

PGCOMP - Programa de Pós-Graduação em Ciência da Computação  
Universidade Federal da Bahia (UFBA)  
Av. Adhemar de Barros, s/n - Ondina  
Salvador, BA, Brasil, 40170-110

<http://pgcomp.dcc.ufba.br>  
[pgcomp@ufba.br](mailto:pgcomp@ufba.br)

A detecção de vizinhança em redes veiculares ad hoc envolve a determinação de veículos vizinhos para uma organização colaborativa. Esse problema apresenta desafios em relação à comunicação entre veículos em redes veiculares devido à natureza dinâmica da rede, que provoca variações imprevisíveis na qualidade do canal de comunicação por conta de fatores, como condições ambientais, interferências e congestionamentos. Nesta dissertação, abordamos a detecção de vizinhança entre pelotões em redes dinâmicas, enfrentando desafios como a movimentação variada dos veículos e as interferências de edificações. A organização em pelotões representa um cenário relevante para o estudo em redes veiculares, uma vez que traz benefícios na qualidade do trânsito, como a redução do consumo de combustível, a melhoria do fluxo de tráfego e o aumento da segurança. Desenvolvemos um protocolo básico de comunicação, projetado para a comunicação entre líderes de pelotões, que garante o conhecimento mútuo entre eles, possibilitando a troca de informações essenciais para a detecção de vizinhança e o funcionamento colaborativo dos pelotões. Verificamos o protocolo utilizando o verificador de modelos PRISM. Utilizando este protocolo, realizamos experimentos para determinar os tempos necessários que ocorrem entre o envio e o recebimento de mensagens trocadas entre líderes de pelotões para a determinação de vizinhança. Realizamos experimentos em canais confiáveis e não confiáveis e determinamos tempos limite de progresso por meio de simulações utilizando o framework VEINS, que integra SUMO e OMNET++. Os resultados dos experimentos forneceram informações sobre os tempos limite de progresso em vários cenários que podem servir de referência para o desenvolvimento de aplicações relacionadas à comunicação em VANETS.

Palavras-chave: VANET, Detecção de Vizinhança, Tempo Limite de Progresso, Pelotão de Veículos, Protocolo de comunicação, Verificação de Modelos.

# Determinação de vizinhança em rede veiculares ad hoc: Uma análise dos tempos limite para entrega de mensagens

Cleber Brito Santos

Dissertação de Mestrado

Universidade Federal da Bahia

Programa de Pós-Graduação em  
Ciência da Computação

Setembro | 2023

MSC | 164 | 2023

Determinação de vizinhança em rede veiculares ad hoc: Uma análise dos tempos limite para entrega de mensagens

Cleber Brito Santos

UFBA





Universidade Federal da Bahia  
Instituto de Computação

Programa de Pós-Graduação em Ciência da Computação

**DETERMINAÇÃO DE VIZINHANÇA EM  
REDE VEICULARES AD HOC: UMA  
ANÁLISE DOS TEMPOS LIMITE PARA  
ENTREGA DE MENSAGENS**

Cleber Brito Santos

DISSERTAÇÃO DE MESTRADO

Salvador  
05 de setembro de 2023



CLEBER BRITO SANTOS

**DETERMINAÇÃO DE VIZINHANÇA EM REDE VEICULARES AD  
HOC: UMA ANÁLISE DOS TEMPOS LIMITE PARA ENTREGA  
DE MENSAGENS**

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientadora: Profa. Dra. Aline Maria Santos Andrade

Co-orientador: Prof. Dr. Alirio Santos de Sá

Salvador

05 de setembro de 2023

Ficha catalográfica elaborada pela Biblioteca Universitária de  
Ciências e Tecnologias Prof. Omar Catunda, SIBI – UFBA.

S237 Santos, Cleber Brito

Determinação de vizinhança em redes veiculares ad hoc:  
uma análise dos tempos limite para entrega de mensagens /  
Cleber Brito Santos. – Salvador, 2023.

103 f.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Aline Maria Santos  
Andrade. Co-orientador: Prof. Dr. Alirio Santos de  
Sá

Dissertação (Mestrado) – Universidade Federal da Bahia.  
Instituto de Computação, 2023.

1. Veículos - Redes. 2. Sistema de comunicação móvel. 3.  
Mnsagens. I. Andrade, Aline Maria Santos. II. Sá, Alirio Santos  
de. III. Universidade Federal da Bahia. IV. Título.

CDU 004.41



---

*“Determinação de vizinhança em rede veiculares ad hoc: Uma análise dos tempos limite para entrega de mensagens”*

Cleber Brito Santos

Dissertação apresentada ao Colegiado do Programa de Pós-Graduação em Ciência da Computação na Universidade Federal da Bahia, como requisito parcial para obtenção do Título de Mestre em Ciência da Computação.

**Banca Examinadora**

---

Prof. Dr.ª. Aline Maria Santos Andrade (Orientadora)

---

Prof. Dr. Allan Edgard Silva Freitas (IFBA)

---

Prof. Dr. Leobino Nascimento Sampaio (PGCOMP)

---



*Para o meu querido filho Daniel, a quem dedico esta dissertação de mestrado. Sua presença é a minha inspiração e motivação constante. Te amo além das palavras.*



## AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todos que contribuíram para a realização desta dissertação de mestrado. Os apoios, orientações e incentivos foram fundamentais para o meu crescimento acadêmico e pessoal ao longo dessa jornada desafiadora.

Primeiramente, gostaria de agradecer ao meu filho, Daniel. Você é minha maior inspiração e motivação. Sua presença constante em minha vida me impulsiona a ser melhor a cada dia. Obrigado por compreender os momentos em que precisei me dedicar aos estudos e por ser uma fonte inesgotável de amor e alegria.

A Dra. Norma Patrícia merece um agradecimento especial. Foi ela quem me introduziu à beleza da ciência e me deu o suporte necessário para dar os primeiros passos nessa área fascinante. Sua orientação e encorajamento foram essenciais para que eu adquirisse confiança em minhas capacidades e me comprometesse com a pesquisa acadêmica.

Gostaria de expressar minha sincera gratidão à minha grande amiga, Marcela. Durante toda a minha trajetória no mestrado, você esteve ao meu lado, apoiando-me incondicionalmente. Suas palavras de estímulo e seu ombro amigo foram um suporte vital em momentos de dificuldade. Sou grato por ter você como companheira nessa jornada acadêmica.

Um agradecimento especial também é direcionado a Vanessa. Desde o início até a minha qualificação, você foi uma presença fundamental. Suas valiosas dicas sobre como escrever corretamente e sua disposição em compartilhar seu conhecimento foram fundamentais para o desenvolvimento deste trabalho. Sou grato por ter contado com sua ajuda e amizade.

Gostaria de agradecer à Nilvana, que me apoiou em todo processo. Sua presença e incentivo foram importantes para que eu continuasse avançando, mesmo diante dos desafios que surgiram.

Por fim, gostaria de expressar minha gratidão aos professores Dra. Aline e Dr. Alirio. Suas orientações e paciência foram cruciais para o sucesso desta pesquisa. Agradeço por compartilharem seu conhecimento, dedicando seu tempo e esforço para me guiar no caminho da excelência acadêmica.

A todos os mencionados e a todos aqueles que de alguma forma contribuíram para esta conquista, meu mais profundo agradecimento. Vocês fizeram parte dessa jornada e sou grato por terem acreditado em mim.



*Não importa o quão devagar você vá, desde que você não pare.*

—CONFÚCIO



## RESUMO

A detecção de vizinhança em redes veiculares ad hoc envolve a determinação de veículos vizinhos para uma organização colaborativa. Esse problema apresenta desafios em relação à comunicação entre veículos em redes veiculares devido à natureza dinâmica da rede, que provoca variações imprevisíveis na qualidade do canal de comunicação por conta de fatores, como condições ambientais, interferências e congestionamentos. Nesta dissertação, abordamos a detecção de vizinhança entre pelotões em redes dinâmicas, enfrentando desafios como a movimentação variada dos veículos e as interferências de edificações. A organização em pelotões representa um cenário relevante para o estudo em redes veiculares, uma vez que traz benefícios na qualidade do trânsito, como a redução do consumo de combustível, a melhoria do fluxo de tráfego e o aumento da segurança. Desenvolvemos um protocolo básico de comunicação, projetado para a comunicação entre líderes de pelotões, que garante o conhecimento mútuo entre eles, possibilitando a troca de informações essenciais para a detecção de vizinhança e o funcionamento colaborativo dos pelotões. Verificamos o protocolo utilizando o verificador de modelos PRISM. Utilizando este protocolo, realizamos experimentos para determinar os tempos necessários que ocorrem entre o envio e o recebimento de mensagens trocadas entre líderes de pelotões para a determinação de vizinhança. Realizamos experimentos em canais confiáveis e não confiáveis e determinamos tempos limite de progresso por meio de simulações utilizando o framework VEINS, que integra SUMO e OMNET++. Os resultados dos experimentos forneceram informações sobre os tempos limite de progresso em vários cenários que podem servir de referência para o desenvolvimento de aplicações relacionadas à comunicação em VANETs.

**Palavras-chave:** VANET, Detecção de Vizinhança, Tempo Limite de Progresso, Pelotão de Veículos, Protocolo de comunicação, Verificação de Modelos



## ABSTRACT

Neighborhood detection in ad hoc vehicular networks involves determining neighboring vehicles for collaborative organization. This problem presents challenges to communication between vehicles in vehicular networks due to the dynamic nature of the network, which leads to unpredictable variations in the quality of the communication channel due to factors such as environmental conditions, interferences, and congestion. In this dissertation, we address the neighborhood detection between platoons in dynamic networks, facing challenges such as varied vehicle movements and building interferences. The platooning organization represents a relevant scenario for studying vehicular networks, as it brings benefits to traffic quality, such as fuel consumption reduction, traffic flow improvement, and increased safety. We have developed a basic communication protocol designed for communication between platoon leaders, which ensures mutual knowledge between them, enabling the exchange of essential information for neighborhood detection and collaborative operation of the platoons. We have verified the protocol using the PRISM model checker. Using this protocol, we conducted experiments to determine the required times that occur between the sending and receiving of messages exchanged between platoon leaders for neighborhood determination. We performed experiments on both reliable and unreliable channels and determined progress time bound through simulations using the VEINS framework, which integrates SUMO and OMNET++. The results of the experiments provided information about the progress time bound in various scenarios that can serve as a reference for the development of applications related to communication in VANETs.

**Keywords:** VANET, Neighborhood Detection, Progress Time Bound, Platoon, Communication Protocol, Model Checking.



# SUMÁRIO

<b>Capítulo 1—Introdução</b>	1
1.1 Trabalhos relacionados . . . . .	3
1.2 Contribuições . . . . .	4
1.3 Escopo da dissertação . . . . .	4
1.4 Publicações . . . . .	5
1.5 Estrutura da dissertação . . . . .	5
<b>Capítulo 2—Redes veiculares ad hoc</b>	7
2.1 Tipos de sistema de comunicação aplicados às redes veiculares . . . . .	7
2.2 Características e desafios das VANETs . . . . .	9
2.3 Comunicação em grupo aplicada às VANETs . . . . .	12
2.4 Pelotões . . . . .	14
2.5 Padrão IEEE 802.11p . . . . .	15
2.5.1 Unidade de dados de protocolo da camada de controle de acesso ao meio . . . . .	16
2.6 Camada MAC abstrata . . . . .	17
2.7 Detecção de vizinhança em uma rede veicular ad hoc . . . . .	19
2.7.1 Alcance de transmissão . . . . .	20
2.7.2 Tamanho das mensagens . . . . .	20
2.7.3 Periodicidade no envio de mensagens . . . . .	21
2.7.4 Quantidade de nós que utilizam o canal de comunicação . . . . .	21
2.7.5 Estudos dos prazos necessários para detecção de vizinhança nas redes veiculares ad hoc . . . . .	22
2.8 Conclusão . . . . .	23
<b>Capítulo 3—Verificação de modelos com o PRISM</b>	25
3.1 Linguagem de modelagem PRISM . . . . .	26
3.1.1 Especificação de propriedade . . . . .	27
3.1.2 Contraexemplos e testemunhas . . . . .	28
3.1.3 Método de verificação . . . . .	29
3.2 Conclusão . . . . .	30

<b>Capítulo 4—Protocolo Básico para Comunicação entre Pelotões</b>	<b>31</b>
4.1 Camada de rede . . . . .	31
4.2 Camada MAC abstrata . . . . .	32
4.3 Sistema de Pelotão . . . . .	32
4.4 Protocolo Básico para Comunicação entre Pelotões . . . . .	34
4.4.1 Descrição do Algoritmo . . . . .	34
4.5 Sistema de gerenciamento multipelotões . . . . .	36
4.6 Método de detecção de vizinhança utilizado no PBCP . . . . .	37
4.7 Verificação do PBCP . . . . .	40
4.8 Verificação do Modelo em PRISM . . . . .	42
4.9 Conclusão . . . . .	43
<b>Capítulo 5—Análise de tempos limite de progresso para detecção de vizinhança</b>	<b>45</b>
5.1 Análise dos tempos limite de progresso para determinação de vizinhança	46
5.1.1 Experimentos utilizando o PBCP . . . . .	48
5.1.2 Cenário 1 - Determinação dos tempos limite de progresso em função da quantidade de líderes de pelotão e tamanho da mensagem em uma VANET. . . . .	49
5.1.3 Cenário 2 - Determinação dos tempos limite de progresso em função do período de envio. . . . .	50
5.1.4 Cenário 3 - Determinação das vizinhanças detectadas em um canal de comunicação não confiável variando o tempo de espera com o período fixado. . . . .	50
5.1.5 Cenário 4 - Determinação das vizinhanças detectadas em um canal de comunicação não confiável variando o tempo de espera com o período fixado através de uma pista reta. . . . .	52
5.2 Conclusão . . . . .	54
<b>Capítulo 6—Conclusão e trabalhos futuros</b>	<b>55</b>
6.1 Limitações . . . . .	56
6.2 Trabalhos futuros . . . . .	56
<b>Referências Bibliográficas</b>	<b>57</b>
<b>Apêndice A—Simuladores</b>	<b>63</b>
A.1 SUMO . . . . .	63
A.2 OMNET++ . . . . .	64
A.3 VEINS . . . . .	65
<b>Apêndice B—Noções sobre verificação de modelos</b>	<b>67</b>
B.1 Lógica de árvore de computação . . . . .	67

B.2	Lógica de árvore de computação probabilística . . . . .	70
<b>Apêndice C—Modelagem do PBCP em PRISM</b>		<b>73</b>
C.1	Modelagem do protocolo em PRISM . . . . .	73



## LISTA DE FIGURAS

1.1	Ilustração de vizinhança estável . . . . .	4
2.1	Tipos de transmissões . . . . .	8
2.2	Sistemas de comunicação das redes veiculares (CUNHA et al., 2016) . . .	10
2.3	Tipos de compartilhamento . . . . .	10
2.4	Típico cenário de uma VANET (SANGUESA et al., 2016) . . . . .	11
2.5	Ilustração de uma comunicação entre grupos através dos líderes . . . . .	13
2.6	Ilustração do pelotão de veículos . . . . .	14
2.7	Alocação de canal IEEE 802.11p (ALVES et al., 2009) . . . . .	15
2.8	Arquitetura do sistema de rede <i>ad hoc</i> utilizando a camada MAC abstrata	18
3.1	A linguagem PRISM: Exemplo de um jogo de moeda eletrônica . . . . .	26
3.2	Especificando propriedade em PRISM . . . . .	28
3.3	Estrutura do PRISM. . . . .	29
3.4	Um <i>screenshot</i> da interface gráfica do PRISM . . . . .	30
4.1	Arquitetura do sistema de comunicação em camadas . . . . .	32
4.2	Estrutura do relacionamento entre o sistema de pelotão e o sistema de gerenciamento multipelotões. . . . .	36
4.3	Detecção de vizinhança - Canal de comunicação sem perda de mensagens	39
4.4	Detecção de vizinhança - Canal de comunicação com perda de mensagens	40
4.5	Tipos de vizinhança . . . . .	40
5.1	Variação do tempo limite de progresso em relação à quantidade de pelotões.	50
5.2	Variação do tempo limite de progresso em relação valores dos períodos do envio de mensagens. . . . .	51
5.3	Percepção de vizinhança estável em diferentes tempos de espera em um canal de comunicação não confiável. . . . .	51
5.4	Percepção de vizinhança estável em diferentes tempos de espera em um canal de comunicação não confiável. . . . .	52
5.5	Percepção de vizinhança estável em diferentes tempos de espera em um canal de comunicação não confiável. . . . .	53
5.6	Percepção de vizinhança estável em diferentes tempos de espera em um canal de comunicação não confiável. . . . .	54
A.1	Simulador SUMO . . . . .	63
A.2	Arquitetura do VEINS e conexões com o OMNET++ e SUMO . . . . .	65

B.1	Estrutura de Kripke e árvore de computação infinita . . . . .	69
B.2	Operadores temporais da lógica CTL. . . . .	70
B.3	Exemplo de uma DTMC. . . . .	71

## LISTA DE TABELAS

4.1	Propriedades verificadas do PBCP . . . . .	43
5.1	Requisitos de tempo relacionado ao envio de mensagem . . . . .	47
5.2	Parâmetros de configuração de rede . . . . .	47
5.3	Parâmetros de configuração de mobilidade . . . . .	48
5.4	Restrições de tempo relacionados à periodicidade no envio de mensagens e o tempo de espera . . . . .	49



## LISTA DE SIGLAS

<b>CCH</b>	Control CHannel
<b>CSMA/CA</b>	Carrier Sense Multiple Access with Collision Avoidance
<b>DTMC</b>	Discrete Time Markov Chains
<b>DSRC</b>	Dedicated Short Range Communications
<b>FIFO</b>	First In First Out
<b>GNSS</b>	Global Navigation Satellite System
<b>GPS</b>	Global Positioning System
<b>HVC</b>	Hybrid Vehicular Communication
<b>IVC</b>	Inter-Vehicle Communication
<b>MAC</b>	Media Access Control
<b>MANET</b>	Mobile Ad Hoc Networks
<b>MSDU</b>	MAC Service Data Unit
<b>MIVC</b>	Multi-Inter-Vehicle Communication
<b>MPDU</b>	MAC Protocol Data Unit
<b>NED</b>	Network Description
<b>OMNET++</b>	Objective Modular Network Testbed in C++
<b>PBCP</b>	Protocolo Básico para Comunicação entre Pelotões
<b>PRISM</b>	PRobabilistic Symbolic Model checker
<b>RVC</b>	Roadside-to-Vehicle Communication
<b>SCH</b>	Service CHannel
<b>SIVC</b>	Simple-Inter-Vehicle Communication
<b>TRACI</b>	Traffic Control Interface
<b>VEINS</b>	VEhIcular Network Simulation
<b>WAVE</b>	Wireless Access in the Vehicular Environment



## INTRODUÇÃO

A questão de segurança no trânsito em grandes centros urbanos é um problema que motiva a pesquisa por tecnologias e soluções computacionais que minimizem os riscos decorrentes deste ambiente. Mais recentemente, a utilização de sensores e unidades de controle eletrônico têm tornado os veículos mais inteligentes e basicamente mais seguros para dirigir (DIANGE et al., 2018). Como tendência para o futuro, considerando as inovações alcançadas no campo das tecnologias de rede e, particularmente, das comunicações móveis sem fio, tecnologias habilitadoras têm sido propostas a fim de possibilitar a comunicação entre veículos, permitindo a detecção de vizinhança e a troca de informações sobre condições de tráfego, obstáculos e alertas de segurança, entre outros (AMEEN et al., 2020). Essas transformações terão um impacto significativo na forma como as pessoas dirigirão no futuro e como os sistemas de transporte serão utilizados.

As comunicações veiculares por meio de redes dedicadas vão além de simples transferências de dados básicos, oferecendo oportunidades significativas para aprimorar a segurança no trânsito. As aplicações e vantagens potenciais dessas comunicações são diversas e abrangem áreas como a coordenação de movimentos entre veículos para otimizar o fluxo de tráfego e reduzir os tempos de viagem, além de melhorar a eficiência energética e diminuir as emissões de gases poluentes por meio de estratégias de condução cooperativa (BARIAH et al., 2015). O interesse nesse campo tem crescido consideravelmente nos últimos anos, recebendo uma atenção notável da comunidade científica (LEE; ATKINSON, 2021; AMEEN et al., 2020). Uma subcategoria dessas redes é conhecida como Redes Veiculares *Ad Hoc* (VANET, Vehicular Ad hoc NETWORK), compostas por um conjunto de dispositivos móveis (veículos) que se conectam automaticamente de forma arbitrária, operando sem uma administração de rede hierárquica estrita (QURESHI; ABDULLAH, 2013).

Um tipo de organização que utiliza VANETs é a formação de veículos em pelotão, o que oferece diversas vantagens. Primeiramente, há o potencial de redução do consumo de combustível e da poluição ambiental. Em segundo lugar, essa abordagem melhora o fluxo de tráfego e reduz os congestionamentos nas rodovias. Em terceiro lugar, a formação

em pelotão aumenta a segurança no trânsito, uma vez que a probabilidade de acidentes devido a erro humano é reduzida.

Na comunicação entre veículos em VANETs, a detecção de vizinhança é essencial. Isso permite que os veículos se organizem em grupos e ajam de forma colaborativa. A detecção de vizinhança envolve desafios principalmente devido à natureza dinâmica da rede de veículos, que apresenta variações imprevisíveis na qualidade do canal de comunicação devido a fatores como condições ambientais, interferências e congestionamentos, além de interferências no enlace de comunicação (RAJESH; GNANASEKAR, 2016; AHMED; GHARAVI, 2018). Um desses desafios é a determinação de tempos de entrega de mensagens.

