abaixo:

$$Time_i(join) = 5 + i \quad (5.1)$$

A fórmula 5.1 mostra que o i -ésimo nó tentará se juntar ao grupo *multicast* no instante $Time_i(join)$ da simulação. Assim, em um dado momento da simulação, todos os nós da rede, teoricamente, eram para estar pertencendo ao grupo *multicast* e recebendo os dados da aplicação CBR. Um pacote CBR teria que ser entregue para todos os membros do grupo *multicast*.

Além desse tráfego de pacotes de dados, um outro tráfego que usa o meio sem fio da rede é o tráfego de mensagens *beacons*. Todos os nós/veículos da rede utilizam a aplicação de mensagens *beacons*. Tais mensagens servem para manter os nós conscientes uns dos outros sobre posição, velocidade e direção dos veículos, de acordo com as informações de GPS. Essa aplicação transmite em *broadcast* de apenas um salto. Além disso, todo nó da rede envia uma mensagem *beacon* a cada x segundos. O valor de x é escolhido aleatoriamente dentro do intervalo $[0.5, 1.5]$.

5.5 Análise de Resultados

Cinco métricas foram utilizadas para analisar o desempenho do protocolo proposto neste trabalho: o atraso máximo fim-a-fim da árvore *multicast*, a variação do atraso fim-a-fim, o *overhead* de roteamento, a taxa de entrega de pacotes e a redundância de pacotes (todas essas métricas com seus valores médios).

5.5.1 Atraso Máximo Fim-a-Fim

A primeira métrica a ser analisada foi a média do atraso máximo fim-a-fim de uma estrutura *multicast*. Essa métrica representa o maior atraso da estrutura *multicast* durante toda a simulação. Tal métrica é um bom indicativo de quantas falhas acontecem nas estruturas *multicast*. Quanto mais falhas, maior o tempo para se recompor as rotas e maior o atraso máximo fim-a-fim.

As Figuras 31 e 32 mostram um comparativo entre os protocolos MAV-AODV, MAODV e PUMA usando essa métrica para os cenários Manhattan e auto-estrada com alcance de comunicação de 350 metros.

No cenário Manhattan, podemos perceber que o MAV-AODV teve desempenho melhor do que o MAODV com relação à essa métrica. Já no cenário de auto-estrada, o MAV-AODV teve desempenho semelhante ao do MAODV, exceto com 25 e 50 veículos, onde o MAV-AODV

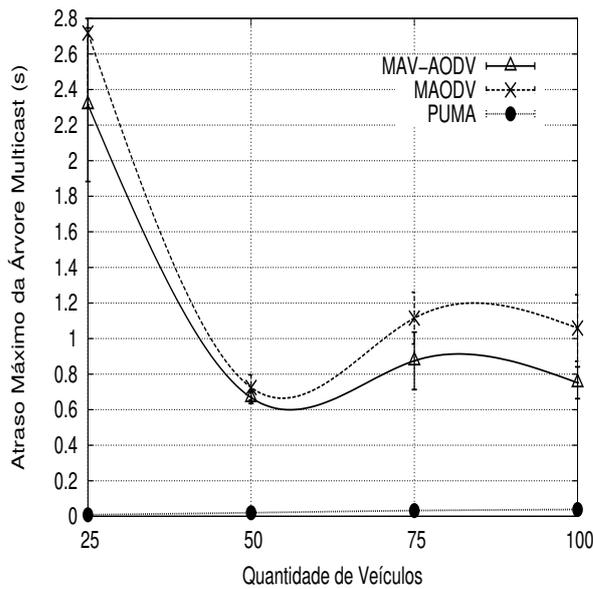


Figura 31: Atraso Máximo Fim-a-Fim (Manhattan)

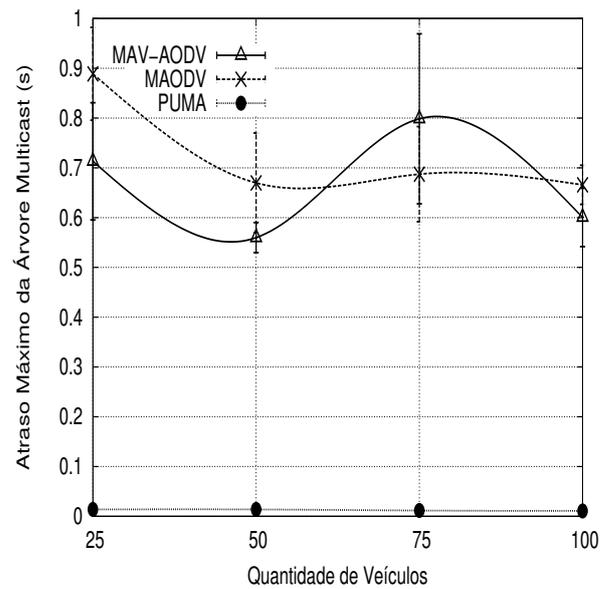


Figura 32: Atraso Máximo Fim-a-Fim (Auto-Estrada)

teve desempenho melhor. Isso significa que o MAV-AODV escolheu rotas mais duráveis do que o MAODV em cenários esparsos (Manhattan e auto-estrada com poucos veículos). Em um cenário esparsos existem poucas opções de rotas. Se uma rota falhar, pode não existir outra rota disponível no momento. Como o MAODV escolhe rotas menos duráveis, as rotas dele falharam antes que a do MAV-AODV. Com essa falha, o MAODV teve que esperar mais tempo para refazer o caminho e assim entregar o pacote *multicast* no destino final, aumentando assim o atraso máximo fim-a-fim.