A análise dos tempos para entrega de mensagens torna-se complexa pois requer suposições de comunicação relacionadas a detalhes de baixo nível, como por exemplo: se as transmissões simultâneas causam colisões; se colisões podem ser detectadas; se problemas relacionados à interferência na rede podem influenciar na comunicação, entre outros. Uma alternativa para tratar estas questões tem sido a utilização de uma camada de acesso ao meio abstrata situada entre a camada física utilizada para o encaminhamento das mensagens e a aplicação utilizada nos veículos. Esta foi proposta inicialmente por Kuhn, Lynch e Newport (2009). *A priori* o uso desse serviço permite que projetos e análises de algoritmos que compartilhem informações entre processos em uma rede de computadores não necessitem lidar com problemas de baixo nível, tratando-os como limites de atrasos abstratos. Assim, neste trabalho, abstraímos os detalhes de baixo nível, e consideramos os atrasos das entregas de mensagens como limites abstratos de tempo, chamados de tempos limite de progresso (CORNEJO; VIQAR; WELCH, 2014).

O objetivo desta dissertação é analisar e determinar os tempos de entrega de mensagens durante o processo de detecção de vizinhança entre pelotões de veículos, considerando as variações nas condições de comunicação. Essas variações incluem fatores como o tamanho das mensagens trocadas, a dimensão dos pelotões, interferências causadas por edificações e variações nos intervalos de troca de mensagens entre os pelotões. Esses tempos representam prazos essenciais para que as mensagens alcancem os vizinhos de forma adequada e confiável, permitindo a detecção da vizinhança de pelotões. Os resultados podem servir como uma referência para futuros projetos que busquem garantir a comunicação em VANETs, a exemplo da gestão entre múltiplos pelotões.

Desenvolvemos o protocolo PBCP (Protocolo Básico para Comunicação entre Pelotões) para a comunicação entre pelotões de veículos, utilizado em simulações para determinar os tempos de entrega de mensagens. O PBCP foi projetado com base em uma camada abstrata de acesso ao meio, utilizando operações básicas de envio e recebimento de mensagens. Esse protocolo foi desenvolvido para atender requisitos da detecção de vizinhança em VANETs, buscando garantir uma comunicação eficaz e segura entre os pelotões.

As simulações realizadas utilizaram o *framework* VEINS, *VEhIcular Net-work Simulation* (SOMMER et al., 2019). O VEINS integra o SUMO (*Simulation of Urban MObility*) com o OMNET++ (*Objective Modular Network Testbed in C++*) (VARGA; HORNIG, 2010). Os detalhes completos dos simuladores estão disponíveis no Apêndice A. O SUMO é usado para simular a mobilidade dos veículos, enquanto que o OMNET++ é usado para

simular a comunicação entre veículos.

## 1.1 TRABALHOS RELACIONADOS

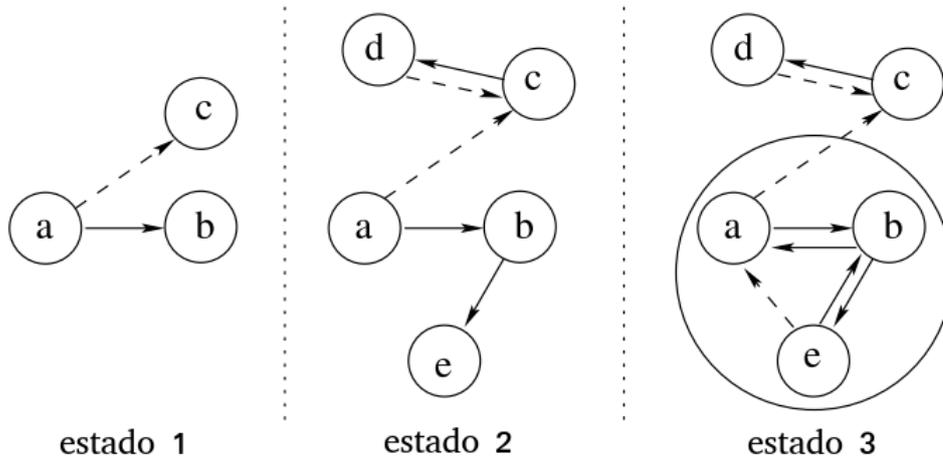
A determinação de tempos limite na entrega de mensagens em redes sem fio e que não possuem uma infraestrutura definida, tais como as redes veiculares *ad hoc*, são tratados em diferentes trabalhos na literatura. Muitos destes utilizam uma abstração da camada de acesso ao meio de comunicação para determinação desses limites (YU et al., 2021).

Yu et al. (2021), por exemplo, analisam o problema da comunicação em redes dinâmicas. É considerado que a rede sofre com a interferência do sinal sem fio durante o processo de comunicação em função das transmissões simultâneas geradas pelos nós. A dinamicidade é analisada sob o ponto de vista dos nós que entram e saem da vizinhança, bem como pela mobilidade dos nós, causando falhas ou participação intermitente dos nós na rede. São considerados neste estudo os tempos limite de progresso durante as trocas de mensagens. As abordagens apresentadas neste trabalho são aplicáveis aos cenários das VANETs, principalmente pela característica dinâmica dos nós. Assim como o trabalho de Yu et al. (2021) consideramos a dinamicidade da rede para determinar os tempos limite de progresso estendendo a um cenário com múltiplos pelotões de veículos.

O comportamento dinâmico dos pelotões faz com que a comunicação veicular não seja trivial. Durante o processo de comunicação entre pelotões é necessário que haja alguma estabilidade de maneira a garantir que aplicações (e.g. detecção da vizinhança) possam ser processadas. No trabalho de Arantes et al. (2010), por exemplo, é considerado que um conjunto de processos possuem uma vizinhança estável quando estes não travam nem saem da partição (grupo) e novos membros não são aceitos. A vizinhança estável é composta por um conjunto de processos vizinhos que sempre podem se comunicar uns com os outros por meio de caminhos dinâmicos, dentro de um tempo limite. Assim, se um nó  $q$  é alcançável a partir de um nó vizinho  $p$ , e vice-versa, então  $q$  e  $p$  fazem parte de uma vizinhança estável. A Figura 1.1 ilustra a definição de uma vizinhança estável. Todos os nós estão corretos (no sentido de que não falham durante a execução) e o gráfico evolui do estado 1 para o estado 3. Setas contínuas correspondem a caminhos dinâmicos em que a comunicação é realizada dentro de um período de tempo. Antes do estado 3 nenhum dos nós possui uma vizinhança estável, ou seja, não há garantia de que os processos possam enviar mensagens a tempo uns para os outros. No estado 3, após a troca de mensagens, como sempre existem caminhos entre os nós  $a$ ,  $b$  e  $e$ , estes nós formam uma vizinhança estável.

Assim como no trabalho de Arantes et al. (2010), consideramos o conceito de vizinhança estável durante o processo de detecção de pelotões vizinhos. Em nosso trabalho, a vizinhança estável ocorre quando dois pelotões vizinhos possuem informações atualizadas um do outro em um intervalo de tempo utilizado para que haja a detecção da vizinhança. Desta forma, os pelotões podem tomar decisões de forma descentralizada e mais confiáveis.

Alguns trabalhos que lidam com o desenvolvimento de abordagens voltadas ao gerenciamento das operações de unificação e subdivisão de pelotões não consideram o problema de entregar informações atualizadas e corretas de pelotões vizinhos (AMOOZADEH et al.,



**Figura 1.1** Ilustração de vizinhança estável  
**Fonte:** Adaptado de Arantes et al. (2010)

2015; JIA et al., 2016). Assim, não dando garantias de que os módulos de gerenciamento multipelotão, instalados em cada pelotão, possam tomar decisões de forma descentralizada e confiáveis. Em nossos experimentos, no entanto, levamos em consideração que as variações de mobilidade entre os pelotões podem provocar perdas de mensagens, diferentes tempos limite nas entregas das mensagens e alterações de conectividade mais frequentes, que podem fazer com que as informações não sejam atualizadas ou corretas.

## 1.2 CONTRIBUIÇÕES

A principal contribuição desta dissertação está relacionada aos resultados dos experimentos referentes à detecção de vizinhança em uma rede veicular *ad hoc* composta por pelotões de veículos. Nesses experimentos, determinamos os tempos limite para a entrega de mensagens, considerando cenários de comunicação relevantes. Isso fornece uma base para que aplicações que possuam restrições de tempos para entrega de mensagens sejam desenvolvidas com uma segurança na efetividade da comunicação.

## 1.3 ESCOPO DA DISSERTAÇÃO

A dissertação trata da detecção de vizinhança entre múltiplos pelotões em formação de fila. Assume-se que as informações de cada pelotão, como a quantidade de veículos e suas identificações, são fornecidas por meio de uma solução dedicada à criação dos grupos de veículos em formação de pelotão. O desenvolvimento dessa solução não faz parte do escopo deste trabalho.

Nos pelotões, à medida que o número de veículos aumenta consideravelmente, a coordenação e a sincronização das ações dos veículos podem ser impactadas, como indicado por Peng et al. (2017). Isso cria um desafio para manter uma formação coesa e garantir que todos os veículos sigam as mesmas orientações. A capacidade de controlar a formação dos veículos no pelotão pode ser comprometida, levando à necessidade de subdividir o

pelotão em unidades menores. Da mesma forma, é possível unificar pelotões quando o número de veículos diminui significativamente. As manobras de junção e separação de pelotões não são tratadas neste trabalho.

## 1.4 PUBLICAÇÕES

SANTOS, Cleber Brito; ANDRADE, Aline Maria; SÁ, Alirio Santos de. **Uma análise dos tempos limite para entrega de mensagens e para determinação de vizinhança em rede veiculares ad hoc.** In: *In Anais do VII Workshop de Computação Urbana, maio 22, 2023, Brasília/DF, Brasil. SBC, Porto Alegre, Brasil, 93-106.* Disponível em (<https://doi.org/10.5753/courb.2023.799>).

## 1.5 ESTRUTURA DA DISSERTAÇÃO

Este trabalho está organizado da seguinte maneira:

- No Capítulo 2, são abordados aspectos gerais a respeito das redes veiculares *ad hoc*, envolvendo sua característica dinâmica, uma explicação sobre o padrão de comunicação para redes veiculares *ad hoc*, as vantagens da formação em pelotão de veículos, uma abstração da camada de acesso ao meio utilizada para troca de mensagens em redes veiculares e, por fim, os conceitos fundamentais relacionados à detecção de vizinhança nas VANETs.
- No Capítulo 3, realizamos a verificação de propriedades dos tipos *liveness* e *safety* no protocolo, relacionadas à capacidade dos veículos de se detectarem uns aos outros como vizinhos em canais confiáveis e não confiáveis, utilizando um verificador de modelos probabilístico.
- No Capítulo 4, apresentamos o Protocolo Básico para Comunicação entre Pelotões utilizado neste trabalho para a realização dos experimentos de determinação dos tempos limite de progresso para entrega de mensagens.
- No Capítulo 5, são apresentados os resultados dos experimentos realizados para determinar os tempos limite de progresso necessários na detecção de veículos vizinhos, bem como o desempenho associado à capacidade de detectar a vizinhança em diferentes cenários. Essas análises foram realizadas considerando tanto canais de comunicação confiáveis quanto não confiáveis.
- No Capítulo 6, são apresentadas as considerações finais do trabalho e indicados trabalhos futuros da pesquisa.
- Por fim, no Apêndice A, são apresentados os simuladores utilizados para a realização dos experimentos. No Apêndice B, é feita uma introdução sobre a ferramenta PRISM, utilizada neste trabalho com a lógica de árvore de computação e sua extensão, a lógica de árvore de computação probabilística. No Apêndice C, é apresentado o código utilizado na verificação do protocolo PBCP.



## REDES VEICULARES AD HOC

Este capítulo apresenta fundamentos sobre tipos de comunicação aplicados às redes veiculares, uma visão geral dos conceitos utilizados neste trabalho relacionados às redes veiculares *ad hoc* e à formação de pelotão de veículos. Dentre esses aspectos, destacam-se: as principais características e desafios das redes veiculares *ad hoc*; características relacionadas à formação de veículos em pelotões; uma visão geral do padrão IEEE 802.11p e da camada de controle de acesso ao meio (MAC, *Media Access Control*) baseada nesse padrão. Além disso, é apresentada uma abstração da camada MAC. Por fim, são discutidos os mecanismos fundamentais para garantir um estabelecimento eficaz da detecção de vizinhança em uma rede veicular *ad hoc*.

### 2.1 TIPOS DE SISTEMA DE COMUNICAÇÃO APLICADOS ÀS REDES VEICULARES

Existem dois tipos principais de sistema de comunicação em sistemas distribuídos aplicados às redes veiculares: sistema de comunicação síncrono e assíncrono. Segundo Hadzilacos e Toueg (1994), através do sistema de comunicação síncrono, é possível estimar prováveis limites superior e inferior para o tempo de execução de um processo, para o atraso das mensagens e para a variação de desvio dos relógios físicos dos nós da rede em um sistema distribuído (COULOURIS et al., 2011). Em face disso, o sistema de comunicação síncrono faz fortes suposições sobre o tempo.

Um sistema de comunicação assíncrono, por outro lado, é aquele em que as mensagens podem ser adiadas por períodos arbitrários de tempo e os relógios podem estar fora de sincronia (COULOURIS et al., 2011).

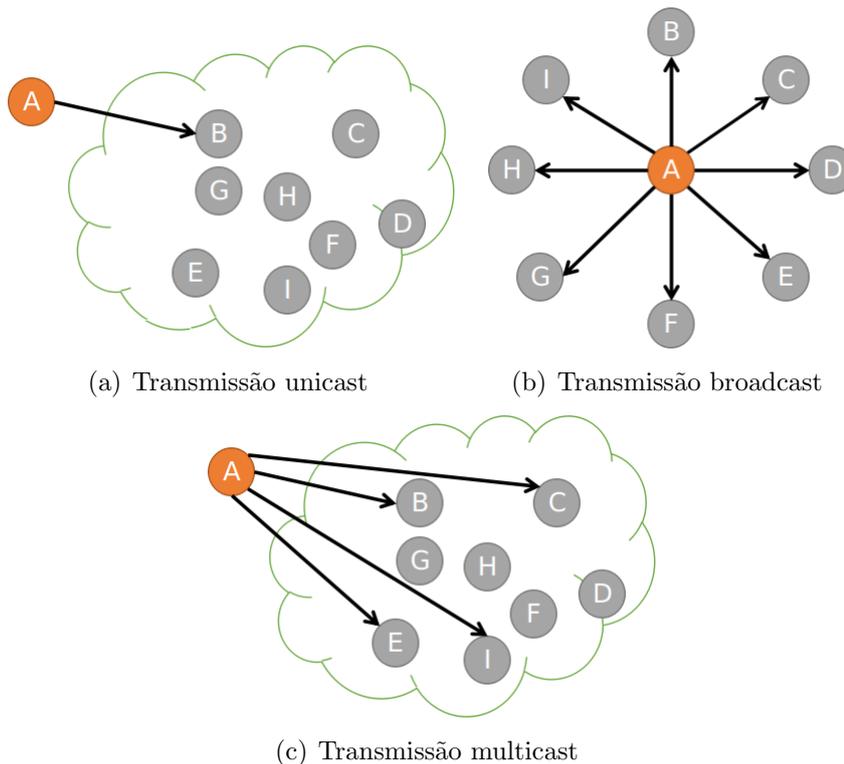
Os sistemas assíncronos são particularmente adequados para redes sujeitas a atrasos longos (FALL, 2004). Isso ocorre porque, em tais redes, a latência e a perda de pacotes são comuns, o que pode dificultar a sincronização precisa entre os dispositivos. Além disso, em sistemas distribuídos, em que os dispositivos estão fisicamente separados, a sincronização de relógios pode ser difícil de ser realizada.

Relacionado ao sistema de comunicação síncrono, este requer que os relógios físicos dos processos estejam sincronizados (HASAN et al., 2018). Especialmente para as redes

veiculares, os processos que fazem parte da rede devem ser sincronizados para trocar informações sobre o posicionamento dos veículos e mensagens de aviso para várias aplicações de segurança rodoviária.

A sincronização dos relógios pode ser realizada de várias formas, dependendo exclusivamente da acurácia pretendida. Uma das formas de sincronização de relógios é usar uma referência externa transmitida e distribuída usando um sistema de rádio global. Os sistemas de rádio globais típicos incluem o Sistema de Navegação por Satélite (GNSS, *Global Navigation Satellite System*) e a estação de ondas curtas (HASAN et al., 2018).

Em ambos os tipos de comunicação os veículos podem transmitir mensagens na rede em três diferentes formas: transmissões *unicast*, *broadcast* e *multicast*. Uma transmissão *unicast* ocorre quando se estabelece uma ligação entre dois nós (Figura 2.1(a)). São transmissões muito comuns dentro de qualquer rede, mas podem tornar-se bastante onerosas dependendo do tipo de tráfego necessário. As transmissões *broadcast* ocorrem quando se estabelece uma ligação em que os dados são transmitidos simultaneamente a todos os nós da rede (Figura 2.1(b)). O processo de *broadcast* é chamado de difusão. Evoluindo-se este conceito chega-se à difusão seletiva, ou *multicast*, em que um grupo específico de dispositivos recebe e trata a mensagem, conforme indicado na Figura 2.1(c).



**Figura 2.1** Tipos de transmissões

A difusão é um dos principais modos de entrega de mensagens nas redes veiculares (SANGUESA et al., 2016). Estas podem ser utilizadas para o envio de mensagens de rotina ou de emergência. As mensagens de rotina são transmitidas periodicamente a seus vizinhos para anunciar a localização, a velocidade, a direção, a aceleração e outros dados

úteis de um ou mais veículos. As mensagens de emergência são geradas e transmitidas quando surge uma situação de acidente em que os veículos deverão parar bruscamente ou quando houver uma alteração no movimento destes, mudando a faixa ou realizando uma ultrapassagem (YANG; ZHENG; SHEN, 2011).

Uma das vantagens do envio de mensagens por difusão é a sua simplicidade para implementar. Em redes *ad hocs* os nós podem se comunicar diretamente uns com os outros sem depender de um ponto de acesso central.

No entanto, esse método também apresenta algumas limitações. A difusão de mensagens consome largura de banda, o que pode causar sobrecarga na rede e reduzir a eficiência do sistema. Além disso, em redes *ad hoc* de grande escala, a difusão de mensagens pode causar congestionamento e aumentar a probabilidade de colisões e perda de pacotes (LI; CHIGAN, 2011). Assim, torna-se crucial otimizar tanto o intervalo de transmissão quanto a taxa de difusão de mensagens, a fim de alcançar um desempenho ideal e melhorar a comunicação nessas redes.

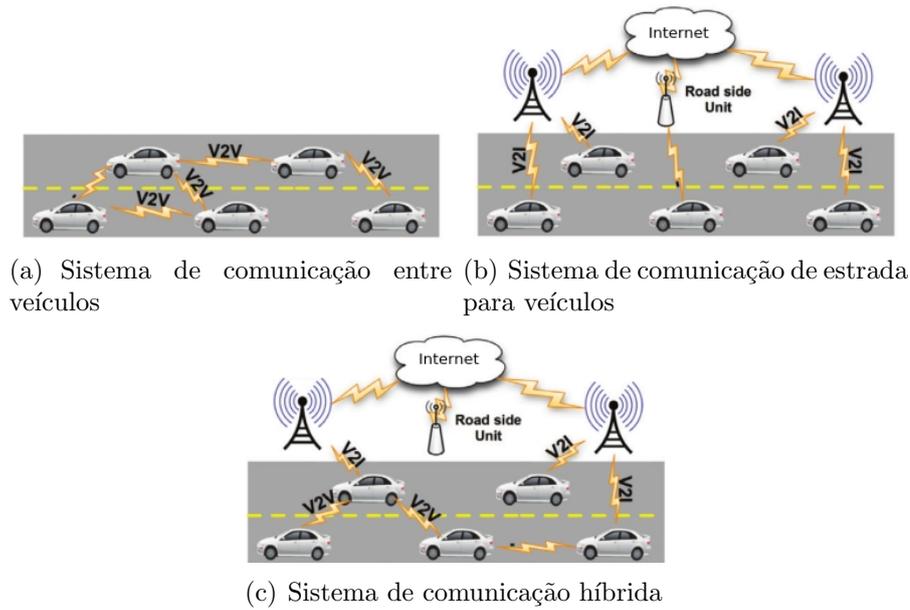
## 2.2 CARACTERÍSTICAS E DESAFIOS DAS VANETS

As VANETs contribuem para melhorar o desempenho e tornar o sistema de tráfego mais seguro e eficiente. Mais seguro significa fornecer informações sobre engarrafamentos, acidentes, condições perigosas da estrada, possíveis desvios, condições climáticas e localização de instalações (e.g. postos de gasolina e restaurantes). Mais eficiente significa um aumento da capacidade da rede viária, redução de congestionamentos e poluição, tempos de viagem mais curtos e previsíveis, melhor gerenciamento e controle da rede rodoviária (CUNHA et al., 2016). Essas aplicações são típicos exemplos das vantagens no uso das redes veiculares, cujo objetivo é melhorar a segurança, a eficiência, a conscientização e o aproveitamento urbano dos sistemas de transporte através do uso de novas tecnologias de informação e comunicação.