Em cenários densos (auto-estrada com mais de 75 veículos), o MAV-AODV tem desempenho semelhante ao do MAODV. Isso porque os veículos estão muito juntos e há menos falhas de rotas.

Já comparado ao PUMA, o MAV-AODV teve desempenho bem pior. Isso porque o PUMA usa a estrutura em malha que é muito menos suscetível a falhas de rotas. A estrutura em malha permite que haja vários caminhos para o mesmo par origem-destino. O MAV-AODV possui caminhos únicos entre pares origem-destino. Assim, o MAV-AODV está bem mais sujeito à falhas. Quando há falhas, o MAV-AODV tem que esperar o restabelecimento de rotas para poder encaminhar o pacote *multicast* para o destino. Essa espera faz com que o atraso máximo fim-a-fim do MAV-AODV seja bem maior do que do PUMA.

5.5.2 Variação do Atraso Fim-a-Fim

Outra métrica utilizada foi a média da variação do atraso fim-a-fim de uma estrutura *multicast*. Essa métrica é definida como a diferença entre o atraso máximo e o atraso mínimo

fim-a-fim de uma estrutura *multicast*. Tal métrica é um bom indicativo sobre o tempo que os receptores levam para receber os pacotes *multicast*.

As Figuras 33 e 34 mostram um comparativo entre os protocolos MAV-AODV, MAODV e PUMA usando essa métrica para os cenários Manhattan e auto-estrada com alcance de comunicação de 350 metros.

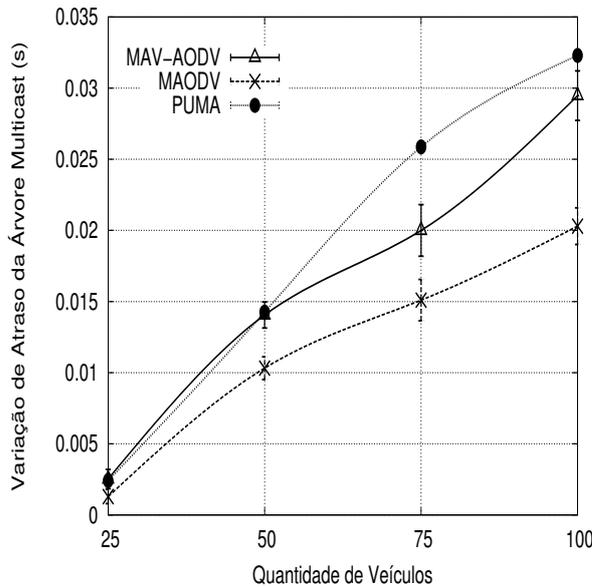


Figura 33: Variação do Atraso Fim-a-Fim (Manhattan)

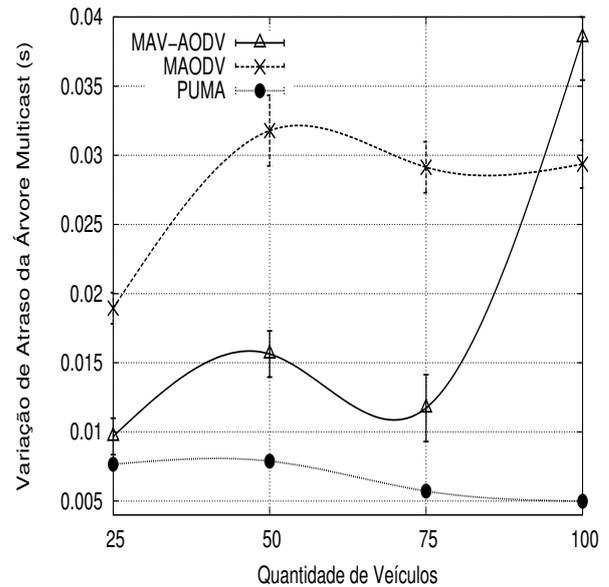


Figura 34: Variação do Atraso Fim-a-Fim (Auto-Estrada)

Nos dois cenários, a variação do atraso fim-a-fim foi muito pequena (no máximo 37 ms). No cenário Manhattan, percebe-se que a variação do atraso cresce de acordo com o crescimento do número de veículos na rede. Isso significa que quanto mais opções de rotas (cenário denso possui mais rotas diferentes), maior é a probabilidade de se escolher rotas diferentes e, conseqüentemente, uma maior variação do atraso.

No cenário de auto-estrada, os protocolos registram aumento e diminuição do atraso em função do aumento da quantidade de veículos. Isso se dá porque, além das falhas de rotas, outros fatores estão envolvidos, tais como: disputa por acesso ao meio sem fio e maior quantidade de nós receptores (40% do total de veículos da rede).