Conforme demonstrado por Mihail e Maria (2008), existem três principais sistemas relacionados ao processo de comunicação de redes veiculares: a comunicação entre veículos (IVC, *Inter-Vehicle Communication*), a comunicação da estrada e veículo (RVC, *Roadside-to-Vehicle Communication*) e a comunicação veicular híbrida (HVC, *Hybrid Vehicular Communication*). No sistema IVC a troca de mensagens entre veículos independe do suporte de uma infraestrutura fixa. Estas redes veiculares são do tipo *ad hoc*. O sistema RVC utiliza a infraestrutura da estrada para o estabelecimento de comunicação, principalmente para aplicações de coleta de informações e dados. Por fim, a comunicação híbrida combina os tipos, IVC e RVC.

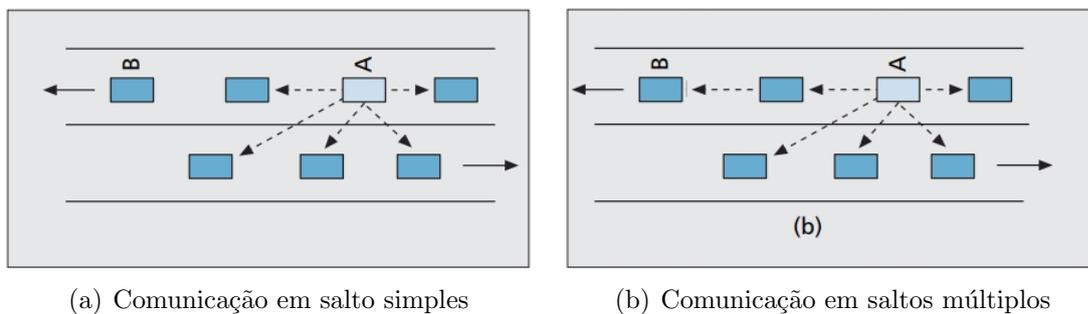
Na Figura 2.2 são mostrados os diferentes tipos de sistemas de comunicação entre veículos. Tanto os sistemas de comunicação de estrada para veículos (Figura 2.2(b)) quanto os sistemas híbridos (Figura 2.2(c)) possibilitam a conexão de longa distância como, por exemplo, à Internet. Diferentemente destes, a comunicação veículo a veículo limita-se à comunicação entre veículos (Figura 2.2(a)). O sistema de comunicação estritamente entre veículos será utilizado nesta dissertação.

A comunicação das IVCs pode ser realizada de duas formas: em um único salto (SIVC, *Simple-Inter-Vehicle Communication*) ou em múltiplos saltos (MIVC, *Multi-Inter-*



**Figura 2.2** Sistemas de comunicação das redes veiculares (CUNHA et al., 2016)

*Vehicle Communication*). Os sistemas SIVCs são úteis para aplicações que requerem comunicações de curto alcance e com restrições temporais brandas, a exemplo da fusão de faixa, do controle de cruzeiro cooperativo adaptativo etc. Quanto aos sistemas MIVCs, estes são mais complexos que os SIVCs e suportam aplicações que exigem comunicações de longo alcance como monitoramento de tráfego (MIHAIL; MARIA, 2008). A principal diferença entre os sistemas SIVC e MIVC é mostrada na Figura 2.3. No cenário em que o sistema de comunicação é SIVC, o veículo A pode enviar uma mensagem apenas para os carros que estão no seu alcance de transmissão, isto é, o veículo B nunca recebe a mensagem vinda do veículo A (Figura 2.3(a)). Por outro lado, em um sistema MIVC, o veículo A pode retransmitir a mensagem de tal modo que os veículos que não estejam ao seu alcance, como o veículo B, também recebam a mensagem. Portanto, um sistema MIVC requer uma camada de rede capaz de retransmitir as mensagens (Figura 2.3(b)).



**Figura 2.3** Tipos de compartilhamento

Os sistemas IVCs permitem que várias classes de aplicações possam tornar as viagens

por estrada mais seguras, evitando colisões, diminuindo o tempo de viagem, reduzindo congestionamentos, aumentando a capacidade das rodovias, bem como tornando o trajeto mais agradável através do envio de informações de interesse do condutor (FERNANDES; NUNES, 2010). Por não necessitar de infraestrutura de beira de estrada, este sistema é considerado muito barato, comparado aos outros sistemas.

Nesta dissertação consideramos a comunicação veículo a veículo, mais especificamente a SIVC. No entanto, a falta de infraestrutura leva a problemas de rede específicos, com isto necessitando de protocolos totalmente distribuídos e que possuam formas de compartilhamento seguro de mensagens.

As redes veiculares *ad hoc* são um caso particular das redes móveis *ad hoc* (MANET, *Mobile Ad Hoc Networks*). Estas são compostas por um conjunto de veículos e a comunicação entre estes é estabelecida em uma rede sem infraestrutura (veja Figura 2.4). A característica móvel dos veículos impõe uma condição adversa para que aplicações utilizadas nestas redes possam trocar informações.



**Figura 2.4** Típico cenário de uma VANET (SANGUESA et al., 2016)

Os métodos de transmissão de mensagem em uma rede veicular *ad hoc* pode ser orientada a eventos ou periódica (OMAR; ZHUANG; LIT, 2013). A comunicação orientada a eventos visa diminuir o tráfego de mensagens de controle em cenários com alta densidade de veículos, tais como, quando há uma mudança na composição de um grupo e os outros vizinhos necessitem ter esta informação em tempo real (GALAVIZ-MOSQUEDA et al., 2014; SHAH et al., 2018). A transmissão periódica por sua vez é utilizada para ampliar a percepção que os veículos possuem em relação à sua vizinhança. Diferentes aplicações de VANETs (e.g. pelotão de veículos) utilizam este método através do envio de mensagens de salto único para o estabelecimento de uma direção cooperativa. Este método permite que o veículo mantenha uma imagem da sua vizinhança. As mensagens são trocadas entre vizinhos em um único salto (HASSANABADI; VALAEE, 2014).

A detecção de veículos vizinhos é um serviço básico para que diferentes aplicações, utilizadas nas redes veiculares *ad hoc*, possam ter ciência de quais são os nós (veículos) da rede em seu entorno para, por exemplo, colaborar ou compartilhar informações. O gerenciamento de múltiplos pelotões, responsável pela fusão e subdivisão dos pelotões, são exemplos de aplicações que requerem um serviço de descoberta de vizinho subjacente (CABALLERO-GIL; CABALLERO-GIL; MOLINA-GIL, 2010).

Diversos desafios são impostos durante o processo de detecção de veículos vizinhos em uma VANET entre os quais podemos citar: o controle de colisão de mensagens; a rotatividade, ou seja, veículos que entram e saem do alcance uns dos outros; conexões intermitentes provocadas pela velocidade e mobilidade dos veículos; interferência no sinal de comunicação e redução da potência do sinal ao longo do meio de transmissão em função do distanciamento entre veículos; tamanho das mensagens trocadas entre os veículos; colisões de mensagens provocadas pelo envio simultâneo de mensagens. A colisão de mensagem pode ser acentuada a depender do tamanho da rede e se o método de transmissão de mensagem utilizada for o periódico (LI; CHIGAN, 2011; DU; WANG; ZHANG, 2020; ZHU et al., 2016; ABASSI, 2019).

### 2.3 COMUNICAÇÃO EM GRUPO APLICADA ÀS VANETS

A organização de veículos em grupo é útil para reduzir a propagação de informações irrelevantes e redundantes em situações em que há uma grande densidade de nós em uma área restrita e também pode ser usada em projetos de caravanas ligados em rede (MEZGHANI et al., 2015). Neste sentido, o grupo é considerado uma abstração importante para facilitar a colaboração entre nós vizinhos (VALÉRIO; PEDRO; FRANCISCO, 2006). Essa abstração permite que seja mantida uma lista, conhecida como visão dos nós ativos e conectados, possibilitando que cada nó do grupo saiba que são os nós que fazem parte de uma determinada computação distribuída.

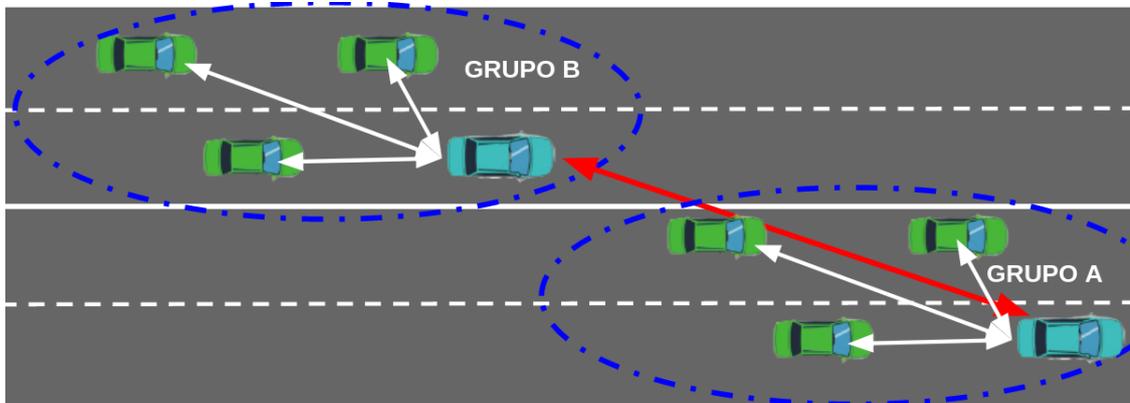
Para garantir a representação correta das informações de todos os nós, é fundamental que as visões estejam sempre atualizadas. Quando ocorrer uma mudança na composição do grupo, seja pela entrada ou saída de um veículo, os membros do grupo devem perceber a mudança e instalar uma nova visão. Para facilitar essas alterações na composição, é possível, por exemplo, que um dos membros assuma o papel de líder do grupo – sendo responsável por decidir e informar aos demais membros qual é a nova composição. Esses grupos são denominados de grupos hierárquicos ou assimétricos e, nesse exemplo particular, o líder também é responsável por coordenar as ações dos membros do grupo, aprovar a entrada de novos nós e excluir aqueles que apresentarem falhas ou cujas informações estejam indisponíveis por um período prolongado. Para a organização dos veículos em grupo, são necessárias três manobras básicas: o ingresso de um nó no grupo mediante solicitação de acesso, a saída do grupo por decisão própria do nó ou pelo líder do grupo, e a exclusão do grupo em caso de travamento ou indisponibilidade prolongada de informações.

Uma importante propriedade monotônica no contexto de instalação de visões em nós é a garantia de ordem na instalação das visões (CHOCKLER; KEIDAR; VITENBERG, 2001). Em outras palavras, se um nó  $p$  instala uma visão  $V$ , então ele se torna membro dessa visão. Além disso, se um serviço em um nó  $p$  instalar uma visão  $V'$  após a instalação

de uma visão  $V$ , a identificação de  $V'$  será maior do que a identificação de  $V$ . Isso garante que a mesma visão não seja instalada mais de uma vez e que, se dois nós instalarem as mesmas visões e, posteriormente, instalarem novas visões, elas serão instaladas na mesma ordem. Essa propriedade é fundamental para garantir a consistência da instalação de visões em sistemas distribuídos.

A construção de mecanismos em sistemas distribuídos com múltiplos grupos para operações de unificação e subdivisão em VANETs (PENG et al., 2017) apresenta desafios devido à independência dos grupos, o que permite que cada grupo trafegue em diferentes velocidades. Isso torna a rede mais dinâmica e sujeita a mudanças frequentes de conectividade, resultando em maiores latências e perdas de mensagens nas trocas de informações entre os grupos. Essas diferenças de mobilidade requerem uma abordagem cuidadosa na implementação de mecanismos de gerenciamento de grupos, a fim de garantir uma operação eficiente e confiável do sistema como um todo.

Em um ambiente com múltiplos grupos assimétricos, por exemplo, para que dois ou mais grupos realizem operações conjuntas, é necessário que os seus respectivos líderes entrem em um acordo – isto é, estejam cientes da visão atual uns dos outros, de modo a evitar que uma unificação de dois ou mais grupos seja realizada com informações desatualizadas. A Figura 2.5 ilustra um cenário em que dois grupos ( $A$  e  $B$ ) estabelecem comunicação com o objetivo de obterem informações um do outro, através dos seus líderes, em uma topologia veículo-a-veículo.

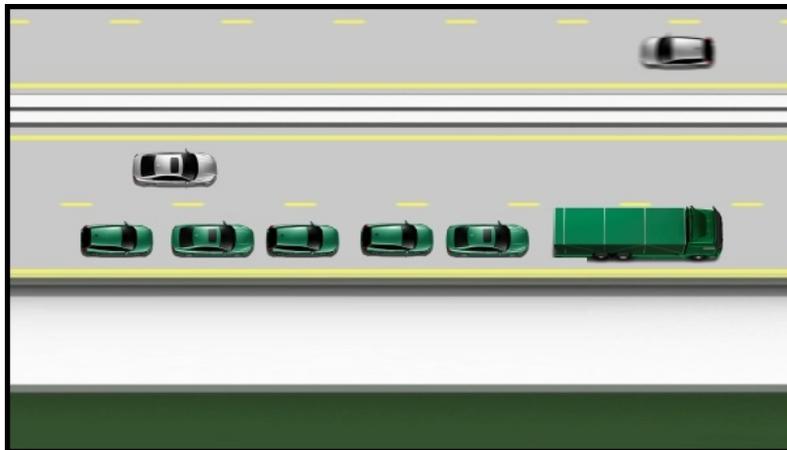


**Figura 2.5** Ilustração de uma comunicação entre grupos através dos líderes

Uma opção de aplicação para a formação de grupos de veículos que se apresenta como uma tecnologia habilitadora para a construção de sistemas de transporte colaborativos e automatizados, compostos por veículos autônomos (ou semi-autônomos) e dotados dos mecanismos necessários para se auto-organizarem em uma mesma via, é a organização do grupo em formação de pelotão (JIA et al., 2016; WANG; LIU; KATO, 2019). Contudo, é crucial que a organização dos veículos em formação de pelotão e as informações sobre eles estejam corretas e atualizadas, para que os nós possam tomar decisões cooperativas seguras e apropriadas.

## 2.4 PELOTÕES

A formação de um grupo de veículos em pelotão é basicamente uma forma automatizada (ou semi-automatizada) de condução de um conjunto de veículos inteligentes que visa reduzir o consumo de combustível e emissão de gás, alcançando um transporte seguro e eficiente. Em geral, um pelotão é composto por um líder (veículo que possui as informações de todos os membros do grupo) e demais seguidores (veículos que recebem informações fornecidas pelo líder). Os veículos que compõem o pelotão devem agir cooperativamente para controlar e gerenciar as ações do pelotão, incluindo formação do pelotão, fusão, divisão, manutenção etc (WANG; LIU; KATO, 2019). No exemplo apresentado na Figura 2.6, é possível observar um pelotão liderado por um caminhão, seguido por cinco veículos de passeio que recebem e seguem as informações do veículo líder.



**Figura 2.6** Ilustração do pelotão de veículos  
**Fonte:** Adaptado de Robinson e Coelingh (2018)

Por conta do mecanismo distribuído de controle automático e colaborativo das velocidades dos veículos do pelotão, integrados a outros mecanismos de gestão de manobras e direção segura, os veículos podem trafegar em distâncias mais curtas, evitando colisões e variações bruscas de velocidade. Isso, conseqüentemente, aumenta a segurança, o conforto e capacidade de veículos nas vias, e reduz congestionamento de tráfego e a emissão de gases poluentes (AMOOZADEH et al., 2015; CABALLERO-GIL; CABALLERO-GIL; MOLINA-GIL, 2010).

Para maximizar os benefícios da aplicação do pelotão, deseja-se que o maior número de veículos possível seja acomodado em um mesmo pelotão. Entretanto, se a quantidade de veículos aumenta demasiadamente, o controle das variações de velocidade dos veículos do pelotão pode se tornar ineficiente ou ineficaz (i.e., instável) (AMOOZADEH et al., 2015; HAIXIA et al., 2017), sendo necessário que o pelotão se subdivida em pelotões menores. Do mesmo modo, pelotões devem ser unificados quando o número de veículos diminui consideravelmente.

Uma vez que cada pelotão é independente, pelotões distintos podem trafegar com diferentes velocidades, o que traz desafios na construção de mecanismos distribuídos de

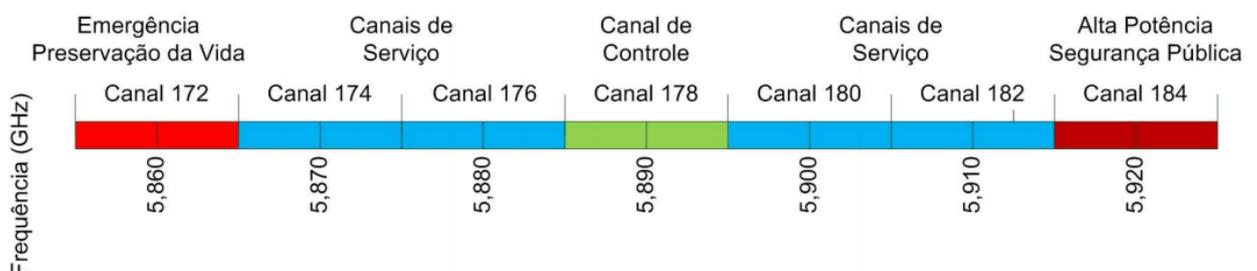
gerenciamento multipelotão, voltados para a realização das operações de unificação e subdivisão de pelotões sobre a estrutura da VANET (HAIXIA et al., 2017). Isso acontece porque essas diferenças de mobilidade tornam a estrutura da VANET mais dinâmica, estando sujeita a mudanças de conectividade mais frequentes, maiores latências e perdas de mensagens nas trocas de informações entre os pelotões. Deste modo, é um problema para o sistema de gerenciamento de multipelotões, distribuídos em cada pelotão, obter visões atualizadas e confiáveis sobre os demais pelotões com os quais podem colaborar para realizar as operações de gerenciamento (i.e., unificações e subdivisões de pelotões).

Durante as comunicações entre veículos em uma VANET, uma camada de acesso ao meio deve ser adotada para atender às características de mobilidade dos veículos e os diferentes requisitos do serviço de detecção de vizinhança. Com este propósito foi desenvolvido o padrão IEEE 802.11p. Este tem o objetivo de formalizar e normalizar as características básicas da comunicação sem fio em ambientes veiculares. (SUN et al., 2013).

## 2.5 PADRÃO IEEE 802.11P

Para possibilitar a comunicação entre os veículos a fim de oferecer um sistema de transporte inteligente e mais seguro, o IEEE desenvolveu um processo de padronização das comunicações para redes veiculares conhecido como WAVE, *Wireless Access in the Vehicular Environment*. O principal objetivo deste padrão de comunicação é prover a comunicação entre veículos (V2V) e entre veículos e infraestrutura (V2I).

A tecnologia de comunicação sem fio proposta para redes veiculares utiliza o sistema dedicado de comunicações de curto alcance ou DSRC, *Dedicated Short Range Communications*, que opera em uma banda de frequência dedicada de 5,85 GHz a 5,925 GHz, conforme mostrado na Figura 2.7, e sete canais de 10 MHz cada para transmissão. Suporta tanto aplicações de segurança quanto de entretenimento em ambientes de comunicação V2V e V2I e foi padronizado como IEEE 802.11p, constituindo a camada física e de controle de acesso ao meio (ALVES et al., 2009). O canal 178 é o canal de controle (CCH, *Control CHannel*), que é reservado para transmitir mensagens de controle (*beacons*). Os dois canais nas extremidades do espectro são reservados para usos especiais. Os demais são canais de serviço (SCH, *Service CHannel*) disponíveis para transferência de dados não seguros, para entretenimento e para aplicações de uso do usuário (AKBAR et al., 2014).



**Figura 2.7** Alocação de canal IEEE 802.11p (ALVES et al., 2009)

No padrão IEEE 802.11p a camada de controle de acesso ao meio (MAC, *Media Access*

*Control*) controla o hardware responsável pela interação com o meio de transmissão sem fio e fornece mecanismos de acesso ao canal de comunicação. As operações de comunicação são baseadas em um controle de acesso múltiplo com prevenção de colisão (CSMA/CA, *Carrier Sense Multiple Access with Collision Avoidance*), em que os veículos escutam o canal sem fio antes de enviar uma mensagem. Este método é necessário já que, nas redes veiculares, os transceptores dos veículos não podem detectar colisões de mensagens no canal durante as transmissões. Assim, dependendo do estado do canal, ocioso ou ocupado, a transmissão é iniciada ou adiada.

Embora o método utilizado no CSMA/CA seja útil para evitar colisões entre mensagens, em cenários onde há uma grande quantidade de veículos na via disputando o canal de comunicação, nem sempre isso é possível. O protocolo de comunicação veicular que utiliza o padrão 802.11p para troca de mensagens muitas vezes sofre com as colisões consecutivas quando há uma grande quantidade de veículos tentando enviar mensagens. Uma estratégia utilizada para evitar que mensagens simultâneas colidam é a alocação de um período de espera para que veículos possam acessar o meio (janela de contenção) e transmitir mensagens. Esta pode se tornar insuficiente quando a quantidade de veículos aumenta posteriormente. Além disso, após uma transmissão bem-sucedida, o tamanho da janela de contenção é redefinido novamente para o valor mínimo, sem qualquer memória do estado do canal atual (DENG et al., 2011).

A competição pelo acesso ao meio por parte dos veículos, associada a uma janela de contenção e o tempo em que as mensagens percorrem para chegar até o destino podem provocar atrasos excessivos no encaminhamento de mensagens, comprometendo assim o processo de detecção de outros veículos vizinhos (BILSTRUP et al., 2008).

Além das colisões e atrasos provocados pela competição do acesso ao meio por parte dos veículos outro fator importante que pode degradar o desempenho dos veículos é a interferência no canal. Quando o algoritmo CSMA/CA é utilizado nas redes sem fio, interferências externas no sinal de rádio podem facilmente obstruir uma área geográfica, de forma que os veículos nesta área adiariam seu acesso ao meio de comunicação mesmo que não houvesse tráfego de dados em tempo real presente. Um sistema CSMA/CA é, portanto, mais suscetível a interferências, uma vez que nenhum acesso ocorrerá enquanto a interferência for detectada no canal (KHAIRNAR; KOTTECHA, 2013).