5.5.3 Overhead de Roteamento

Uma outra métrica avaliada foi o *overhead* de roteamento. Essa métrica é definida como o número de pacotes de controle dividido pelo número de pacotes de dados. Essa métrica revela o tráfego extra gerado pelos protocolos de roteamento para que os pacotes de dados sejam entregues com sucesso. O PUMA não foi incluído nessa avaliação porque, como a estrutura dele é em malha, existem muito mais pacotes de dados redundantes trafegando pela rede do que nos

outros protocolos. Isso diminuiria consideravelmente o *overhead* de roteamento do PUMA e tornaria a avaliação injusta.

As Figuras 35 e 36 mostram um comparativo entre os protocolos MAV-AODV e MAODV utilizando a métrica *overhead* de roteamento para os cenários Manhattan e auto-estrada com alcance de comunicação de 350 metros.

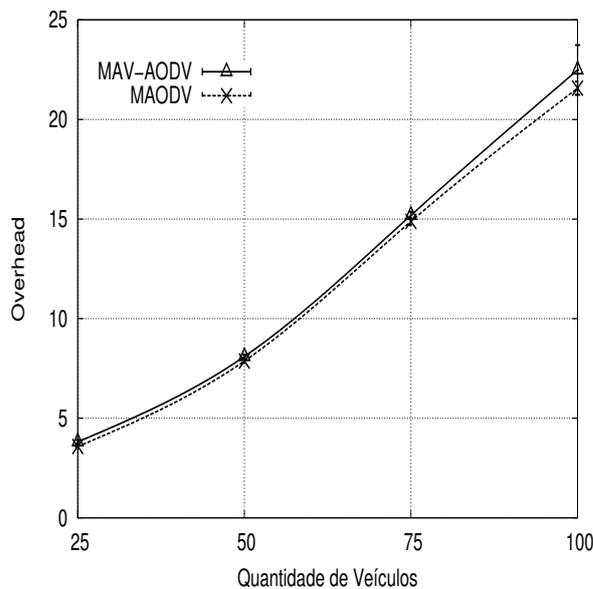


Figura 35: *Overhead* de Roteamento (Manhattan)

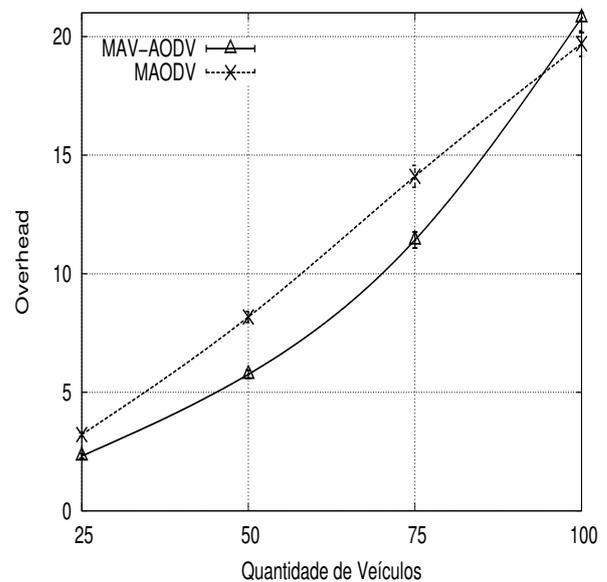


Figura 36: *Overhead* de Roteamento (Auto-Estrada)

No cenário Manhattan, podemos observar que MAV-AODV tem desempenho bastante parecido com o do MAODV (o MAV-AODV tem os mesmos tipos de mensagens de controle que o MAODV). Já no cenário de auto-estrada, o MAV-AODV gasta menos com mensagens de controle, exceto com 100 veículos no cenário. Como o cenário é mais denso e o MAODV possui uma maior frequência de falhas, a propagação das mensagens de recuperação de rotas são transmitidas por mais nós e o MAODV fica com um maior *overhead* de roteamento. Com um cenário muito denso (auto-estrada com 100 nós), há menos falhas de rotas e o *overhead* de roteamento dos dois protocolos são semelhantes.

5.5.4 Taxa de Entrega de Pacotes

A penúltima métrica avaliada foi a taxa de entrega de pacotes. Essa métrica serve para mostrar o grau de sucesso na entrega dos pacotes. Por exemplo, no cenário avaliado, se um nó transmite um pacote *multicast* para um grupo composto de 24 nós, todos os 24 membros do grupo *multicast* deveriam receber esse pacote. Se só 12 membros receberam, a taxa de entrega é $12/24 = 50\%$. Essa métrica indica a conectividade da estrutura *multicast*. Se a estrutura não estiver bem conectada, a taxa de entrega será pequena.

As Figuras 37 e 38 mostram um comparativo entre os protocolos MAV-AODV, MAODV e PUMA usando essa métrica para os cenários Manhattan e auto-estrada com alcance de comunicação de 350 metros.