O tamanho das mensagens enviadas no canal desempenha um papel crucial na comunicação das redes veiculares, especialmente no funcionamento do protocolo de acesso ao meio CSMA/CA. Um conceito fundamental relacionado às unidades de dados em redes sem fio é o tamanho máximo que uma unidade de dados pode ser transmitida com sucesso através da rede. Nesse contexto, é importante entender que, além da carga útil que será transmitida, uma mensagem em uma rede veicular contém também uma estrutura básica de dados que possuem informações de controle. A próxima subseção abordará a importância desse conceito e suas implicações nas comunicações em redes veiculares.

### **2.5.1 Unidade de dados de protocolo da camada de controle de acesso ao meio**

A Unidade de Dados do Protocolo da Camada de Controle de Acesso ao Meio (MPDU, *MAC Protocol Data Unit*) e a Unidade de Dados do Serviço da Camada de Controle

de Acesso ao Meio (MSDU, *MAC Service Data Unit*) são termos usados em redes para descrever diferentes tipos de unidades de dados transmitidas através do protocolo de comunicação (MUHAMMAD; ZHANG; DYUO, 2018).

MPDU refere-se à unidade de dados que é transmitida através do meio de comunicação. Esta é a unidade básica de dados que contém os dados do usuário, bem como informações de controle e cabeçalhos necessários para que os dispositivos entendam a mensagem. Em outras palavras, a MPDU é a estrutura completa da mensagem que é enviada entre dispositivos em uma rede (SUN, 2011).

O MSDU é uma parte da MPDU que contém apenas a carga útil, ou seja, os dados reais do usuário, sem os cabeçalhos e informações de controle adicionais. Em outras palavras, o MSDU representa o conteúdo efetivo da mensagem transmitida, excluindo elementos de gerenciamento e controle (SUN, 2011). O tamanho máximo do MSDU é limitado a 2304 bytes. (LABERTEAUX; HARTENSTEIN, 2009). Essa restrição de tamanho tem implicações significativas no design e na eficiência da comunicação em redes veiculares. O uso otimizado do espaço disponível para transmissão de dados é essencial para garantir a confiabilidade e a eficácia das trocas de informações entre os veículos e a infraestrutura de transporte.

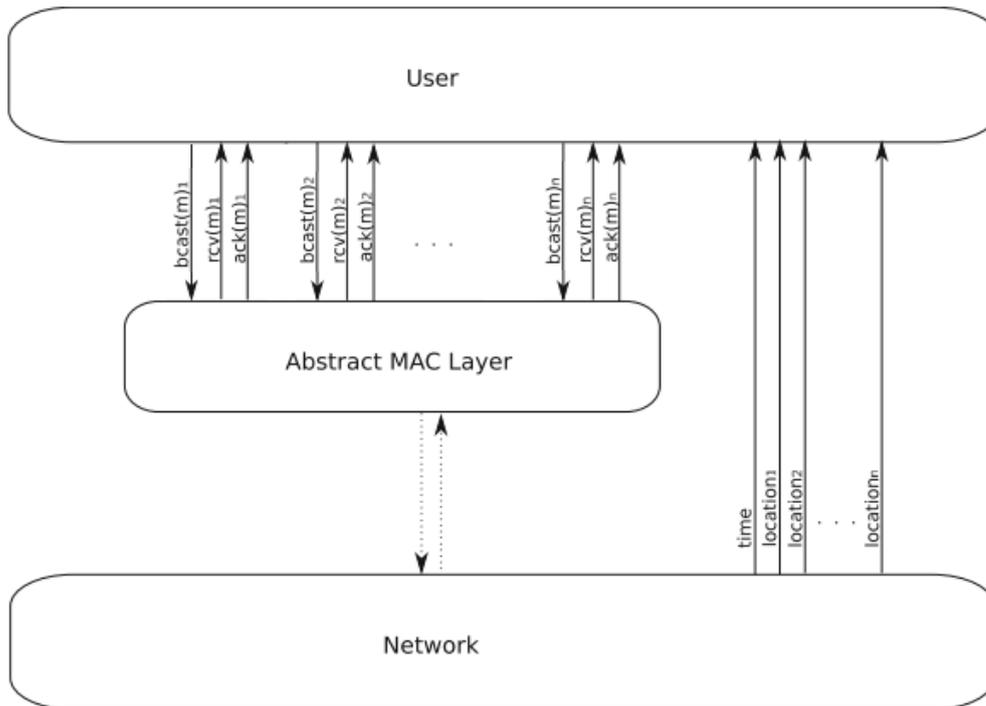
Em resumo, a MPDU é a unidade completa de dados que inclui informações de controle e carga útil, enquanto a MSDU é apenas a carga útil, que é a parte essencial da mensagem que contém os dados que estão sendo transmitidos entre dispositivos sem fio.

## 2.6 CAMADA MAC ABSTRATA

A diversidade de desafios e suposições de comunicação possíveis torna complexo o estudo relacionado ao comportamento das redes veiculares *ad hoc*, principalmente em relação aos limites de tempo para que a troca de mensagens entre veículos seja estabelecida. Os resultados para este conjunto de desafios levantados podem se mostrar inválidos para um conjunto ligeiramente diferente de suposições. O trabalho de Kuhn, Lynch e Newport (2009) aborda essas questões através da camada de acesso ao meio.

A camada MAC abstrata foi proposta inicialmente por Kuhn, Lynch e Newport (2009). *A priori* o uso desse serviço é para permitir que projetos e análises de algoritmos que compartilhem informações entre nós em uma rede de computadores não necessitem lidar com problemas como gerenciamento de contenção (ou método de acesso aleatório ao meio) e detecção de colisão. Em vez disso, os problemas de baixo nível, conforme citados pelos autores, poderão ser tratados como limites abstratos de tempo.

No modelo proposto por Kuhn, Lynch e Newport (2009) a arquitetura do sistema de rede *ad hoc* consiste em três componentes principais: a camada da rede, a camada MAC abstrata e a camada de usuário como mostrado na Figura 2.8. A camada de rede modela as propriedades relevantes do mundo real: tempo, localização e comportamento da camada física. Esta camada fornece uma interface de camada física que captura a comunicação de baixo nível no canal sem fio. Ele também pode gerar a localização e a hora do usuário. Não é assumido neste modelo uma interface externa para controlar o movimento. Ou seja, a mobilidade é inteiramente encapsulada na camada da rede e independente do comportamento de outros componentes do sistema.



**Figura 2.8** Arquitetura do sistema de rede *ad hoc* utilizando a camada MAC abstrata  
**Fonte:** Adaptado de Kuhn, Lynch e Newport (2009)

A camada MAC abstrata faz interface entre a aplicação de usuário e a rede. Cada usuário  $i$  interage com a camada MAC através das entradas  $bcast(m)$  e saídas da camada MAC  $rcv(m)$  e  $ack(m)$ , em que  $m$  é a mensagem. Nesta análise propriedades relevantes do mundo real como tempo, localização e o comportamento da camada física (e.g. relação sinal-ruído) são restringidas em uma abstração da camada MAC. Os limites de latência de entrega das mensagens são definidos pelo autores como:

- Limite para recebimento  $rcv(m)$ : Este é um limite superior na quantidade de tempo até que um processo receba uma mensagem enviada por um vizinho.
- Limite de progresso  $bcast(m)$ : Este é um limite superior na quantidade de tempo até que um processo receba alguma mensagem e todas as outras mensagens enviadas anteriormente sejam recebidas.
- Limite de confirmação  $ack(m)$ : Este é um limite superior da quantidade de tempo até que o processo receba uma confirmação de mensagem enviada. Para limitar o valor no pior caso calculado é considerado o tempo de contenção no remetente e em todos os vizinhos da rede.

A camada de usuário modela  $n$  nós com rótulos exclusivos de  $1, \dots, n$ . Cada nó  $i$  se conecta a uma camada MAC por meio das interfaces  $bcast$ ,  $rcv$  e  $ack$  descritas acima.

Ele também pode receber a localização da rede e as saídas de tempo, dependendo do que é necessário para o protocolo que está sendo modelado.

Na próxima seção apresentamos aspectos importantes relacionados ao serviço básico de detecção de vizinhança. A identificação de veículos vizinhos é uma tarefa complexa quando se considera os aspectos de redes dinâmicas como as VANETs. Mostraremos que esta operação não é útil apenas para a segurança rodoviária, mas que também é uma etapa necessária para aplicações cooperativas como o gerenciamento entre múltiplos pelotões.

## 2.7 DETECÇÃO DE VIZINHANÇA EM UMA REDE VEICULAR AD HOC

A detecção de vizinhança em uma rede *ad hoc* consiste em determinar todos os nós com os quais um determinado nó pode se comunicar diretamente possibilitando a comunicação cooperativa entre veículos (CHEN; BIAN, 2016). Esse processo é realizado por meio da troca de mensagens entre os dispositivos da rede, permitindo que cada um construa uma lista de vizinhos sem depender de nós intermediários, já que os dispositivos se comunicam sem uma infraestrutura centralizada. Há desafios na detecção de vizinhança em VANETs devido à alta mobilidade dos veículos e a suscetibilidade à interferência, obstrução de sinal e variação de propagação, o que pode levar a falsos positivos ou negativos. É importante enfrentar esses desafios para garantir o funcionamento eficiente e seguro das redes veiculares *ad hoc*, a fim de assegurar a eficiência da detecção de vizinhança, visando ao pleno funcionamento dessas redes e ao alcance dos benefícios esperados (AHMED; GHARAVI, 2018; ABASSI, 2019).

Um ponto importante quando consideramos os desafios durante o processo de detecção de vizinhança é a identificação de alguma estabilidade que garanta que os nós em uma rede sejam considerados detectados. Essa estabilidade é especialmente importante quando um grupo de veículos precisa cooperar entre si. Conforme indicado na Seção 1.1, de acordo com o trabalho de Arantes et al. (2010), a estabilidade de uma vizinhança ocorre quando um conjunto de nós vizinhos podem se comunicar uns com os outros por meio de enlaces dinâmicos dentro de um tempo limite. Assim, se um nó  $q$  é alcançável a partir de um nó vizinho  $p$ , e vice-versa, então  $q$  e  $p$  fazem parte de uma vizinhança estável.

Ao lidar com a comunicação entre vizinhos em uma rede *ad hoc* composta por veículos, é importante considerar as principais métricas que medem a capacidade dos veículos se detectarem mutualmente como vizinhos. A primeira métrica é o atraso na descoberta, que é crucial para garantir uma comunicação eficiente entre os veículos. Dependendo do cenário de aplicação, é necessário minimizar o atraso esperado na descoberta.

Nas próximas subseções, serão abordados os quatro aspectos fundamentais para garantir um estabelecimento eficaz da detecção de vizinhança em uma rede veicular *ad hoc*. Esses aspectos incluem o alcance de transmissão, o tamanho das mensagens, a periodicidade no envio das mensagens e a quantidade de nós que utilizam o canal de controle para a troca de informações.

### 2.7.1 Alcance de transmissão

O alcance de transmissão desempenha um papel crucial em uma VANET, pois é fundamental para a comunicação entre os nós e, conseqüentemente, para a detecção de vizinhança. Um alcance de transmissão adequado é essencial para garantir que as informações sejam distribuídas rapidamente e possam alcançar todos os veículos na vizinhança de forma eficiente. É por meio desse alcance que a cooperação e a troca de dados entre os veículos são facilitadas, permitindo um funcionamento eficaz e seguro das redes veiculares.

Cada nó em uma rede *ad hoc* possui um alcance de transmissão limitado, ou seja, uma distância máxima na qual pode enviar e receber dados. A detecção de vizinhança é baseada na proximidade física entre os nós, permitindo que um nó detecte e estabeleça conexão apenas com outros nós que estejam dentro de seu alcance de transmissão. Portanto, o primeiro critério relacionado à detecção de vizinhança consiste em verificar se os nós estão dentro do alcance um do outro (ARENA; PAU; SEVERINO, 2020).

Na rede veicular *ad hoc* que utiliza o padrão 802.11p, também conhecido como padrão de comunicação sem fio para veículos, a distância máxima de comunicação entre dois nós é influenciada por diversos fatores, como potência de transmissão, sensibilidade do receptor, interferências e obstáculos presentes no ambiente. Em condições ideais, a faixa típica de comunicação pode alcançar alguns quilômetros (STIBOR; ZANG; REUMERMAN, 2007). Essa especificação visa atender às demandas de comunicação veículo-a-veículo (V2V) e proporcionar a conectividade necessária para aplicações e serviços em redes veiculares.

### 2.7.2 Tamanho das mensagens

O tamanho da mensagem também desempenha um papel importante nas redes veiculares *ad hoc*. Nessas redes, onde os dispositivos se comunicam diretamente entre si, a divisão da mensagem em pacotes é necessária para a transmissão. O tamanho da mensagem afeta diretamente a quantidade de pacotes e, conseqüentemente, o tempo necessário para transmiti-la (ROY; ACHARYA; DASBIT, 2018).

Em redes veiculares *ad hoc*, onde os veículos estão em constante movimento, a mobilidade pode causar mudanças rápidas nas condições de comunicação. Quando uma mensagem é grande, existe uma maior probabilidade de ocorrer perda de mensagens durante a transmissão, devido ao tempo prolongado de transmissão e à suscetibilidade a interferências.

Além disso, nas VANETs, a largura de banda é compartilhada entre os dispositivos presentes na área de comunicação. Quando uma mensagem grande está sendo transmitida, ela ocupa uma maior parte da largura de banda disponível, o que pode levar a um congestionamento na rede e aumentar a latência para todas as comunicações.

Uma maneira de lidar com essa influência é dividir a mensagem em pacotes menores, através da segmentação, e reagrupá-los no destino final. Isso reduz o tempo de transmissão de cada pacote e diminui a probabilidade de perda. No entanto, é importante considerar que a segmentação e o reagrupamento de pacotes adicionam sobrecarga ao processamento dos dispositivos e à transmissão dos pacotes de controle (PITKANEN; KERANEN; OTT, 2008).

Em suma, o tamanho da mensagem tem impacto nas redes veiculares ad hoc devido ao tempo de transmissão, às perdas de pacotes e à ocupação da largura de banda. A segmentação e o reagrupamento de pacotes são técnicas que podem ser aplicadas para mitigar esses efeitos, mas é necessário equilibrar a eficiência do processamento com o tamanho da mensagem para obter um desempenho adequado na rede.

### 2.7.3 Periodicidade no envio de mensagens

O intervalo entre o envio de mensagens em uma rede sem fio *ad hoc* exerce uma influência significativa no desempenho durante o processo de detecção de vizinhança. Esse intervalo se refere ao tempo decorrido entre o envio de uma mensagem e o envio da próxima mensagem.

Em redes veiculares *ad hoc*, onde os dispositivos se comunicam diretamente entre si, é importante garantir um intervalo adequado entre o envio de mensagens. Um intervalo muito curto pode levar a colisões de pacotes, onde múltiplas mensagens são transmitidas ao mesmo tempo e se sobrepõem, resultando em perdas de pacotes e atrasos na transmissão (PARK et al., 2017). Por outro lado, um intervalo muito longo pode resultar em subutilização da capacidade da rede, com períodos de ociosidade em que nenhum dado está sendo transmitido.

Em redes sem fio, o compartilhamento do meio de transmissão é uma preocupação central. Quando vários dispositivos tentam transmitir mensagens simultaneamente, ocorrem colisões, e os pacotes precisam ser retransmitidos. Isso leva a um aumento da latência e a uma redução da taxa de entrega efetiva.

Para evitar colisões e melhorar o desempenho, são utilizados mecanismos de controle de acesso ao meio. Esses mecanismos determinam como os dispositivos podem acessar o meio de transmissão e coordenam a transmissão de dados. Em redes veiculares *ad hoc*, o protocolo comumente usado é o CSMA/CA, onde os dispositivos escutam o canal antes de transmitir e aguardam um período de tempo aleatório antes de tentar transmitir novamente em caso de colisão (RAMANATHAN et al., 2018).

O intervalo entre o envio de mensagens também está relacionado à taxa de transmissão e à capacidade da rede. Uma taxa de transmissão mais alta permite que as mensagens sejam enviadas com intervalos menores, resultando em uma maior quantidade de dados transmitidos em um determinado período de tempo. No entanto, é importante considerar a capacidade da rede e as limitações do meio de transmissão para evitar congestionamentos e degradação do desempenho.

Em resumo, o intervalo entre o envio de mensagens em redes veiculares *ad hoc* desempenha um papel crucial no desempenho da comunicação. É necessário encontrar um equilíbrio entre intervalos adequados para evitar colisões de pacotes e garantir uma taxa de transmissão eficiente, considerando as características da rede e os mecanismos de controle de acesso ao meio utilizados.

### 2.7.4 Quantidade de nós que utilizam o canal de comunicação

A quantidade de nós que utilizam o canal de comunicação em uma rede sem fio ponto a ponto, como as redes veiculares *ad hoc*, desempenha um papel importante no desempenho

do sistema. Esse fator refere-se ao número de dispositivos que estão transmitindo dados simultaneamente no canal de comunicação.

Nas VANETs, em que os veículos são os nós da rede, a quantidade de nós ativos que utilizam o canal de comunicação pode variar dinamicamente. Quanto maior for a quantidade de nós transmitindo dados ao mesmo tempo, maior será a competição pelo acesso ao meio de transmissão.

O aumento da quantidade de nós ativos resulta em um maior congestionamento na rede, o que pode levar a um aumento da latência e a uma redução na taxa de entrega efetiva. Isso ocorre porque, em redes sem fio, o compartilhamento do meio de transmissão é fundamental e o acesso ao canal precisa ser coordenado para evitar colisões e garantir a transmissão adequada dos dados (RAMANATHAN et al., 2018).

Quando muitos nós tentam transmitir dados simultaneamente, ocorrem colisões de pacotes, onde as mensagens se sobrepõem e se perdem. Como resultado, os pacotes perdidos precisam ser retransmitidos, o que aumenta a latência e compromete o desempenho da rede. Além disso, o aumento da quantidade de nós ativos também pode levar ao esgotamento da capacidade da rede, resultando em uma menor taxa de entrega de dados para cada nó (MAHAJAN et al., 2006).

Para lidar com esse desafio, são utilizados mecanismos de controle de acesso ao meio nas redes sem fio, como o CSMA/CA. Esses mecanismos permitem que os dispositivos escutem o canal antes de transmitir e aguardem um período de tempo aleatório antes de tentar transmitir novamente em caso de colisão. Essa abordagem ajuda a reduzir a ocorrência de colisões, mas não elimina completamente o problema em situações de alta carga de tráfego (RAMANATHAN et al., 2018).

### **2.7.5 Estudos dos prazos necessários para detecção de vizinhança nas redes veiculares ad hoc**

A análise dos tempos necessários para detectar vizinhos está intimamente ligada aos atrasos das mensagens na comunicação em redes *ad hoc*. Os tempos de entrega das mensagens em uma rede *ad hoc* são fundamentais para garantir a eficiência da comunicação entre os nós cooperativos. A análise dos tempos de entrega permite identificar gargalos na rede e ajustar parâmetros de transmissão para melhorar o desempenho da rede veicular *ad hoc*.

O estudo realizado por Cornejo, Viqar e Welch (2014) aborda a relação entre a detecção de vizinhos e os prazos para entrega de mensagens, que são definidos como tempos limite de progresso. O artigo destaca a importância dos tempos limite de progresso para a detecção confiável dos vizinhos em uma rede *ad hoc*, já que é necessário que as mensagens sejam entregues dentro de um limite de tempo para que os nós da rede consigam se comunicar de forma eficaz. Assim, é fundamental ter em mente que prazos precisos para a detecção de vizinhos são cruciais para garantir a efetividade e eficiência das redes *ad hoc*.

No artigo (RAJESH; GNANASEKAR, 2016) é proposto um método de detecção de vizinhos que utiliza uma combinação de técnicas baseadas em tempo e em distância para aumentar a precisão e a eficiência da detecção de vizinhos. O artigo argumenta que a detecção de vizinhança com um tempo limite de progresso curto e preciso é crucial para

o desempenho geral de uma rede *ad hoc* móvel.

Para garantir a correção dos algoritmos de detecção de vizinhança apresentados no trabalho de Rajesh e Gnanasekar (2016), eles utilizam a especificação formal, uma técnica que permite verificar a correção lógica dos protocolos.

Em resumo, o artigo destaca a importância de estudar e aprimorar a detecção de vizinhos em redes móveis *ad hoc* para melhorar o desempenho geral da rede.

## 2.8 CONCLUSÃO

Neste capítulo, apresentamos uma rápida introdução sobre conceitos relacionados às redes veiculares, que apoia o presente trabalho de dissertação. Além disso, abordamos aspectos importantes sobre a detecção de vizinhança, que é o serviço básico nas redes veiculares *ad hoc*. Foram introduzidos conceitos relativos às características e desafios dessas redes, que operam sem uma infraestrutura definida. Tratamos sobre o padrão IEEE 802.11P e como ele é aplicado nas redes veiculares *ad hoc*, abrangendo também aspectos gerais da abstração do acesso ao meio de comunicação.