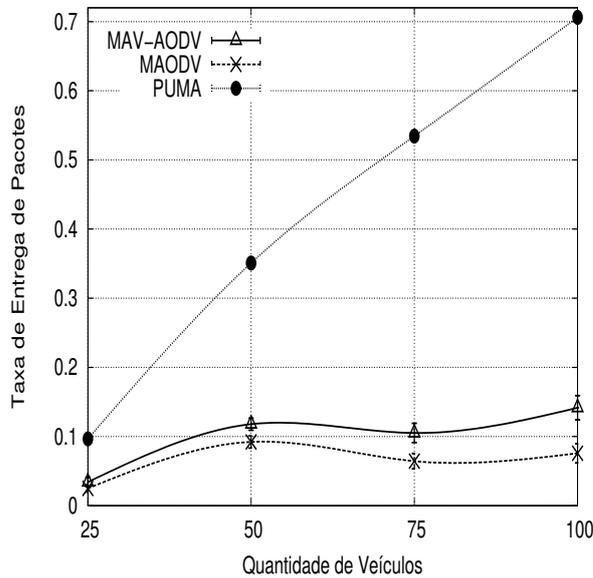


Figura 37: Taxa de Entrega de Pacotes (Manhattan)

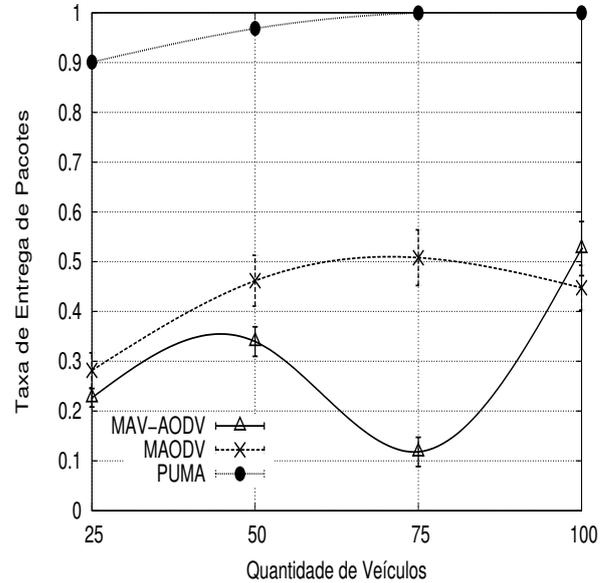


Figura 38: Taxa de Entrega de Pacotes (Auto-Estrada)

Como era de se esperar, nos dois cenários, o PUMA obteve resultados melhores em termos de taxa de entrega de pacotes. Isso se dá porque a estrutura do PUMA é em malha e existe uma probabilidade menor de haver falha na entrega de pacotes, já que existem várias rotas alternativas.

No cenário Manhattan, o MAV-AODV obteve desempenho melhor do que o MAODV. Isso significa que as árvores *multicast* construídas pelo protocolo MAV-AODV tem um maior grau de conectividade. Quanto mais tempo um nó fica conectado à árvore, mais pacotes ele irá receber.

No cenário de auto-estrada, o MAV-AODV teve um desempenho pior do que o MAODV. Isso se deu porque a previsão de mobilidade não funcionou bem nesse cenário. Um veículo que chegue no final da pista de 1km fará um retorno e mudará drasticamente sua direção. Esse comportamento não é bem avaliado por nosso modelo cinemático de previsão de tempo de vida de rotas.

5.5.5 Redundância de Pacotes

A última métrica analisada foi a redundância de pacotes. Essa métrica mostra a quantidade de vezes que um mesmo pacote foi transmitido no meio sem fio. Numa única simulação obtemos a média dessa métrica para todos os pacotes *multicast* que passaram pela rede.

As Figuras 39 e 40 mostram um comparativo entre os protocolos MAV-AODV, MAODV e PUMA usando essa métrica para os cenários Manhattan e auto-estrada com alcance de comunicação de 350 metros.

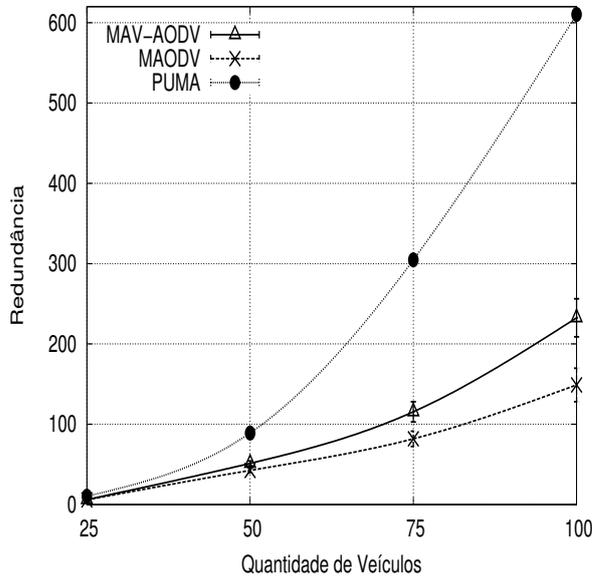


Figura 39: Redundância de Pacotes (Manhattan)

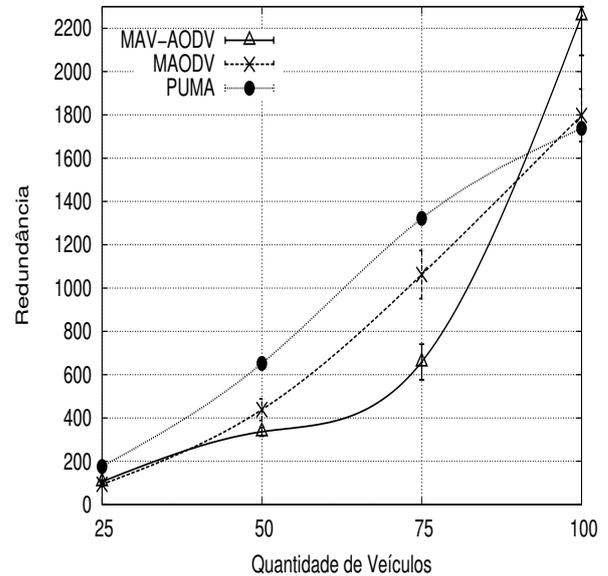


Figura 40: Redundância de Pacotes (Auto-Estrada)

No cenário Manhattan e na auto-estrada com até 75 veículos, como era de se esperar, o PUMA teve os maiores índices de redundância. Isso é devido ao fato de o PUMA usar a estrutura em malha. Essa estrutura utiliza vários caminhos alternativos para se transmitir os pacotes *multicast*.

No cenário Manhattan, o MAV-AODV teve uma redundância maior do que o MAODV. Isso se deu porque o MAV-AODV escolheu rotas um pouco maiores do que as do MAODV, mas que tinham um maior tempo de vida. Como quanto maior é a rota, maior é quantidade de retransmissões do mesmo pacote *multicast*, a redundância do MAV-AODV foi maior do que a do MAODV.

Já no cenário de auto-estrada com até 75 veículos, o MAV-AODV teve menos redundâncias do que o MAODV. Isso está associado à menor quantidade de entrega de pacotes devido à aplicação do modelo de mobilidade não adequado ao cenário proposto.

No cenário de auto-estrada e com 100 veículos, o MAV-AODV obteve redundância maior do que o MAODV. Isso também está relacionado com a maior entrega de pacotes. Com 100 veículos, o trânsito fica muito lento e o cenário proposto se adequa melhor à predição de mobilidade proposta pelo MAV-AODV. Nesse mesmo cenário, o PUMA obteve menos redundância. Isso se deu porque o PUMA aciona um mecanismo que identifica as redundâncias pelos identificadores dos pacotes e diminui as retransmissões dos pacotes. Esse mecanismo foi

acionado mais vezes devido à maior proximidade dos veículos entre si.

6 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, desenvolvemos e avaliamos um protocolo de roteamento *multicast* chamado MAV-AODV para redes *ad hoc* veiculares. O MAV-AODV utiliza princípios de colônia de formigas para tornar seu roteamento mais eficiente e estável em redes *ad hoc* veiculares. A árvore *multicast* construída pelo MAV-AODV utiliza caminhos com maiores tempos de vida e menores quantidades de saltos. Tais caminhos são escolhidos probabilisticamente, dependendo da quantidade de feromônio depositada pelas formigas (mensagens de roteamento).

Além disso, o MAV-AODV pode ser aplicado aos cenários veiculares porque leva em conta a mobilidade dos nós e, portanto, pode escolher rotas mais estáveis de acordo com o tempo de vida de cada rota. O tempo de vida de cada rota é calculado pelo tempo de vida de cada enlace da rota. A estimativa do tempo de vida de cada enlace é feita por um protocolo de mensagens *beacons*, que desenvolvemos e que atua juntamente com o MAV-AODV.

Ao analisar os resultados da simulação, o MAV-AODV obteve um desempenho melhor do que seu protocolo-base (MAODV) no cenário Manhattan. O MAV-AODV cumpriu o objetivo de manter a árvore *multicast* com um índice mais alto de conectividade. Pudemos observar isso ao verificar que a taxa de entrega de pacotes do MAV-AODV foi maior do que a do MAODV. Além disso, o atraso máximo da árvore do MAV-AODV também foi melhor do que o do MAODV. Além de manter, níveis de *overhead* de roteamento semelhantes aos do protocolo MAODV. Nesse mesmo cenário, como era de se esperar, o PUMA (protocolo que usa estrutura em malha) obteve desempenho melhor que o MAV-AODV e o MAODV (protocolos que usam a estrutura em árvore) em todas as métricas, exceto a métrica de redundância. Isso quer dizer que o PUMA desperdiçou recursos da rede ao transmitir várias vezes o mesmo pacote. Em ambientes veiculares densos e com grandes quantidades de dados trafegando pela rede, esse comportamento do PUMA pode causar problemas ao funcionamento da rede.

Dessa forma, o MAV-AODV tem suas principais contribuições listadas abaixo:

- Protocolo *multicast* tradicional (não *geocast*) aplicado às redes *ad hoc* veiculares;
- Cooperação com um protocolo de mensagens *beacons*, típico das redes *ad hoc* veiculares, para obter informações de predição de mobilidade;
- Utiliza a objetividade da transmissão em árvores, acrescentando robustez à essa estrutura e tornando o roteamento mais estável;
- Aplicação de princípios de colônias de formigas ao problema de roteamento *multicast* em redes *ad hoc* veiculares;

- Melhora no desempenho no cenário Manhattan, comparado ao protocolo MAODV, tendo uma maior taxa de entrega de pacotes e um menor atraso máximo de árvore.