## VERIFICAÇÃO DE MODELOS COM O PRISM

Para verificar a correção de um protocolo muitas vezes é utilizado a verificação de modelos. A verificação de modelos é uma técnica que visa comprovar a correção do sistema através da exploração de todos os estados possíveis do modelo representado através de um modelo finito de transições de estados. Segundo Baier e Katoen (2008), a verificação ocorre tradicionalmente em três fases:

- **Modelagem:** esta etapa consiste em construir um modelo do sistema, utilizando uma linguagem formal utilizando uma ferramenta de modelagem. A estrutura que contém todos os comportamentos possíveis é conhecida como espaço de estados do sistema;
- **Especificação:** nesta fase são especificadas as propriedades comportamentais desejáveis do sistema descritas normalmente através de lógicas temporais;
- **Verificação:** esta etapa consiste em submeter o modelo e as especificações a um verificador de modelos. O objetivo é verificar se todos os comportamentos possíveis do modelo são atendidos através de sua especificação. O verificador pode produzir como resultado um valor verdade que indica se a especificação é satisfeita ou não, no modelo. Em caso negativo, é fornecida uma sequência de estados alcançáveis, chamada de contraexemplo, que demonstra que a especificação não é satisfeita no modelo.

O verificador de modelos PRISM (*PRobabilistic Symbolic Model checker*) é uma ferramenta para modelagem e análise de correção de sistemas que exibe comportamento probabilístico. Baseia-se na construção de um modelo matemático preciso de um sistema a ser analisado. As propriedades desse sistema são expressas formalmente na lógica temporal e analisadas automaticamente em relação ao modelo construído.

A ferramenta PRISM incorpora várias lógicas temporais probabilísticas bem conhecidas. Contudo, para o escopo desta dissertação, atentaremos-nos à lógica de árvore de computação e sua extensão, a lógica de árvore de computação probabilística. Estas são explicadas no Apêndice B.

O PRISM tem sido usado para analisar sistemas de muitos domínios de aplicação diferentes, incluindo protocolos de comunicação, como por exemplo, redes veiculares (JIN; ZHANG; ZHU, 2017), algoritmos distribuídos aleatórios (FKAIER; KHALGUI; FREY, 2021), protocolos de segurança (OUCHANI, 2018) e muitos outros.

### 3.1 LINGUAGEM DE MODELAGEM PRISM

A modelagem em PRISM é realizada através de uma linguagem simples e de alto nível, baseada em estados. Esta compreende um conjunto de módulos que representam diferentes componentes do sistema que estão sendo modelados. O estado de cada módulo é representado por um conjunto de variáveis e seus comandos (BRUNO et al., 2015). As variáveis descrevem os possíveis estados dos componentes. Os comandos descrevem o comportamento de cada componente. Estes são indicados por colchetes no início, no qual são separados em *guarda* e *atualização*. Os comandos são especificados como:

$$\boxed{\phantom{x}} \textit{guarda} \rightarrow \textit{prob}_1 : (\textit{atualizacao}_1) + \dots + \textit{prob}_n : (\textit{atualizacao}_n);$$

em que *guarda* é um predicado sobre todas as variáveis no modelo (locais e globais),  $\textit{prob}_1 + \dots + \textit{prob}_n$  são valores reais (não negativos) que determinam as probabilidades para as mudanças de estados, sendo que a soma das probabilidades é igual a 1.

Cada atualização  $i$  é dada por  $(x'_{i1} = u_{i1}) \ \& \ (x'_{i2} = u_{i2}) \ \& \ \dots \ \& \ (x'_{ik} = u_{ik})$ , em que  $u_{ij}$  são expressões que fornecem novos valores às variáveis, tal que  $x'_{ij}$  representa o valor de  $x_{ij}$  no novo estado.

Um exemplo de código em PRISM é ilustrado na Figura 3.1.

```
// A coin process

dtmc

const int HEADS = 1;
const int TAILS = 2;

module coin
  x : [0..3] init 0;

  [] (x = 0) -> 0.5 : (x' = HEADS) + 0.5 : (x' = TAILS);
  [] (x > 0) -> 1 : (x' = x);
endmodule
```

**Figura 3.1** A linguagem PRISM: Exemplo de um jogo de moeda eletrônica

**Fonte:** Adaptado de Kwiatkowska, Norman e Parker (2006)

Este código implementa um jogo de lançamento de moeda, que atribuirá *HEADS* ou *TAILS* à variável  $x$  com probabilidade 0.5 ( $x$  denota variável atualizada) quando  $x$  é zero, e caso contrário, reterá o valor anterior de  $x$ . Neste caso, existe apenas um estado inicial, mas PRISM permite a especificação de um conjunto de estados iniciais.

Para verificar um modelo construído em PRISM, é necessário identificar propriedades do modelo que possam ser avaliadas pela ferramenta. A linguagem de especificação de

propriedades em PRISM inclui várias lógicas temporais probabilísticas bem conhecidas, como por exemplo PCTL.

### 3.1.1 Especificação de propriedade

A lógica temporal PCTL permite a especificação de uma ampla gama de propriedades relevantes do sistema. Podemos citar pelo menos três tipos que são utilizadas nesta dissertação: *safety* (“algo ruim nunca acontece”), *liveness* (“algo de bom acabará acontecendo”) e *rewards* (“a cada estado alcançado, uma recompensa é adicionada”) (BAIER; KATOEN, 2008), (KWIATKOWSKA; NORMAN; PARKER, 2006).

A ferramenta PRISM pode fornecer resultados verdadeiros e falsos (se a descrição do sistema satisfaz a propriedade ou não) e retornar um valor probabilístico ou quantitativo esperado das propriedades. Existem três operadores principais para especificar propriedades: o operador  $P$ , que se refere à probabilidade de ocorrência de um evento; o operador  $S$ , que se refere à probabilidade em um estado estacionário; e o operador  $R$ , que se refere ao valor esperado das recompensas.

O operador  $P$  é usado para raciocinar sobre a probabilidade da ocorrência de um evento, conforme indicado nos exemplos abaixo:

$$P < 0.1[F \leq 100 \text{num\_errors} > 5]$$

*”a probabilidade de ocorrer mais de 5 erros nas primeiras 100 unidades de tempo é menor que 10%”*

$$P = ?[! \text{proc2\_terminate} \ U \ \text{proc1\_terminate}]$$

*”qual a probabilidade de que o evento `proc1_terminate` termine antes do evento `proc2_terminate`”* - a exclamação “!” antes de “`proc2_terminate`” indica a negação desse evento.

Os modelos PRISM podem ser especificados com informações sobre recompensas. A ferramenta pode analisar propriedades relacionadas aos valores esperados dessas recompensas. Isso é obtido usando o operador  $R$ , que funciona de maneira semelhante aos operadores  $P$  e  $S$ , e pode ser usado em uma consulta com valor booleano, por exemplo:

$$R \leq 9.5[Fz = 2]$$

*”a partir de um estado `s` o valor da recompensa será menor ou igual a 9.5 se no futuro um estado `z` for igual a 2”*

PRISM também suporta a verificação de modelo das lógicas temporais em que não é utilizado quantificação probabilísticos em suas propriedades, tais como CTL (lógica da árvore de computação).

Alguns exemplos de propriedades de CTL especificadas no verificador PRISM.

$E[F \text{ ”goal”}]$  - Existe um caminho no futuro que é alcançado um estado que satisfaça “`goal`”

$A[Gx \leq 10]$  - A variável  $x$  tem o valor máximo de 10 ao longo de todos os caminhos do modelo

$E[F \text{ "primeiro"} \& (X \text{ "segundo"})]$  - Existe um caminho no qual o rótulo "primeiro" se torna verdadeiro e o rótulo "segundo" é verdadeiro imediatamente depois

$A[(GFx = 1) | (GFx = 2)]$  - Em todos os caminhos,  $x = 1$  ou  $x = 2$  é verdadeiro infinitamente

A ferramenta PRISM possui uma interface gráfica que auxilia na construção da especificação de propriedades do modelo. A Figura 3.2 mostra uma captura da tela da interface do PRISM, em que é possível realizar a configuração das especificações de um modelo. Para isto, basta selecionar a guia "Properties". Em seguida, poderá adicionar especificações e salvar em um arquivo de extensão "props", criar ou editar especificações existentes. Na aba "property editor" é feita a especificação do modelo.

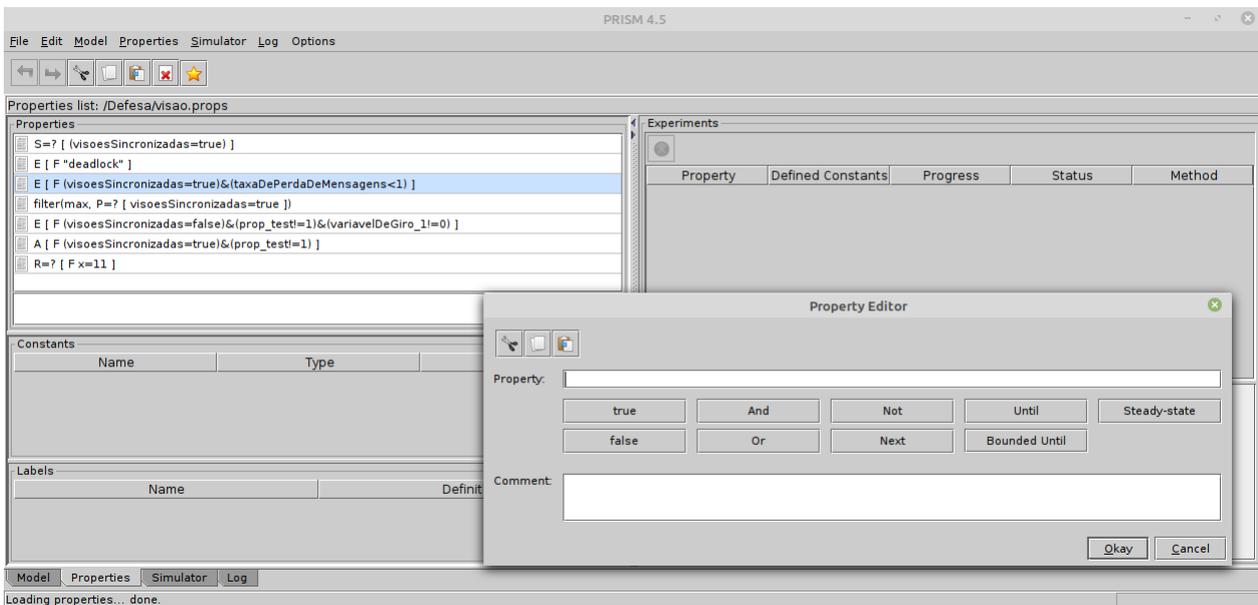


Figura 3.2 Especificando propriedade em PRISM

### 3.1.2 Contraexemplos e testemunhas

Em PRISM é possível verificar uma propriedade CTL do tipo  $A[G \text{ inv}]$ . Se o resultado da verificação for falso, será gerado um contraexemplo na forma de um caminho que alcança um estado em que "inv" não é verdadeiro. Isso é exibido no simulador (na interface gráfica ou na linha de comando). Da mesma forma, se você verificar  $E[F \text{ objetivo}]$ , se o resultado for verdadeiro, uma testemunha (um caminho que alcança o estado "objetivo") será gerada. O mesmo se aplica à verificação de propriedades PCTL.

Para ilustrar, considere um exemplo envolvendo a probabilidade de detecção de vizinhança entre líderes de pelotões de veículos:

Suponhamos que estamos verificando a seguinte propriedade PCTL:

$$P > 0,8[(F (\text{lider}_A \text{ detecta}_v \vee \text{lider}_B \text{ detect}_v))]$$

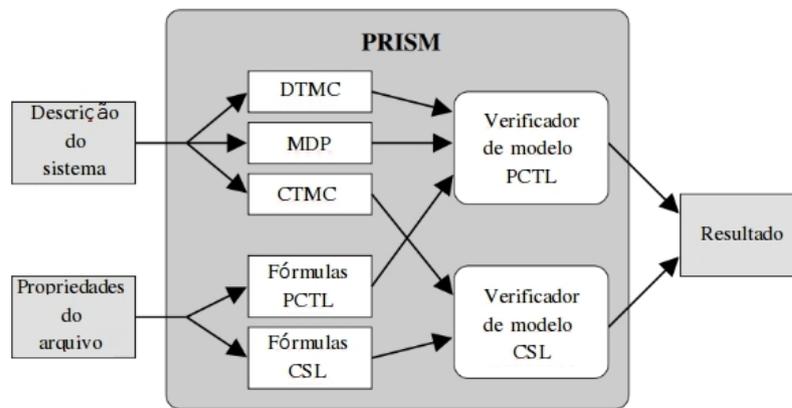
Esta propriedade expressa que, com uma probabilidade maior do que 0,8, em pelo

menos uma execução futura, pelo menos um dos líderes dos pelotões  $A$  ou  $B$  conseguirá detectar a presença do vizinho.

Caso a verificação dessa propriedade resulte em falso, o PRISM também pode gerar um contraexemplo, que é um caminho de execução específico do sistema que viola a propriedade verificada. Além disso, se a verificação for bem-sucedida, uma testemunha que descreve como a propriedade foi satisfeita pode ser gerada.

### 3.1.3 Método de verificação

Agora, fornecemos uma visão geral da funcionalidade do verificador de modelo PRISM e como é realizada a verificação, conforme indicado na Figura 3.3



**Figura 3.3** Estrutura do PRISM.

**Fonte:** Adaptado de Kwiatkowska, Norman e Parker (2006)

Durante o processo de verificação, inicialmente examinamos a descrição do modelo, que é a representação abstrata do sistema que queremos analisar. Em seguida, com base nessa descrição do modelo, construímos uma representação interna do modelo probabilístico. Nessa representação interna, calculamos os estados que podem ser alcançados e eliminamos aqueles que não são acessíveis. Em nosso trabalho, utilizamos o modelo DTMC (*Discrete Time Markov Chains*).

O modelo DTMC, constitui uma representação em que os estados denotam diferentes configurações possíveis do sistema. As transições entre esses estados ocorrem em intervalos discretos de tempo e possuem probabilidades associadas a elas. Esses modelos capturam processos estocásticos discretos, nos quais a distribuição de probabilidade dos estados futuros depende exclusivamente do estado presente, como destacado por Filieri, Ghezzi e Tamburrelli (2011).

Em seguida, um algoritmo de verificação de modelos é utilizado no modelo. Em alguns casos, como para propriedades que incluem um limite de probabilidade (PCTL, por exemplo), o PRISM simplesmente relata um resultado verdadeiro / falso, indicando se cada propriedade é ou não satisfeita pelo modelo atual. Também é possível obter um contraexemplo para a verificação da validade das propriedades.

A Figura 3.4 mostra uma captura de tela da interface gráfica do usuário da ferramenta PRISM, ilustrando os resultados de um experimento de verificação de modelo sendo plotado em um gráfico. Como alternativa, toda a funcionalidade de verificação de modelo também está disponível em uma versão em linha de comando da ferramenta. Nessa figura, é possível observar duas abordagens distintas para verificar um programa: identificando um resultado falso para uma propriedade e gerando um gráfico com os resultados de recompensas.

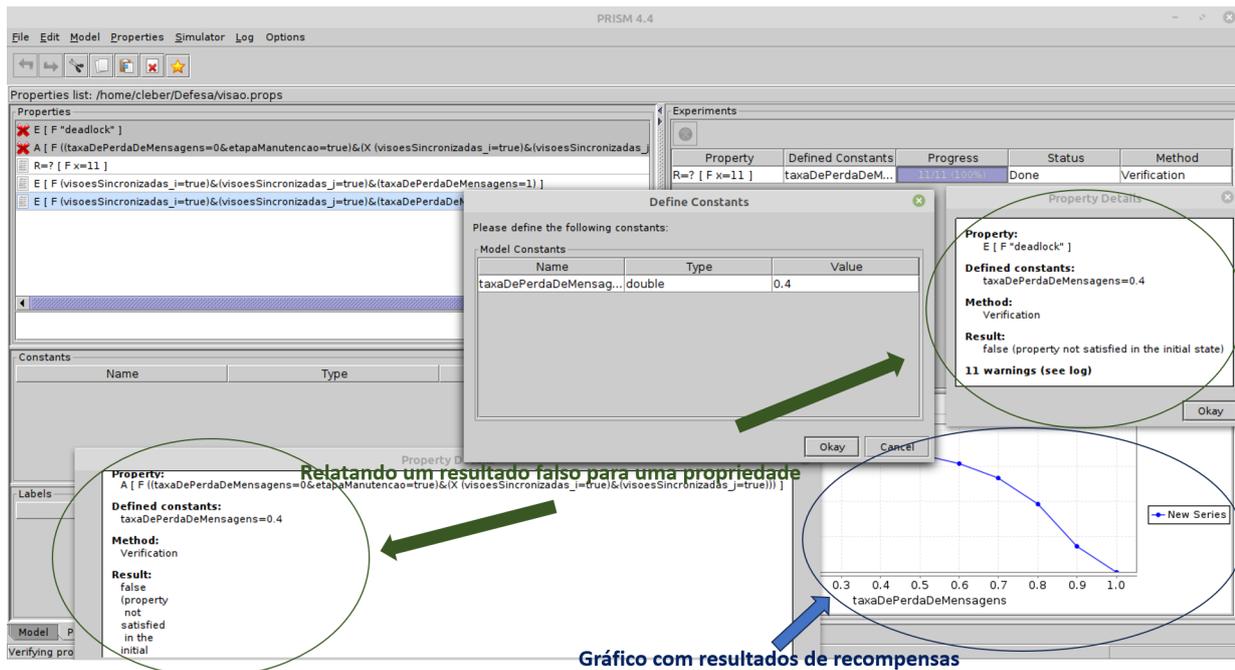


Figura 3.4 Um *screenshot* da interface gráfica do PRISM

## 3.2 CONCLUSÃO

A verificação probabilística de modelos é uma técnica formal e automática utilizada para modelar e analisar sistemas estocásticos. No próximo capítulo, será realizada a modelagem e verificação do protocolo por meio do verificador de modelos PRISM, com o objetivo de garantir a correção do protocolo.

## **PROTOCOLO BÁSICO PARA COMUNICAÇÃO ENTRE PELOTÕES**

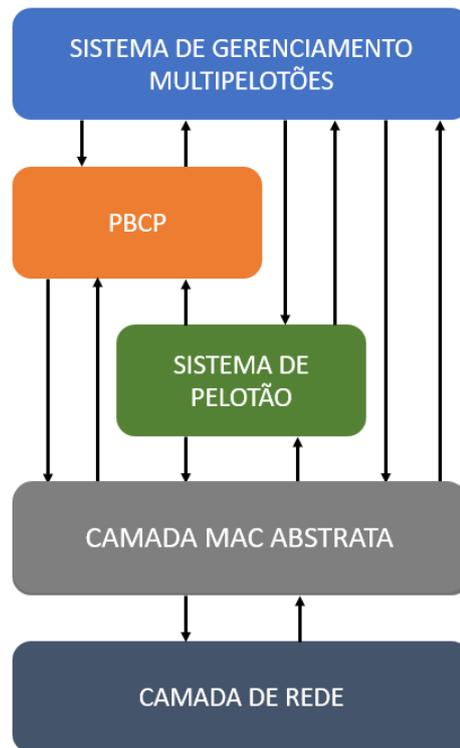
Neste capítulo, apresentamos o PBCP, Protocolo Básico para Comunicação entre Pelotões. Este protocolo tem um comportamento (envio e recebimento de mensagens) simples o suficiente de maneira a possibilitar uma análise efetiva dos resultados referentes aos tempos limite de progresso para entrega de mensagens durante o processo de detecção de vizinhança. Além disso, o método utilizado para detecção de vizinhança pode ser facilmente adaptado a diferentes protocolos aplicados às VANETs. O processo de detecção de vizinhança retrata a comunicação e gerenciamento de múltiplos pelotões sem perda de generalidade, já que pode ser aplicado na comunicação e gerenciamento entre dois veículos.

A arquitetura projetada para a utilização do PBCP nesta dissertação é estruturada em cinco camadas: camada de rede, camada MAC abstrata (explicada no Capítulo 2), camada do sistema de pelotão, camada do PBCP e a camada do sistema de gerenciamento multipelotões, conforme mostrado na Figura 4.1.

### **4.1 CAMADA DE REDE**

A camada de rede desempenha um papel fundamental no gerenciamento de propriedades do mundo real, incluindo o comportamento da parte física da rede. Neste contexto, assumimos que esta camada fornece uma interface que captura a comunicação do canal de comunicação sem fio, abrangendo a mobilidade, interferência do sinal de comunicação, colisão de mensagens e outros aspectos. Nesta dissertação consideramos que a mobilidade dos veículos, bem como as outras características do mundo real, é modelada como algo completamente encapsulado dentro da camada de rede e independente do comportamento das outras camadas da arquitetura.

Esta camada desempenha um papel crucial na decisão de encaminhamento de dados, fazendo com que as mensagens sejam entregues aos destinatários. A precisão das informações contidas nas mensagens é de extrema importância, uma vez que afeta diretamente o processo de detecção de vizinhança apresentado neste trabalho.



**Figura 4.1** Arquitetura do sistema de comunicação em camadas

## 4.2 CAMADA MAC ABSTRATA

A camada MAC abstrata utilizada na arquitetura do sistema de comunicação deste trabalho possui interface entre todas as camadas. Os veículos interagem com a camada MAC por meio do envio e recebimento de mensagens em difusão. Essa camada estabelece os tempos limite de progresso para a entrega das mensagens entre os líderes de pelotão durante o processo de detecção de vizinhança. Ela desempenha um papel essencial na mediação da troca de mensagens entre os veículos e a rede, permitindo que as mensagens sejam entregues aos destinatários apropriados. A camada MAC abstrata se integra à rede por meio da interface da camada de rede.