Como trabalhos futuros, pretendemos melhorar a predição de mobilidade para cenários em que os veículos mudam drasticamente a direção da sua trajetória. Pretendemos também monitorar a qualidade dos enlaces da árvore *multicast* e predizer uma possível falha de enlace. A partir dessa predição, pretende-se já escolher um caminho alternativo, antes de o enlace quebrar de fato. Planejamos também comparar o MAV-AODV com outros protocolos de roteamento *multicast*. Além disso, um cenário de *highways* também poderia ser estudado. Um outro trabalho, seria tornar o envio, em *broadcast*, das mensagens RREQ-J mais eficiente.

REFERÊNCIAS

- ALVES, R.S. et al. Redes veiculares: Princípios, aplicações e desafios. *Minicursos do Simpósio Brasileiro de Redes de Computadores, SBRC*, 2009.
- BACHIR, A.; BENSLIMANE, A. A multicast protocol in ad hoc networks inter-vehicle geocast. In: IEEE. *Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual*. [S.l.], 2003. v. 4, p. 2456–2460.
- BADARNEH, O.S.; KADOCH, M. Multicast routing protocols in mobile ad hoc networks: a comparative survey and taxonomy. *EURASIP Journal on Wireless Communications and Networking*, Hindawi Publishing Corp., v. 2009, p. 26, 2009.
- BERKELEY, UC; ISI, USC. The network simulator ns-2. *Part of the VINT project. Available from <http://www.isi.edu/nsnam/ns>*, 1998.
- BIRADAR, R.; MANVI, S.; REDDY, M. Link stability based multicast routing scheme in manet. *Computer Networks*, Elsevier, v. 54, n. 7, p. 1183–1196, 2010.
- BISWAS, S.; TATCHIKOU, R.; DION, F. Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety. *Communications Magazine, IEEE*, IEEE, v. 44, n. 1, p. 74–82, 2006.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, ACM, v. 35, n. 3, p. 268–308, 2003.
- CHAKERES, I.; PERKINS, C. Dynamic manet on-demand (dymo) routing. *draft-ietf-manet-dymo-19.txt (work in progress)*, 2010.
- CHANG, B.J.; LIN, Y.M.; LIANG, Y.H. Distributed wireless links repair for maximizing reliability and utilization in multicast manet. In: IEEE. *22nd International Conference on Advanced Information Networking and Applications-Workshops*. [S.l.], 2008. p. 744–749.
- CHEN, Q. et al. Overhaul of ieee 802.11 modeling and simulation in ns-2. In: ACM. *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*. [S.l.], 2007. p. 159–168.
- CHEN, X.; ZHONG, Q.; LIU, D. An improved maodv protocol based on mobility prediction and self-pruning flooding. In: IEEE. *Communications and Mobile Computing, 2009. CMC'09. WRI International Conference on*. [S.l.], 2009. v. 2, p. 127–131.
- CHEN, Y.S.; LIN, Y.W.; LEE, S.L. A mobicast routing protocol in vehicular ad-hoc networks. *Mobile Networks and Applications*, Kluwer Academic Publishers, v. 15, n. 1, p. 20–35, 2010.
- COELHO, L.S.; NETO, R.F. Tavares. Colônia de formigas: Uma abordagem promissora para aplicações de atribuição quadrática e projeto de layout. *XXIV ENEGEP*, 2004.
- CORREIA, S.L.O.B.; CELESTINO, J.; CHERKAOUI, O. Mobility-aware ant colony optimization routing for vehicular ad hoc networks. In: IEEE. *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*. [S.l.], 2011. p. 1125–1130.

- COUTINHO, M.M. Network simulator–guia básico para iniciantes. *Universidade Federal do Pará (UFPA), Tech. Rep.*, 2003.
- DORIGO, M. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*, 1992.
- DORIGO, M.; BLUM, C. Ant colony optimization theory: A survey. *Theoretical computer science*, Elsevier, v. 344, n. 2-3, p. 243–278, 2005.
- DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, IEEE*, v. 26, n. 1, p. 29–41, 1996.
- DORIGO, M. et al. Positive feedback as a search strategy. Citeseer, 1991.
- FRACCHIA, R.; MEO, M.; ROSSI, D. Vanets: To beacon or not to beacon? *Proc. of the Autonet'06 at IEEE Globecom'06*, 2006.
- GIPPS, P.G. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, Elsevier, v. 15, n. 2, p. 105–111, 1981.
- GIPPS, P.G. A model for the structure of lane-changing decisions. *Transportation Research Part B: Methodological*, Elsevier, v. 20, n. 5, p. 403–414, 1986.
- GLOVER, F.; KOCHENBERGER, G.A. *Handbook of metaheuristics*. [S.l.]: Springer, 2003.
- GUNES, M.; SORGES, U.; BOUAZIZI, I. Ara-the ant-colony based routing algorithm for manets. In: IEEE. *Parallel Processing Workshops, 2002. Proceedings. International Conference on*. [S.l.], 2002. p. 79–85.
- HÄRRI, J.; BONNET, C.; FILALI, F. Kinetic mobility management applied to vehicular ad hoc network protocols. *Computer Communications*, Elsevier, v. 31, n. 12, p. 2907–2924, 2008.
- HARTENSTEIN, H.; LABERTEAUX, K.P. A tutorial survey on vehicular ad hoc networks. *Communications Magazine, IEEE, IEEE*, v. 46, n. 6, p. 164–171, 2008.
- HASSAN, A.; AHMED, M.H.; RAHMAN, MA. Performance evaluation for multicast transmissions in vanet. In: IEEE. *Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference on*. [S.l.], 2011. p. 001105–001108.
- HUANG, C.J. et al. Constructing alternative routes for vanet using machine learning techniques. 2007.
- HUANG, L.; HAN, H.; HOU, J. Multicast routing based on the ant system. *Applied Mathematical Sciences*, v. 1, n. 57, p. 2827–2838, 2007.
- HWANG, F.K.; RICHARDS, D.S. Steiner tree problems. *Networks*, Wiley Online Library, v. 22, n. 1, p. 55–89, 1992.
- IEEE. Ieee family of standards for wireless access in vehicular environments. In: *available from IEEE Standards*. [S.l.: s.n.], 2011.
- ISSARIYAKUL, T.; HOSSAIN, E. *Introduction to network simulator NS2*. [S.l.]: Springer Verlag, 2008.

- JIANG, D.; DELGROSSI, L. Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments. In: IEEE. *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*. [S.l.], 2008. p. 2036–2040.
- JIE, HU et al. Novel improvement based on stable path for maodv protocol. *Journal of Computer Applications*, v. 11, 2009.
- JOSHI, H.P.; SICHITIU, M.L.; KIHIL, M. Distributed robust geocast: a multicast routing for inter-vehicle communication. In: *WEIRD workshop on WiMax, wireless and mobility*. [S.l.: s.n.], 2007.
- JUNHAI, L.; LIU, X.; DANXIA, Y. Research on multicast routing protocols for mobile ad-hoc networks. *Computer Networks*, Elsevier, v. 52, n. 5, p. 988–997, 2008.
- KARIM, R. Vanet: Superior system for content distribution in vehicular network applications. *Rutgers University, Department of Computer Science, Tech. Rep*, 2008.
- KARLSON, P.; LÜSCHER, M. ‘pheromones’: a new term for a class of biologically active substances. Nature Publishing Group, 1959.
- KIHIL, M. et al. Reliable geographical multicast routing in vehicular ad-hoc networks. *Wired/Wireless Internet Communications*, Springer, p. 315–325, 2007.
- KIHIL, M.; SICHITIU, M.; JOSHI, H.P. Design and evaluation of two geocast protocols for vehicular ad-hoc networks. *Journal of Internet Engineering*, v. 2, n. 1, 2008.
- KO, Y.B.; VAIDYA, N.H. Geocasting in mobile ad hoc networks: Location-based multicast algorithms. In: IEEE. *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA’99. Second IEEE Workshop on*. [S.l.], 1999. p. 101–110.
- KUNZ, T. Multicasting in mobile ad-hoc networks: achieving high packet delivery ratios. In: IBM PRESS. *Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research*. [S.l.], 2003. p. 156–170.
- LEE, S.J.; GERLA, M.; CHIANG, C.C. On-demand multicast routing protocol. In: IEEE. *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*. [S.l.], 1999. p. 1298–1302.
- LI, Yunxin (Jeff). An overview of the dsrc/wave technology. In: *7th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. [S.l.: s.n.], 2010.
- LIN, Y.W.; CHEN, Y.S.; LEE, S.L. Routing protocols in vehicular ad hoc networks: A survey and future perspectives. *Journal of Information Science and Engineering*, Institute of Information Science Academia Sinica, v. 26, n. 3, p. 913–932, 2010.
- MAIA, J.E.B.; CELESTINO, J. Approximate analysis of a multiple geocast routing among geographical regions in vanet applications. In: IEEE. *Vehicular Networking Conference (VNC), 2009 IEEE*. [S.l.], 2009. p. 1–6.
- MANGAI, S.; TAMILARASI, A.; VENKATESH, C. Dynamic core multicast routing protocol implementation using ant colony optimization in ad hoc wireless networks. In: IEEE. *Computing, Communication and Networking, 2008. ICCCN 2008. International Conference on*. [S.l.], 2008. p. 1–5.