## 4.3 SISTEMA DE PELOTÃO

O sistema de pelotão considerado é composto por um conjunto finito, desconhecido e dinâmico de veículos dotados de sistema computacional embarcado e de um sistema operacional de tempo real, sobre os quais executam tarefas concorrentes capazes de produzir resultados dentro de prazos conhecidos. Cada veículo possui uma identificação única e é municiado por GPS, *Global Positioning System* e outros sensores que lhe permitem conhecer sua posição, velocidade e aceleração a cada instante. Os relógios físicos dos veículos estão sincronizados por GPS e possíveis desvios entre os relógios dos mesmos não são significantes para as aplicações veiculares.

Os veículos interagem diretamente entre si via troca de mensagens através de canais

de comunicação sem fio. Os canais de comunicação entre os veículos são síncronos e não confiáveis, existindo um tempo máximo para a transmissão da mensagem e a possibilidade de perda de mensagens. Quando as mensagens não são perdidas, estas são recebidas na mesma ordem em que são enviadas.

É considerado que o sistema de pelotão permite aos veículos que o compõe terem uma visão sempre atualizada de todos os membros do pelotão. O primeiro veículo à frente de cada pelotão é designado como o líder e determina as ações a serem seguidas por todos os membros. Apenas o líder é responsável por gerenciar as solicitações de entrada e de saída do pelotão, mantendo sempre uma visão atualizada dos veículos que lhe permite ter uma visão confiável e atualizada de todos os membros de seu grupo.

Quando o líder deixa o pelotão, um novo veículo é designado como líder e os demais como seus seguidores. Durante esta operação é possível que um mesmo pelotão se subdivida em pelotões menores com líderes distintos. Eventualmente estes líderes podem também negociar a união com outros líderes para aumentar o tamanho do pelotão quando for o caso. Dentro do contexto descrito, enumeramos os seguintes requisitos para o sistema de pelotão:

- **[R-001]** O pelotão deverá possuir um mecanismo que permita a inserção e desassociação de veículos de modo que o maior número possível de veículos seja acomodado em um mesmo pelotão;
- **[R-002]** Cada pelotão deverá ser identificado com um endereçamento único para que sejam possíveis ações cooperativas de controle e gerência dos pelotões;
- **[R-003]** Deverá existir um controle de tempo para consolidar as informações do pelotão relacionadas à sua composição (quantidade de membros), de maneira que os pelotões possam responder às mudanças em tempo real das demandas do tráfego.
- **[R-004]** A visão fornecida pelo sistema de pelotão deverá, necessariamente, estar disponível para uso de um serviço do gerenciamento de multipelotões.
- **[R-005]** O sistema de pelotão deverá dar suporte a veículos autônomos e semi-autônomos. Além disso, deverá ser capaz de passar o controle da direção do veículo ao motorista em um tempo adequado quando este não conseguir estabelecer uma comunicação com o veículo líder durante um determinado tempo. Neste caso a informação do veículo membro sairá da visão;
- **[R-006]** O sistema de pelotão deve oferecer escalabilidade suficiente sem comprometer a confiabilidade e fidelidade das informações trocadas no grupo e entre líderes de grupos;

Para maximizar os benefícios da utilização de pelotões, deseja-se que o maior número de veículos possível seja acomodado em um mesmo pelotão. Entretanto, se a quantidade de veículos aumenta demasiadamente ou se os veículos se distanciam uns dos outros, por exemplo, é possível que seja necessário que o pelotão se subdivida em pelotões menores. Do mesmo modo, pelotões devem ser unificados quando for possível.

## 4.4 PROTOCOLO BÁSICO PARA COMUNICAÇÃO ENTRE PELOTÕES

O comportamento do Protocolo Básico para Comunicação entre Pelotões está descrito através de seis requisitos: R-001 a R-006:

- [R-001] O protocolo deve possibilitar a um líder de pelotão ser capaz de detectar a presença de outros pelotões vizinhos através da troca de mensagens entre líderes, mesmo quando o canal de comunicação não for confiável;
- [R-002] A periodicidade e o tempo das mensagens enviadas pelos líderes devem atender ao prazo estabelecido pelo sistema de gerenciamento de múltiplos pelotões;
- [R-003] O protocolo deve ser capaz de suportar atrasos nas mensagens, levando em consideração o tempo limite para o progresso das manobras de divisão e união entre os multipelotões;
- [R-004] O protocolo deve ser resistente a perturbações na rede, tais como: mudanças de conectividades, atrasos e perdas de mensagens.
- [R-005] A percepção da vizinhança fornecida pelo PBCP, deverá ser correta (em relação à composição dos pelotões vizinhos) e atualizadas (no sentido de que apenas a informação mais recente dos vizinhos é disponibilizada);
- [R-006] As visões fornecidas de cada pelotão deverão, necessariamente, estar sempre disponíveis para uso do protocolo PBCP.

### 4.4.1 Descrição do Algoritmo

O PBCP é dividido em duas tarefas (ver Algoritmo 1). A Tarefa *T1* é responsável pelo envio de mensagens periódicas. Inicialmente, para evitar que diferentes visões de um mesmo pelotão sejam armazenadas simultaneamente, é verificado se o endereço do líder que enviou a mensagem pertence ao conjunto *visoesConhecidas*. Caso pertença, a visão do líder será retirada do conjunto *visoesConhecidas* (linhas de 9 a 12). Antes do envio da mensagem, as visões contidas no conjunto *mensagensRecebidas* são adicionadas em *visoesConhecidas*, exceto a visão do líder remetente (linhas 15, 16). Em seguida, uma mensagem é criada contendo o endereço do líder remetente, sua visão e todas as visões conhecidas até aquele momento (linha 19). Por fim, a mensagem é difundida para os líderes vizinhos (linha 20). Essas ações são repetidas a cada intervalo de tempo definido por *periodo*, somadas a um tempo gerado aleatoriamente por uma função *tempoAleatorio()* usada para minimizar a probabilidade de dois ou mais veículos terem o mesmo tempo entre os envios de mensagens (linha 21).

A Tarefa *T2* é responsável por receber as mensagens enviadas por outros líderes e realiza a manutenção das visões, armazenando as informações dos pelotões vizinhos no conjunto *vizinhosDetectados*. Durante um intervalo de tempo predefinido, as mensagens recebidas são armazenadas no conjunto *mensagensRecebidas* (linha 27). Quando o tempo limite é alcançado (linha 28), é iniciado o processo de manutenção das mensagens.

**Algoritmo 1: PBCP****Entrada:**  $idLider$ ,  $visao$ ,  $tempoLimite$ ,  $periodo$ ,  $cronometro$ **Saída:**  $vizinhosDetectados$ 

```

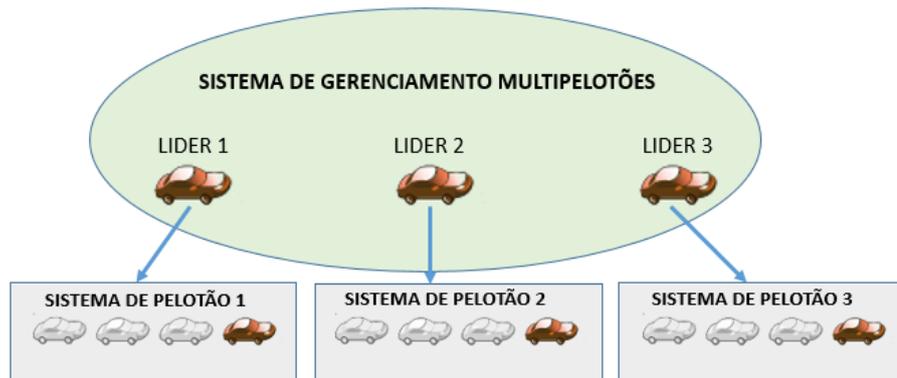
1 início
2   vizinhosDetectados  $\leftarrow \emptyset$ 
3   visoesConhecidas  $\leftarrow \emptyset$ 
4   visoesTemporarias  $\leftarrow \emptyset$ 
5   mensagensRecebidas  $\leftarrow \emptyset$ 
6   mensagensTemporarias  $\leftarrow \emptyset$ 
7   Tarefa T1: início
8     enquanto True faça
9       para cada  $m$  em mensagensRecebidas faça
10        para cada  $v$  em visoesConhecidas faça
11          se  $v.idLider = m.idLider$  então
12            | visoesConhecidas  $\leftarrow visoesConhecidas \setminus \{v\}$ 
13          fim
14        fim
15      se  $m.idLider \neq idLider$  então
16        | visoesConhecidas  $\leftarrow visoesConhecidas \cup \{m.visao\}$ 
17      fim
18    fim
19    mensagem  $\leftarrow (idLider, visao, visoesConhecidas, cronometro)$ 
20    enviarMensagem(mensagem)
21    sleep(periodo + tempoAleatorio())
22  fim
23  fim
24  Tarefa T2: início
25    tempAux  $\leftarrow 0$ 
26    enquanto True faça
27      mensagensRecebidas  $\leftarrow mensagensRecebidas \cup \{receberMensagem()\}$ 
28      se cronometro  $\geq tempAux + tempoLimite$  então
29        mensagensTemporarias  $\leftarrow mensagensRecebidas$ 
30        tempAux  $\leftarrow cronometro$ 
31        para cada  $m$  em mensagensTemporarias faça
32          se  $m.idLider \neq idLider$  então
33            para cada  $v$  em  $m.visoesConhecidas$  faça
34              se  $v = visao \wedge m.cronometro \geq tempAux - tempoLimite$ 
35                então
36                  | visoesTemporarias  $\leftarrow visoesTemporarias \cup \{m.visao\}$ 
37                fim
38            fim
39          fim
40        vizinhosDetectados  $\leftarrow visoesTemporarias$ 
41        visoesTemporarias  $\leftarrow \emptyset$ 
42        mensagensRecebidas  $\leftarrow \emptyset$ 
43        mensagensTemporarias  $\leftarrow \emptyset$ 
44      fim
45    fim
46  fim
47 fim

```

As informações contidas no conjunto *mensagensRecebidas* são copiadas para *mensagensTemporarias* (linha 29). Para cada mensagem recebida, é verificado se dentro do conjunto *visoesConhecidas* existe a visão do líder que recebeu a mensagem (linhas 31 a 34) e se esta corresponde ao último ciclo de envio de mensagens. Se as condições forem verdadeiras, a visão do líder que enviou a mensagem é inserida no conjunto *visoesTemporarias* (linha 35). Ao final da verificação, todas as visões aprendidas são copiadas para *vizinhosDetectados* (linha 40). O objetivo é armazenar apenas visões atualizadas, permitindo que os líderes possam realizar ações de gerenciamento com múltiplos pelotões.

#### 4.5 SISTEMA DE GERENCIAMENTO MULTYPELOTÕES

Neste trabalho, a camada de gerenciamento multipelotões é proposta como uma hipótese, sendo responsável por utilizar informações de detecção de vizinhança fornecidas pelo PBCP para coordenar manobras de coordenação entre pelotões. Periodicamente, o sistema de gerenciamento multipelotões utiliza as informações das visões dos pelotões sobre seus vizinhos, detectadas pelo PBCP, com o intuito de possibilitar a fusão de pelotões distintos, conforme ilustrado na Figura 4.2. Importante ressaltar que a configuração detalhada desse sistema não é especificada neste trabalho.



**Figura 4.2** Estrutura do relacionamento entre o sistema de pelotão e o sistema de gerenciamento multipelotões.

Consideramos que o sistema de gerenciamento multipelotões suporta duas manobras básicas ou elementares: fusão e divisão. Na fusão, dois pelotões, viajando na mesma pista, se fundem em um grande pelotão. Na divisão, um pelotão (com pelo menos dois veículos) se separa para formar dois pelotões menores. Estas manobras são executadas pelo sistema de gerenciamento multipelotões utilizando as informações fornecidas pelos líderes, a saber, a visão de cada pelotão.

Os limites temporais para a disseminação das mensagens são finitos e conhecidos pelos veículos líderes que participam do processo de troca de mensagens. Além disto, o canal permite envios de mensagens FIFO, *First In First Out*, ou seja, não há mudança de ordem nas transmissões, então todas as mensagens serão entregues em qualquer destino, na mesma ordem em que foram enviadas. Contudo, como o canal não é confiável, está

sujeito a falhas por omissão, no qual uma mensagem enviada pelo remetente poderá nunca chegar ao destinatário.

Assume-se que os veículos organizados em múltiplos pelotões compartilham uma mesma rodovia. A quantidade de pelotões é finita, desconhecida e varia ao longo do tempo. Cada pelotão é composto por um número variável de veículos. Um mesmo veículo não pode participar de mais de um pelotão ao mesmo tempo.

Dentro do contexto descrito, enumeramos os seguintes requisitos que deverão ser contemplados para o sistema de gerenciamento multipelotões:

- [R-001] A fusão entre dois pelotões deverá ser iniciada por um dos líderes dos pelotões;
- [R-002] O sistema de gerenciamento multipelotões deverá dar suporte à divisão de um pelotão. A manobra de divisão de pelotão é sempre iniciada pelo líder do pelotão;
- [R-003] A comunicação entre os líderes deverá ser realizada veículo a veículo, não necessitando de uma infraestrutura definida.
- [R-004] O sistema deve operar dentro de limites temporais finitos e conhecidos para a disseminação das mensagens, garantindo a coordenação eficaz entre os pelotões;
- [R-005] Sempre que um veículo membro do pelotão desejar sair, necessariamente, antes que ocorra a desassociação com o pelotão, o veículo membro que está imediatamente atrás será eleito como líder até que o veículo que deseja sair do pelotão saia completamente, quando os dois líderes farão novamente a fusão tornando-se um pelotão.

#### 4.6 MÉTODO DE DETECÇÃO DE VIZINHANÇA UTILIZADO NO PBCP

Esta seção descreve o método utilizado para descoberta de vizinhança em redes veiculares *ad hoc* e como ele pode ser aplicado na detecção de vizinhança em uma rede composta por líderes de pelotões de veículos, por meio do protocolo básico de comunicação entre grupos de pelotões, o PBCP, desenvolvido neste trabalho. O objetivo deste protocolo simples de detecção de vizinhança é detectar pelotões próximos para estabelecer uma comunicação cooperativa. A Seção 4.2 aborda a influência que os tempos limite de progresso exercem sobre a comunicação entre os líderes de pelotões durante o processo de detecção de vizinhança.

O protocolo básico de comunicação entre pelotões desenvolvido neste trabalho e utilizado para a simulação no processo de detecção de vizinhança é implementado sobre uma camada de controle de acesso ao meio abstrata que fornece tempos limite de progresso para entrega de mensagens, abstraindo os detalhes de detecção de colisões, contenção e escalonamento. Seguimos a especificação de uma camada MAC abstrata apresentada em (KUNH; LYNCH; NEWPORT, 2009). Essa abordagem modular torna o algoritmo mais

fácil de ser projetado, compreendido e verificado. No entanto, lidar com padrões de mobilidade arbitrários enquanto tenta maximizar o tempo em que as conexões permanecem ativas ainda é um problema complexo.

PBCP é um protocolo de detecção de vizinhança simples, projetado para ser aplicado em redes compostas por líderes de pelotões de veículos. Ele permite que os líderes de pelotões detectem a presença de pelotões próximos para que possam estabelecer uma comunicação cooperativa. No entanto, a detecção de vizinhança pode ser afetada pela presença de obstáculos físicos, como prédios ou árvores, ou por variações no terreno, o que pode dificultar a comunicação entre os líderes de pelotões.

A detecção de vizinhança refere-se essencialmente a cada líder de pelotão manter-se ciente de seus vizinhos ativos dinamicamente a qualquer momento. Como os veículos se movem constantemente ao longo do tempo, os vizinhos podem mudar de tempos em tempos. Para um determinado líder de pelotão, quando outro pelotão entra em seu alcance de comunicação, ele deve perceber a aproximação deste novo vizinho. Quando um de seus vizinhos sai de seu alcance de comunicação, o líder deve estar ciente da saída deste da vizinhança.

Uma das vantagens da detecção de vizinhança é que ela permite que os líderes de pelotões compartilhem informações de tráfego em tempo real. Por exemplo, se um líder detectar um engarrafamento à frente, ele pode transmitir essa informação para os pelotões próximos, permitindo que eles ajustem sua rota e evitem o congestionamento. Além disso, a detecção de vizinhança é fundamental para a segurança na estrada, pois permite que os líderes detectem a presença de outros pelotões próximos e evitem colisões.

Consideramos que a detecção de vizinhança ocorre por meio da troca de mensagens entre líderes de pelotões. Quando cada par de líderes possui informações atualizadas um do outro, a detecção da vizinhança ocorre, o que significa que apenas os líderes que estão próximos uns dos outros e dentro do alcance de comunicação transmitem e recebem informações entre si. Para isso, periodicamente, o líder de cada pelotão compartilha sua visão atual com os demais líderes em sua vizinhança – isto é, aqueles líderes com os quais podem se comunicar diretamente (em um único salto). As visões recebidas por um líder de seus líderes vizinhos são armazenadas em um conjunto local de visões conhecidas, denominado "visões conhecidas". Nas trocas de mensagens entre os líderes, cada um deles compartilha não apenas sua visão atual, mas também seu conjunto local de visões conhecidas. Dessa forma, uma mensagem  $m$  é definida por:  $m = (idLider, visao, visoesConhecidas)$ , onde  $idLider$  representa o identificador do líder remetente,  $visao$  é a visão do líder remetente e  $visoesConhecidas$  é o conjunto de visões conhecidas do líder remetente.

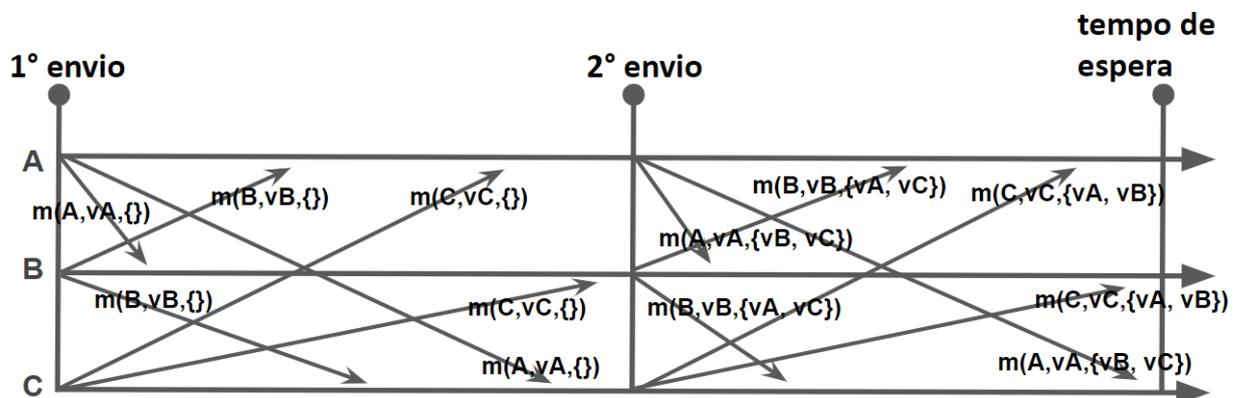
Uma visão é definida pela tupla  $visao = (idLider, versao, membros)$ , em que  $idLider$  representa o identificador do líder responsável pela visão. A informação da  $versao$  é modificada a cada atualização da visão. Por sua vez,  $membros$  é uma lista que contém o identificador de cada veículo pertencente ao pelotão, incluindo o líder. Cada vez que o pelotão sofre uma alteração em sua composição, uma nova visão é criada.

Nas Figuras 4.3 e 4.4, temos um exemplo da operação do protocolo de descoberta de vizinhança proposto com três pelotões  $A$ ,  $B$  e  $C$ . Na representação, o tempo progride da esquerda para a direita e as setas diagonais representam o envio de mensagens em difusão realizados pelos líderes. A detecção de vizinhança ocorre em tempos regulares chamados

de tempo limite.

Durante o primeiro envio de mensagens (Figura 4.3), como nenhum líder possui qualquer informação sobre os demais da vizinhança, o conjunto de visões conhecidas está vazio. As mensagens são recebidas durante o intervalo entre o primeiro e o segundo envio de mensagem. Ao receberem as mensagens os líderes inserem as visões dos líderes remetentes no conjunto *visoesConhecidas*.

No segundo envio (Figura 4.3), cada líder envia a sua visão atual e o conjunto de visões conhecidas até aquele momento. Ao encerrar o tempo limite de espera para detecção de vizinhança cada líder verifica se a sua visão está contida no conjunto de visões conhecidas dos líderes remetentes. Caso esteja, o pelotão ao qual o líder remetente pertence é considerado como um vizinho detectado pelo líder que recebeu a mensagem. No exemplo mostrado, todos os pelotões foram detectados como vizinhos.

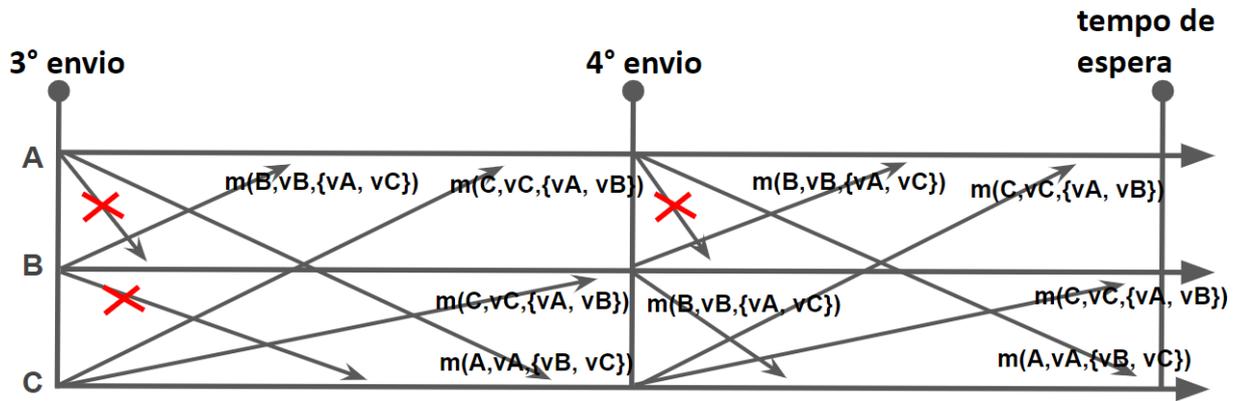


**Figura 4.3** Detecção de vizinhança - Canal de comunicação sem perda de mensagens

A Figura 4.4 mostra o comportamento do serviço de detecção de vizinhança em um cenário em que o canal de comunicação está sujeito a perda de mensagens. Após o terceiro envio algumas mensagens são perdidas, não alcançando todos os líderes de pelotão. Assim, os líderes dos pelotões *B* e *C* não recebem as mensagens dos líderes *A* e *B*, respectivamente.