- MENOUAR, H.; LENARDI, M.; FILALI, F. Movement prediction-based routing (mopr) concept for position-based routing in vehicular networks. In: IEEE. *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*. [S.l.], 2007. p. 2101–2105.
- MOHAPATRA, P.; LI, J.; GUI, C. Multicasting in ad hoc networks. *Ad Hoc Networks*, Springer, p. 91–122, 2005.
- MUÑOZ, A.G. Multicast over vehicle ad hoc networks. *Universidad Carlos III de Madrid, Departamento de Ingeniería Telemática, Tech. Rep*, 2009.
- NAMBOODIRI, V.; GAO, L. Prediction-based routing for vehicular ad hoc networks. *Vehicular Technology, IEEE Transactions on*, IEEE, v. 56, n. 4, p. 2332–2345, 2007.
- NZOUONTA, Josiane. Invent vehicular traffic generator (ivtg) 1.0. *Disponível em: <http://web.njit.edu/borcea/invent/>*, 2008.
- PANICHPAPIBOON, S.; PATTARA-ATIKOM, W. Connectivity Requirements for Self-Organizing Traffic Information Systems. *IEEE Transactions on Vehicular Technology*, v. 57, n. 6, p. 3333–3340, 2008.
- PERKINS, C. et al. Rfc 3561-ad hoc on-demand distance vector (aodv) routing. *IETF, July*, 2003.
- RANGARAJAN, Dr. K. Baskaran J. Performance analysis of multicast protocols: Odmrp, puma and obamp. In: ISSN 0973-7391. *International Journal of Computer Science and Communication*. [S.l.], 2011. p. Volume–II, Number–II of September 2011.
- ROYER, E.M.; PERKINS, C.E. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In: CITESEER. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. [S.l.], 1999. p. 207–218.
- RUIZ, P.M.; GOMEZ-SKARMETA, A.F. Multicast ad hoc routing through mobility-aware steiner tree meshes with consistency across different mobility models. *International Journal of Parallel, Emergent and Distributed Systems*, Taylor & Francis, v. 21, n. 4, p. 257–277, 2006.
- SEBASTIAN, A. et al. A multicast routing scheme for efficient safety message dissemination in vanet. In: IEEE. *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. [S.l.], 2010. p. 1–6.
- SHANKAR, P. et al. CARS: Context-Aware Rate Selection for Vehicular Networks. In: *The sixteenth IEEE International Conference on Network Protocols (ICNP 2008)*. [S.l.: s.n.], 2008. p. 19–22.
- SHEN, C.C. et al. Ant-based distributed constrained steiner tree algorithm for jointly conserving energy and bounding delay in ad hoc multicast routing. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, ACM, v. 3, n. 1, p. 3, 2008.
- SOMMER, C.; DRESSLER, F. The dymo routing protocol in vanet scenarios. In: IEEE. *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*. [S.l.], 2007. p. 16–20.
- SOUZA, A. B. et al. An adaptive mechanism for access control in vanets. In: *ICN 2011, The Tenth International Conference on Networks*. [S.l.: s.n.], 2011.

- STEPANOV, I.; HERRSCHER, D.; ROTHERMEL, K. On the impact of radio propagation models on manet simulation results. In: *Proceedings of 7th IFIP International Conference on Mobile and Wireless Communications Networks (MWCN 2005)*. [S.l.: s.n.], 2005.
- TIRUCHENGODE, I. Multiple constraints for ant based multicast routing in mobile ad hoc networks. *Journal of Computer Science*, v. 5, n. 12, p. 1020–1027, 2009.
- TONGUZ, O. et al. Broadcasting in VANET. *2007 Mobile Networking for Vehicular Environments*, p. 7–12, 2007.
- TORRENT-MORENO, M. et al. Effects of a realistic channel model on packet forwarding in vehicular ad hoc networks. In: *IEEE. Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*. [S.l.], 2006. v. 1, p. 385–391.
- UZCATEGUI, R.; ACOSTA-MARUM, G. Wave: a tutorial. *Communications Magazine, IEEE*, IEEE, v. 47, n. 5, p. 126–133, 2009.
- VAISHAMPAYAN, R. Robust tree-based multicasting in ad hoc networks. In: *IEEE. Performance, Computing, and Communications, 2004 IEEE International Conference on*. [S.l.], 2004. p. 647–652.
- VAISHAMPAYAN, R.; GARCIA-LUNA-ACEVES, J.J. Efficient and robust multicast routing in mobile ad hoc networks. *Proceedings of MASS*, p. 304–313, 2004.
- VARSHNEY, U. Multicast over wireless networks. *Communications of the ACM, ACM*, v. 45, n. 12, p. 31–37, 2002.
- WANG, X.; YANG, Y.; AN, J. Multi-metric routing decisions in vanet. In: *IEEE. 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*. [S.l.], 2009. p. 551–556.
- WETHERALL, D. *OTcl-MIT Object Tcl: The FAQ and Manual*. [S.l.]: September, 1995.
- WISITPONGPHAN, N. et al. Broadcast storm mitigation techniques in vehicular ad hoc networks. *IEEE Wireless Communications*, IEEE, v. 14, n. 6, p. 84, 2007.
- YAN, Q.S.; WAN, X.Y.; FAN, Z.F. Improved routing protocol for maodv based on path stability. *Jisuanji Gongcheng/ Computer Engineering*, Shanghai Computer Society, v. 36, n. 20, 2010.
- YANG, S.; WU, J. New technologies of multicasting in manet. *Design and Analysis of Wireless Networks*, Citeseer, 2005.
- YOUSEFI, S.; ALTMAN, E.; EL-AZOUZI, R. Study of connectivity in vehicular ad hoc networks. In: *Workshop on Spatial Stochastic Models in Wireless Networks (SpasWin2007)*, Cyprus. [S.l.: s.n.], 2007. p. 573–587.
- YOUSEFI, S.; FATHY, M.; BENSLIMANE, A. Performance of beacon safety message dissemination in vehicular ad hoc networks (vanets). *Journal of Zhejiang University-Science A*, Springer, v. 8, n. 12, p. 1990–2004, 2007.
- YOUSEFI, S.; MOUSAVI, M.S.; FATHY, M. Vehicular ad hoc networks (VANETs): challenges and perspectives. In: *ITS Telecommunications Proceedings, 2006 6th International Conference on*. [S.l.: s.n.], 2006. p. 761–766.

ZHU, X.; LIAN, J. A qos multicast routing protocol with mobile prediction based on maodv in manets. In: IEEE. *Computer Science and Software Engineering, 2008 International Conference on*. [S.l.], 2008. v. 3, p. 355–358.