Após o quarto envio, apenas o líder *B* não recebe mensagem do líder *A*. Ao final tempo limite, o líder *C* considera os pelotões *A* e *B* como vizinhos detectados. Entre os líderes dos pelotões *A* e *B*, apenas o líder *A* considera o pelotão *B* como vizinho detectado. Como o conjunto *visoesConhecidas* nunca é apagado, sempre mantendo a última visão recebida de cada líder remetente, mesmo que apenas uma mensagem seja recebida, se esta contiver a visão do líder que recebeu a mensagem, o líder irá considerar o grupo ao qual o líder remetente faz parte como um vizinho detectado.

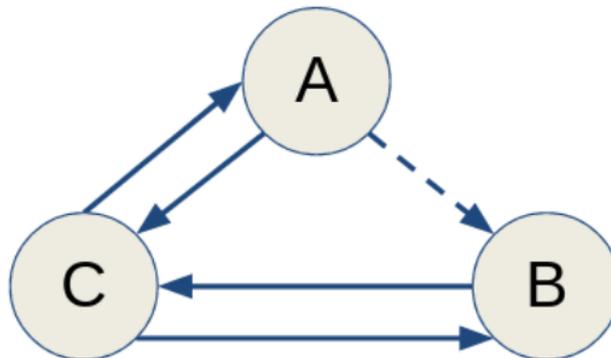
Consideramos dois tipos de vizinhanças detectadas: estável e instável. A primeira ocorre quando, após o cruzamento das vizinhanças detectadas por cada líder, for possível identificar pares de líderes que possuem informações atualizadas um do outro. Isto parece ser útil no sentido de que uma decisão relacionada a fusão de pelotões, realizada por um sistema de gerenciamento de múltiplos pelotões, será tomada com muito mais segurança, já que ambos os líderes estabeleceram uma vizinhança estável, ou seja, possuem



**Figura 4.4** Detecção de vizinhança - Canal de comunicação com perda de mensagens

informações atualizadas sobre a composição dos pelotões um do outro. A vizinhança instável, no entanto, ocorre quando apenas um dos dois considera o outro como um vizinho detectado.

A Figura 4.5 ilustra as definições de vizinhanças estáveis e instáveis. Todos os nós



**Figura 4.5** Tipos de vizinhança

(correspondendo aos grupos de veículos) na figura estão corretos, no sentido de que todos os nós existentes sempre estarão disponíveis para enviar e receber mensagens. As setas sólidas representam o sucesso na entrega de mensagem pelo par de líderes estabelecendo uma vizinhança estável entre estes. A linha tracejada, no entanto, indica que apenas um dos líderes de pelotão conseguiu detectar o vizinho, caracterizando assim a vizinhança como instável.

## 4.7 VERIFICAÇÃO DO PBCP

No processo de verificação do PBCP, empregamos a ferramenta de modelagem e verificação formal PRISM. Nosso propósito é verificar o comportamento da detecção de vizinhança, entre dois líderes de pelotão, considerando distintas taxas de perda de mensagens.

Apresentamos trechos da modelagem das Tarefas T1 e T2, bem como a etapa de manutenção, a modelagem completa é descrita no Apêndice C. Consideramos que dois líderes ( $i$  e  $j$ ) enviam mensagens em difusão em períodos regulares de tempo.

O trecho do modelo representado abaixo descreve a ação de enviar mensagens (Tarefa T1). A fórmula *turnoDoLider* determina qual líder está ativo em um dado momento. Esta pode ser verdadeira para o líder  $i$  ou  $j$ .

A variável *visoesConhecidas\_i*' (linha 43) é atualizada com base nos dados da *visaoRecebida\_i*. Essa expressão modela a atualização das visões recebidas pelo líder no seu conjunto *visoesConhecidas*, conforme indicado na linha 16 do Algoritmo 1.

A transição de líder ocorre após a atualização da variável *mudancaDoTurnoDoLider* (linha 44), que é realizada utilizando aritmética modular. Inicialmente definida como 1 (representando o líder  $i$ ), quando essa variável é incrementada, passa a valer 2 (representando o líder  $j$ ), e ao atingir o valor 2, a variável retorna a 1, completando assim o ciclo de alternância entre os líderes  $i$  e  $j$ .

```
40 - //TAREFA T1 - RESPONSÁVEL PELO ENVIO PERIÓDICO DE MENSAGENS
41 -   [] turnoDoLider & tarefa=enviar
42 -   ->
43 -   ( visao_i '=dadosDaVisao ) & ( visoesConhecidas_i '=visaoRecebida_i ) &
44 -   ( mudancaDoTurnoDoLider '=mudancaDoTurnoDoLider+1 );
```

O trecho do modelo correspondente às linhas 47 a 64 descreve a ação de receber mensagens (Tarefa T2). Se as três condições da linha 47 forem verdadeiras, o sistema realizará uma série de atualizações representadas por uma escolha probabilística. O comportamento do sistema ao receber mensagens é modelado de forma probabilística para refletir incertezas no processo. Com uma probabilidade de  $(1 - taxaDePerdaDeMensagens)$  (linhas 49), as seguintes ações ocorrem: a variável *cronometro\_i*' é incrementada em 1, indicando o tempo decorrido; a variável *visaoRecebida\_i*' será atualizada com a informação da *visao\_j* e *visoesConhecidasRecebidas\_i*' será atualizada com a informação da variável *visoesConhecidas\_j*; essas duas atualizações correspondem aos valores da *visao* e *visoesConhecidas* do líder remetente, respectivamente. Com uma probabilidade de *taxaDePerdaDeMensagens*, conforme indicado na linha 54, as variáveis *visaoRecebida\_i*' e *visoesConhecidasRecebidas\_i*' não serão atualizadas, indicando que houve perda de mensagens.

```
46 - //TAREFA 2 - RESPONSÁVEL PELO RECEBIMENTO DE MENSAGENS
47 -   [] turnoDoLider & tarefa=receber & cronometro_i<tempoLimite
48 -   ->
49 -   (1-taxaDePerdaDeMensagens) : ( cronometro_i '=cronometro_i+1)
50 -   & ( mudancaDoTurnoDoLider '=mudancaDoTurnoDoLider+1) &
51 -   ( visaoRecebida_i ' = visao_j ) &
52 -   ( visoesConhecidasRecebidas_i '=visoesConhecidas_j)
53 -   +
54 -   ( taxaDePerdaDeMensagens ) : ( cronometro_i '=cronometro_i+1) &
55 -   ( mudancaDoTurnoDoLider '=mudancaDoTurnoDoLider+1);
56 -
57 -   [] turnoDoLider & tarefa=receber & cronometro_i=tempoLimite &
58 -   etapaManutencao=false
59 -   ->
```

```

60 - (1-taxaDePerdaDeMensagens) : (etapaManutencao!=true) &
61 - (visaoRecebida_i' = visao_j) &
62 - (visoesConhecidasRecebidas_i'=visoesConhecidas_j)
63 - +
64 - (taxaDePerdaDeMensagens) : (etapaManutencao!=true);

```

O trecho do modelo correspondente às linhas 57 a 58 também especifica o recebimento de mensagens. Contudo, o valor da variável *cronometro\_i* é igual a constante *tempoLimite*. Isso significa que, mesmo quando o tempo para a detecção de mensagens é alcançado, ainda é possível que alguma mensagem alcance um pelotão vizinho, e isso é representado nessa parte do modelo. Neste caso, o sistema realizará uma série de atualizações representadas por uma escolha probabilística. Com probabilidade  $1 - taxaDePerdaDeMensagens$  (linha 60) além das variáveis *visaoRecebida\_i'* e *visoesConhecidasRecebidas\_1'*, a variável *etapaManutencao'* é atualizada com valor booleano *true*. Essas atualizações indicam que o líder recebeu a mensagem do líder de pelotão vizinho.

O segmento do modelo abaixo descreve a etapa de manutenção. Se a variável booleana *etapaManutencao* possui o valor *true*, e *visoesConhecidasRecebidas\_i* for igual a *visao\_i* (linhas 67 e 68) a variável *vizinhancaDetectada'* será atualizada como *true*, indicando que o líder identificou o seu vizinho. Esta parte da modelagem refere-se a linha 34 do Algoritmo 1 que verifica se é encontrado no conjunto *visoesConhecidas* a visão *v* do líder remetente ( $v = visao$ ).

```

66 - //ETAPA DE MANUTENCAO
67 - [] turnoDoLider & etapaManutencao=true &
68 - visoesConhecidasRecebidas_i=visao_i
69 - ->
70 - (vizinhanca_i' = true) & (etapaManutencao!=false) &
71 - (mudancaDoTurnoDoLider!=mudancaDoTurnoDoLider+1) &
72 - (vizinhancaDetectada!=true) & (cronometro_i'=0);
73 -
74 - [] turnoDoLider & etapaManutencao=true &
75 - visoesConhecidasRecebidas_i!=visao_i
76 - ->
77 - (vizinhanca_i' = false) & (etapaManutencao!=false) &
78 - (mudancaDoTurnoDoLider!=mudancaDoTurnoDoLider+1) &
79 - (vizinhancaDetectada!=true) & (cronometro_i'=0);

```

## 4.8 VERIFICAÇÃO DO MODELO EM PRISM

Para executar a verificação do modelo, utilizamos um computador com processador intel Core i5 7th Gen. O sistema operacional foi o Linux Mint 18.3 Sylvia, com 8GB de RAM. Na especificação das propriedades, foi utilizado um subconjunto da lógica de árvore de computação probabilística (PCTL, *Probabilistic Computation Tree Logic*) (MARTA; GETHIN; PARKER, 2012a).

A verificação das propriedades *liveness* e *safety* no modelo do PBCP em PRISM é utilizada nesta dissertação para garantir a correção do sistema, conforme indicado na Tabela 4.1. As propriedades *liveness* indicam que, em cenários com perda de mensagens, o líder detectará em algum momento seu vizinho, desde que não ocorra perda de 100% das mensagens. Da mesma forma, considerando um canal de comunicação no qual todas as mensagens alcance o destino,

todos os líderes conseguirão detectar sua vizinhança.

No que diz se refere à propriedade *safety*, conforme evidenciado na tabela. pode-se notar que é garantido que o sistema não entra em estado de *deadlock*. A outra propriedade *safety* aborda a situação em que o líder não detecta seu vizinho em nenhuma situação em que a perda de mensagens alcance 100%.

**Tabela 4.1** Propriedades verificadas do PBCP

Especificação formal	Resultados em PRISM	Propriedade	Conclusão
$E[F \text{ "deadlock"}]$	false	Propriedade <i>safety</i>	Ausência de <i>deadlock</i>
$E[F (\text{vizinhanca}_i=\text{true}) \ \& \ (\text{vizinhanca}_j=\text{true}) \ \& \ (\text{taxaDePerdaDeMensagens}=1)]$	false	Propriedade <i>safety</i>	Em nenhuma execução futura, o líder irá detectar o seu vizinho em uma rede com 100% de perda de mensagens
$E[F (\text{vizinhanca}_i=\text{true}) \ \& \ (\text{vizinhanca}_j=\text{true}) \ \& \ (\text{taxaDePerdaDeMensagens}<1)]$	true	Propriedade <i>liveness</i>	Em alguma execução futura, em um cenário onde a perda de mensagens é menor que 100%, o líder irá detectar o vizinho
$A[ \ F \ ((\text{taxaDePerdaDeMensagens}=0 \ \& \ \text{atualizaVizinhanca}=\text{true}) \ \& \ (X (\text{vizinhanca}_i=\text{true}) \ \& \ (\text{vizinhanca}_j=\text{true}))) ]$	true	Propriedade <i>liveness</i>	Ao longo de todas as execuções, em um cenário ideal em que não há perda de mensagens, logo após a etapa de manutencao o líder detectará o seu vizinho

De acordo com os resultados da verificação, concluímos que o protocolo atende ao objetivo de detectar a vizinhança em ambientes multipelotão, conforme verificado na correção como está indicado na coluna conclusão da tabela.

## 4.9 CONCLUSÃO

Neste capítulo, apresentamos o protocolo básico de comunicação entre pelotões desenvolvido nesta dissertação. Introduzimos os conceitos básicos do funcionamento do protocolo através de uma arquitetura em cinco camadas. Por fim, mostramos o método de detecção de vizinhança do PBCP.

Além disso, para aumentar a confiança no protocolo apresentado nesta dissertação, empregamos a verificação de modelos probabilística a fim de verificar se o PBCP satisfaz propriedades requeridas. A verificação do modelo do protocolo foi realizada considerando o comportamento aleatório e probabilístico do canal de comunicação, como a perda de mensagens. O resultado da verificação indicou que o protocolo está em conformidade com os requisitos de detecção de vizinhança em um ambiente multipelotão.



## ANÁLISE DE TEMPOS LIMITE DE PROGRESSO PARA DETECÇÃO DE VIZINHANÇA

A detecção de vizinhança em uma rede veicular *ad hoc* é uma tarefa importante para garantir uma comunicação eficiente entre os veículos na rede. Neste trabalho, exploramos a detecção de vizinhança entre pelotões de veículos. Para que dois pelotões sejam considerados detectados como vizinhos é necessário que cada par de líderes de pelotão possua a informação atualizada da visão um do outro.

O tempo para que uma mensagem alcance todos os veículos da rede influi na detecção de vizinhança. A determinação deste tempo constitui um desafio já que os tamanhos das mensagens trocadas entre veículos, os períodos de tempo entre mensagens e o intervalo de tempo necessário para que os pelotões sejam considerados detectados influenciam diretamente na capacidade dos veículos detectarem sua vizinhança.

Neste sentido, situamos o nosso trabalho em relação a dois estudos que tratam sobre tempos limite de progresso para a detecção de vizinhança e um estudo que trata sobre a definição da condição de estabilidade em uma rede dinâmica. Apresentamos as análises dos resultados de experimentos através de quatro cenários. Os experimentos e os parâmetros utilizados nas simulações também foram baseados em trabalhos apresentados na literatura.

Os estudos relacionados à detecção de vizinhança em um ambiente de rede veicular *ad hoc* e os tempos limite necessários para que as mensagens sejam entregues de maneira a possibilitar que a detecção ocorra consideram muitas vezes uma camada abstrata de acesso ao meio. Esta camada abstrata abstrai as questões de baixo nível relacionadas ao canal de comunicação e seus efeitos, a limites de atrasos abstratos (KUHN; LYNCH; NEWPORT, 2009; CORNEJO; VIQAR; WELCH, 2014).

Os tempos limites de progresso determinados para a detecção de vizinhança estão relacionados ao tempo para que uma mensagem de um líder de grupo seja recebida pelos demais líderes vizinhos. Estes tempos são entregues pela camada MAC abstrata. Além disso, diferentemente do trabalho apresentado por Cornejo, Viqar e Welch (2014) em que são tratados aspectos relacionados à detecção de vizinhança mas sem considerar que os nós podem parar de transmitir ou receber mensagens, nós queremos identificar a quantidade de pelotões vizinhos detectados em uma rede cujo o canal de comunicação pode ser confiável ou não, i.e, sujeito a perdas e atrasos de mensagens. Dado o comportamento imprevisível das redes veiculares *ad hoc*, em que é possível que mensagens trocadas entre veículos colidam indeterminadamente, impossibilitando que as

mensagens sejam recebidas pelos destinatários, consideramos um comportamento probabilístico para as perdas de mensagens.

Outro importante desafio encontrado na literatura, relacionado à detecção de vizinhança, trata-se da identificação de algum tipo de estabilidade que garanta a detecção da vizinhança. Questões como a alta mobilidade, a largura de banda limitada, a interferência no canal de comunicação sem fio e o congestionamento tornam a estabilidade necessária. No trabalho de Arantes et al. (2010), a condição de estabilidade é analisada sob o ponto de vista de grupos. Um grupo é estável quando os nós móveis sempre podem se comunicar uns com os outros por meio de um canal de comunicação dinâmico, ou seja, enlaces que se modificam, dentro de um tempo limite.

## 5.1 ANÁLISE DOS TEMPOS LIMITE DE PROGRESSO PARA DETERMINAÇÃO DE VIZINHANÇA

A determinação dos tempos limite de progresso é realizada através de simulação. Para simular a rede veicular, utilizamos o *framework* VEINS, *VEhIcular Network Simulation* (SOMMER et al., 2019). O VEINS integra o SUMO (*Simulation of Urban MObility*) com o OMNET++ (*Objective Modular Network Testbed in C++*) (VARGA; HORNIG, 2010). SUMO é usado para simular a mobilidade dos veículos, enquanto que o OMNET++ é usado para simular a comunicação entre veículos.

Nas simulações, consideramos que os pelotões compartilham uma mesma rodovia, deslocando-se na mesma direção, porém em velocidades distintas. Além disso, a comunicação entre os líderes se dá pelo mesmo canal de comunicação empregado para a interação interna no pelotão de veículos.

Para determinar a quantidade de líderes nos experimentos e o período de entrega de mensagens, baseamo-nos nos trabalhos de Shakeel et al. (2015) e Amoozadeh et al. (2015). Esses estudos avaliam, respectivamente, a capacidade de entrega de mensagens periódicas e o tamanho das mensagens em uma rede veicular *ad hoc*, bem como a comunicação cooperativa entre pelotões de veículos. Shakeel et al. (2015) utiliza 4 veículos individuais para seus experimentos, sem variar a quantidade de veículos. No nosso trabalho, consideramos que a comunicação ocorre entre líderes de pelotões e variamos a quantidade de líderes de 2 a 10 para analisar como a variação influencia no tempo de entrega de mensagens aos destinatários. Cada pelotão é formado por um total de 8 veículos, o que inclui o veículo líder. Essa quantidade está alinhada com a recomendação do projeto Sartre, que estabelece um limite máximo de até 15 veículos para um pelotão de veículos (AMOOZADEH et al., 2015). Para estabelecer a comunicação entre pelotões, inicialmente o período de envio de mensagens é de 100 milissegundos, o mesmo utilizado por Amoozadeh et al. (2015).

Para estabelecer as restrições nos tempos limite de progresso dos experimentos tomamos como base três aplicações para VANET que utilizam o método de envio periódico de mensagens. Os tempos limites de progresso mostrados na Tabela 5.1 são os valores típicos para estes tipos de aplicações tais como aviso de colisões, gerenciamento de tráfego e anúncio e entretenimento (NGUYEN et al., 2019). A primeira está relacionada à segurança (como alerta de ponto cego, detecção pré-colisão). Este tipo de aplicação requer mecanismos de transmissão confiáveis e rápidos e, portanto, cada veículo deve transmitir periodicamente suas informações como a posição, a velocidade e a aceleração. A segunda está relacionada aos serviços de gerenciamento de tráfego que consistem em gerenciamento de interseção, aviso de atraso, prevenção de congestionamentos rodoviários e controle de cruzamento adaptativo cooperativo. A terceira aplicação está

relacionada aos serviços orientados ao usuário como anúncios e entretenimento para os usuários durante a viagem.

**Tabela 5.1** Requisitos de tempo relacionado ao envio de mensagem

Fonte: Adaptado de Nguyen et al. (2019)

Aplicação	Atraso no envio de mensagem (ms)	Alcance da transmissão (m)
Aviso de colisão de cooperação	100	50 - 300
Anúncio e entretenimento	500	0 - 90
Serviço de gerenciamento de tráfego	100	300

Em relação às mensagens, a capacidade nominal de transmissão de  $6Mbps$  está de acordo com o padrão IEEE 802.11p (ALMEIDA et al., 2018), que utilizamos neste trabalho. Utilizamos o padrão WAVE para a comunicação entre os líderes de pelotão. Dado o alto dinamismo da rede, consideramos cenários nos quais a taxa de perda de mensagens varia entre 0 e 90%. Para simplificação dos resultados as taxas 100% não são mostradas. Sendo que 0% representa uma comunicação confiável, em que não há interferências no sinal, e 100% indica que não houve detecção. O tempo de espera padrão é de 1000 milissegundos, o mesmo utilizado no trabalho de (SHAKEEL et al., 2015). A Tabela 5.2 apresenta um resumo dos parâmetros de rede usados na simulação.

**Tabela 5.2** Parâmetros de configuração de rede

Protocolo de comunicação de controle de acesso ao meio	IEEE 802.11p
Padrão de comunicação	WAVE
Taxa de transferência dos enlaces de comunicação sem fio	$6Mbps$
Tempo de espera para detecção da vizinhança	$1000ms$
Percentual de perda de mensagens do canal de comunicação	10 – 90%

Um tempo gerado aleatoriamente é adicionado ao tempo periódico de envio de mensagens para minimizar a probabilidade de dois ou mais líderes enviarem mensagens no mesmo instante de tempo. Cada experimento teve duração de 100 segundos, e os valores médios dos resultados foram representados com base na quantidade de mensagens enviadas ao longo desse período.

A configuração utilizada para construir o cenário de tráfego e o modelo de mobilidade no software SUMO considera que os veículos que compõem os pelotões, cada um composto por 8 veículos de passeio, trafegam em uma via com prédios nas proximidades, curvas e estreitamento e alargamento da pista, mantendo uma velocidade de  $15m/s$ . Além disso, é estabelecido que a aceleração dos veículos durante o percurso é de  $2,6m/s^2$ , enquanto a desaceleração é de  $4,5m/s^2$ . A Tabela 5.3 apresenta um resumo dos parâmetros de mobilidade usados na simulação. Estamos adotando o modelo proposto por Krauss (1998). Este modelo é conhecido como *car-following*

(carro-seguidor), no qual os veículos se posicionam em uma fila mantendo uma distância mínima entre si, chamada de *mingap*, e ajustam suas velocidades com base nas variações de velocidade do veículo que está à frente deles. O modelo de Kraus é uma abordagem que visa criar uma dinâmica de tráfego mais segura e eficiente, onde os veículos se movem de maneira coordenada, mantendo uma distância segura entre si e reagindo às condições de tráfego. Ele é valioso para otimizar o fluxo de veículos e reduzir o risco de acidentes, pois os carros não apenas seguem o líder, mas também respondem a ele de forma adequada, resultando em um tráfego mais fluido e seguro.

**Tabela 5.3** Parâmetros de configuração de mobilidade

Tipo de via	Com obstáculos como prédios, curvas e diferentes larguras de pista
Tamanho do pelotão	8 veículos
Tipo do veículo	Veículo de passeio
Distância entre os veículos	2,5m
Velocidade do veículo	15m/s (54km/h)
Aceleração do veículo	2,6m/s <sup>2</sup>
Desaceleração do veículo	4,5m/s <sup>2</sup>
Controle de direção	Abordagem de Krauss (1998)

As perdas de mensagens são simuladas no canal de comunicação. O critério para definir o sucesso (a perda de mensagens) e o fracasso (a entrega da mensagem ao destino) é definido a partir da distribuição de Bernoulli (DEVORE, 2010). Nesta distribuição discreta de espaço amostral  $\{0, 1\}$ , a variável aleatória  $k$  tem valor 1 com a probabilidade de sucesso  $p$  e valor 0 com a probabilidade de falha  $q = 1 - p$  e a função de probabilidade  $f$  dessa distribuição é dada por:

$$f(k, p) = \begin{cases} p & \text{se } k = 1, \\ 1 - p & \text{se } k = 0. \end{cases} \quad (5.1)$$

Nos experimentos realizados o valor de  $p$  varia entre 0.1 até 0.9.

### 5.1.1 Experimentos utilizando o PBCP

Nesta seção analisamos algumas condições relacionadas à detecção de vizinhança através do PBCP a partir de quatro cenários simulados com o *framework* VEINS. Consideramos em todos os cenários que os pelotões trafegam em uma mesma via e direção. Estes cenários consideram fatores que são determinantes para o estabelecimento de vizinhança entre pelotões.

O cenário 1 tem como objetivo analisar como o tempo limite de progresso é afetado pela variação na quantidade de pelotões utilizando o canal de comunicação. Nesse sentido, o tamanho da mensagem varia em função da quantidade de pelotões presentes na via. Considera-se que cada pelotão possui uma visão composta por 6 bytes correspondente ao identificador do líder, 1 byte para a versão da visão e uma lista contendo 48 bytes correspondendo à identificação

de todos os veículos que compõem o pelotão. Com isso, espera-se que os resultados obtidos permitam avaliar a capacidade de comunicação entre os líderes de pelotão em VANET com diferentes quantidades de pelotões e tamanhos de mensagens.

No cenário 2, analisamos os tempos limite de progresso em relação ao período de envio de mensagens. Os resultados obtidos fornecerão referências para o desenvolvimento de soluções relacionadas à comunicação entre veículos em uma rede considerando o intervalo entre os envios de mensagens *ad hoc*.

No cenário 3, investigamos o impacto da variação do tempo de espera na detecção de vizinhança em um canal de comunicação não confiável. A simulação desse cenário é particularmente relevante em situações em que o canal de comunicação enfrenta altas taxas de perda de mensagens, uma grande quantidade de veículos e a pista apresenta sinuosidades e edificações nas proximidades, criando áreas de sombra.

No cenário 4, examinamos como os resultados obtidos no cenário 3 diferem quando utilizamos uma pista reta, livre de obstáculos e edificações, além de contar com um menor número de veículos envolvidos.

As restrições de tempo para cada cenário são mostradas na Tabela 5.4.

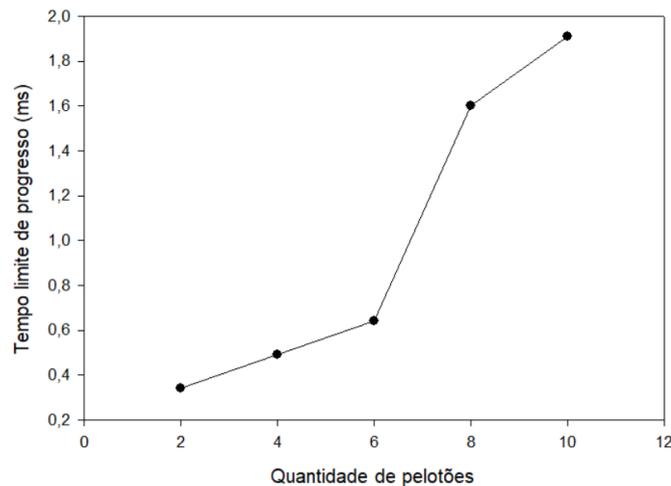
**Tabela 5.4** Restrições de tempo relacionados à periodicidade no envio de mensagens e o tempo de espera .

Cenário	Restrições de tempo (ms)	
	Período	Espera para detecção
1	100	1000
2	10 a 500	1000
3 e 4	500	1000 a 5000

### 5.1.2 Cenário 1 - Determinação dos tempos limite de progresso em função da quantidade de líderes de pelotão e tamanho da mensagem em uma VANET.

Neste cenário, investigamos o impacto da variação no número de líderes de pelotão em uma VANET dentro do tempo limite de progresso. Utilizamos redes com 2, 4, 8 e 10 pelotões nos quais cada líder de pelotão troca mensagens a cada 100 milissegundos. O cenário considera uma rede confiável, sem perda de mensagens, e tanto a comunicação entre pelotões através dos líderes quanto a comunicação intra-pelotões compartilham o mesmo canal de comunicação.

Conforme observado na Figura 5.1, dois fatores influenciaram o tempo limite de progresso: o tamanho das mensagens e o número de veículos competindo pelo canal de comunicação. À medida que o número de pelotões na VANET aumenta, as mensagens se tornam mais longas devido às informações de todos os pelotões vizinhos, o que aumenta a latência. Além disso, a competição pelo canal, tanto entre líderes de pelotão quanto na comunicação intra-pelotão, também afeta o tempo limite de progresso. Por exemplo, em redes com dois líderes, nas quais as mensagens trocadas têm um tamanho de aproximadamente 130 bytes, o tempo limite foi de 0,34 milissegundos. Isso contrasta com redes com 10 líderes, nas quais o tamanho da mensagem é de aproximadamente 570 bytes, resultando em um tempo limite de 1,91 milissegundos.



**Figura 5.1** Variação do tempo limite de progresso em relação à quantidade de pelotões.

Portanto, o atraso na entrega de mensagens é sensivelmente afetado pelo tamanho das mensagens e pelo aumento da competição pelo canal de comunicação.

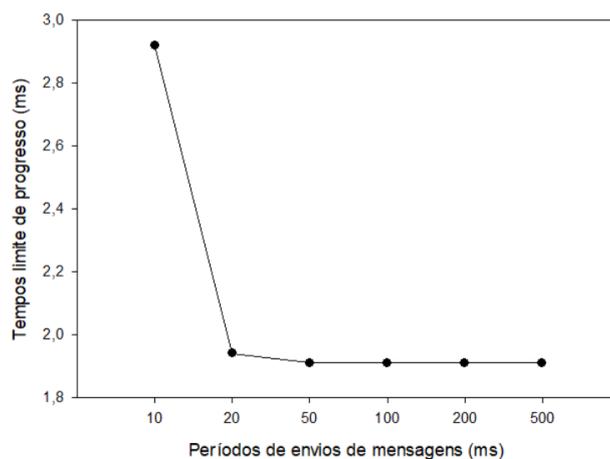
### 5.1.3 Cenário 2 - Determinação dos tempos limite de progresso em função do período de envio.

Neste cenário, determinamos tempos limite de progresso para garantir que as mensagens enviadas entre os líderes de pelotão alcancem todos na vizinhança. Para isso, variamos o período. Supomos que a rede é composta por 10 líderes de pelotão e que o canal de comunicação é confiável. Neste, variamos o período de envio de mensagem entre 10 e 500 milissegundos.

A Figura 5.2 mostra o resultado do experimento. Neste, variamos o período de envio de mensagem entre 10 e 500 milissegundos considerando uma rede com 10 pelotões. Para os períodos com intervalos mais curtos (10 e 20 milissegundos), que representam uma maior quantidade de mensagens enviadas, o tempo limite de progresso alcançou 2,9 e 1,94 milissegundos de atraso, respectivamente. Para períodos entre 50 e 500 milissegundos, o tempo limite de progresso variou muito pouco. O efeito do aumento do número de mensagens enviadas pelos líderes é visível em termos do tempo limite de progresso. À medida que o período diminui maior é a quantidade de mensagens enviadas e maior é o tempo limite de progresso. Isso é causado pelo tempo de contenção do acesso ao canal. Conclui-se, portanto, que a frequência de envio de mensagens e o número de mensagens enviadas têm impacto direto no tempo limite de progresso, devendo ser considerados ao se planejar a disseminação de mensagens em uma rede veicular.

### 5.1.4 Cenário 3 - Determinação das vizinhanças detectadas em um canal de comunicação não confiável variando o tempo de espera com o período fixado.

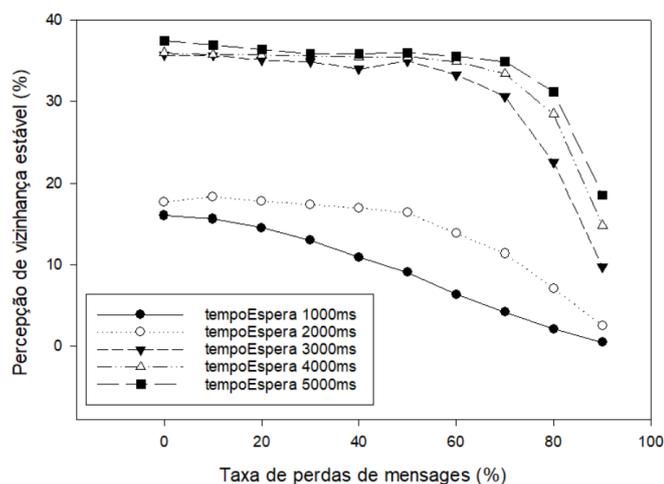
Nessa simulação, consideramos que os pelotões compartilham a pista de Erlangen e se deslocam na mesma direção, mas em velocidades distintas. A comunicação entre os líderes ocorre através de um canal compartilhado, em que tanto os líderes quanto os membros dos pelotões disputam pelo acesso ao meio de comunicação. Neste cenário variamos o tempo de espera entre 1000 milissegundos a 5000 milissegundos.



**Figura 5.2** Variação do tempo limite de progresso em relação valores dos períodos do envio de mensagens.

Como desejamos que o canal de comunicação não sature em função da periodicidade no envio de mensagens fixamos o período em 500 milissegundos. Este foi o maior valor de período apresentado neste artigo. Para a definição da quantidade de líderes escolhemos 10.

Conforme indicado na Figura 5.3, mesmo em situações em que não é imputado, através do simulador, perdas de mensagens, observa-se que a quantidade de vizinhos detectados permanece abaixo de 40%. A baixa taxa de percepção de vizinhança estável se deu em função da alta den-

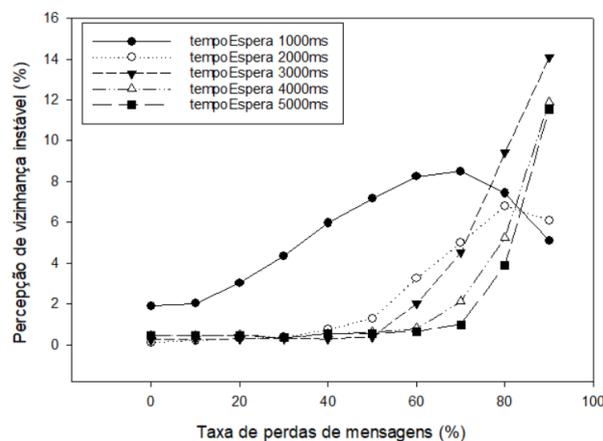


**Figura 5.3** Percepção de vizinhança estável em diferentes tempos de espera em um canal de comunicação não confiável.

sidade de tráfego e o grande número de dispositivos acessando o mesmo canal de comunicação, com isso, causando congestionamento no cana de comunicação. Além disso, a capacidade limitada do padrão 802.11p, quando submetido a situações de alta densidade de tráfego e muitos dispositivos acessando simultaneamente o mesmo canal de comunicação, prejudicou a eficiência da detecção de vizinhança, mesmo em um canal de comunicação considerado confiável ((HAN et al., 2012).

No entanto, nota-se, que a resistência à perda de mensagens aumenta à medida que o tempo de espera cresce. Para o tempo de espera de 5000 milissegundos, por exemplo, em que são enviadas 10 mensagens por cada líder, com até 60% de perdas de mensagens, é possível ter uma percepção de vizinhança estável de quase 40%. Este foi a maior taxa de detecção observada neste cenário.

Com relação à percepção de vizinhança instável (ver Figura 5.4), foi observado que seu aumento está associado à variação na taxa de perda de mensagens no canal e no tempo de espera. Quando a taxa de perda de mensagens aumenta, há uma possibilidade maior de que apenas um líder tenha informações atualizadas do outro líder. Por exemplo, para um tempo de espera de 1000 milissegundos, a percepção de vizinhança instável aumenta até atingir cerca de 70% de taxa de perdas de mensagens e, a partir desse ponto, começa a diminuir gradualmente. A percepção de vizinhança instável pode ser enganosa e não conter informações suficientes para a tomada de decisões importantes, como a junção de dois pelotões. Isso porque é necessário que ambos os pelotões possuam informações um do outro para que possa ocorrer a junção.



**Figura 5.4** Percepção de vizinhança estável em diferentes tempos de espera em um canal de comunicação não confiável.

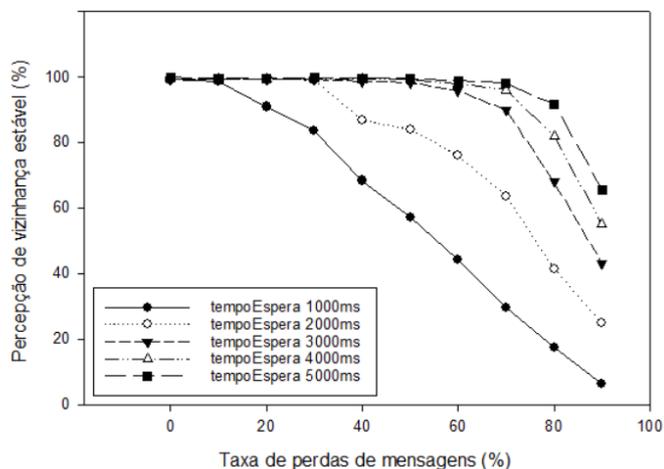
Os resultados obtidos a partir das duas simulações no cenário 3, que analisaram a percepção de vizinhança estável e instável, evidenciam um desempenho insatisfatório do padrão de comunicação 802.11p em ambientes com elevada densidade de veículos. Isso sinaliza a necessidade de soluções capazes de lidar com a recuperação de mensagens em situações de tráfego intenso, visando aprimorar a eficácia da comunicação nesses contextos. Além de mostrar como o tempo de espera pode influenciar na percepção da vizinhança.

#### 5.1.5 Cenário 4 - Determinação das vizinhanças detectadas em um canal de comunicação não confiável variando o tempo de espera com o período fixado através de uma pista reta.

Nessa simulação, consideramos que os pelotões compartilham a mesma rodovia e se deslocam na mesma direção, porém em velocidades diferentes. Além disso, a comunicação entre os líderes ocorre em um canal exclusivo, sem levar em conta a comunicação entre o líder de pelotão e seus seguidores, conforme mencionado em (JIA et al., 2016) e (REN; LI; YANG, 2009). Dessa forma, não há interferência dos demais veículos que compõem os pelotões na competição pelo

acesso ao meio por parte dos líderes.

Para tempos de espera de 1000 milissegundos, em que são enviados 2 por cada líder, a percepção de vizinhança estável apresenta um comportamento regular de queda à medida que a taxa de perdas de mensagens aumenta.



**Figura 5.5** Percepção de vizinhança estável em diferentes tempos de espera em um canal de comunicação não confiável.

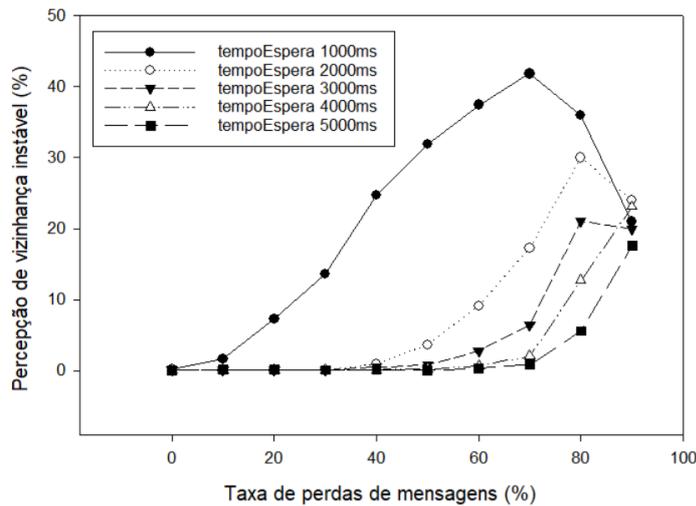
(Figura 5.5). Nota-se, no entanto, que a resistência à perda de mensagens aumenta à medida que o tempo de espera cresce. Para o tempo de espera de 5000 milissegundos, por exemplo, em que são enviadas 10 mensagens por cada líder, com até 70% de perdas de mensagens, é possível detectar quase 100% de vizinhança estável.

Com relação à percepção de vizinhança instável (ver Figura 5.6), foi observado que seu aumento está associado à variação na taxa de perda de mensagens no canal e no tempo de espera. Quando a taxa de perda de mensagens aumenta, há uma possibilidade maior de que apenas um líder tenha informações atualizadas do outro líder. Por exemplo, para um tempo de espera de 1000 milissegundos, a percepção de vizinhança instável aumenta até atingir cerca de 70% e, a partir desse ponto, começa a diminuir gradualmente. A percepção de vizinhança instável pode ser enganosa e não conter informações suficientes para a tomada de decisões importantes, como a junção de dois pelotões. Isso porque é necessário que ambos os pelotões possuam informações um do outro para que possa ocorrer a junção.

Para o tempo de espera de 1000 milissegundos, a percepção de vizinhança instável começa em 1,64% e aumenta para 41,80%, com taxas de perda de mensagens variando de 10% a 70%. Isso ocorre porque, com taxas de perda de mensagens acima de 70%, as chances de que pelo menos um líder detecte outro como vizinho diminuem devido à alta taxa de perda de mensagens.

Outra conclusão deste experimento é a influência que o tempo de espera impõe na capacidade dos líderes perceberem sua vizinhança. Até 70% de perdas de mensagens, o resultado para detecção de vizinhança instável é de aproximadamente 0% para um tempo de espera de 5000 milissegundos. Isto se dá porque, na hipótese de dois líderes estarem trocando mensagens entre si e apenas uma única mensagem seja recebida por cada líder, se as mensagens contiverem as visões mais atuais dos líderes destinatários, ambos os líderes detectarão um ao outro, caracterizando uma vizinhança estável ao invés da detecção de vizinhança instável.

Além disso, comparando os resultados analisados no cenário 3, torna-se evidente como a



**Figura 5.6** Percepção de vizinhança estável em diferentes tempos de espera em um canal de comunicação não confiável.

quantidade de veículos competindo pelo canal de comunicação e as variações na pista, incluindo obstáculos como prédios próximos à pista, estreitamento e alargamento da mesma, exercem uma influência significativa na qualidade da comunicação da rede veicular, mesmo quando consideramos um canal de comunicação sem interferência no sinal. Isso destaca a importância de uma gestão eficiente do espectro de comunicação, bem como a necessidade de estratégias adaptativas para lidar com cenários dinâmicos, a fim de garantir um desempenho confiável e estável da rede veicular em ambientes variados e desafiadores.

## 5.2 CONCLUSÃO

Neste capítulo, apresentamos simulações relacionadas à detecção de vizinhança em uma rede veicular *ad hoc* composta por pelotões de veículos. Em outras palavras, fornecemos uma visão geral sobre o comportamento da detecção de vizinhança em diferentes cenários de perdas e atrasos de mensagens.

A determinação dos tempos limite de progresso e a análise da percepção da vizinhança foram conduzidas por meio de simulações que abrangeram quatro cenários distintos. Cada um desses cenários contemplou uma ampla gama de tempos limite para o progresso, taxas variadas de perda de mensagens e diferentes configurações de pista. Os resultados destes experimentos destacaram a importância de analisar elementos como o número de veículos na rede, a frequência de envio de mensagens e o tamanho das mensagens, pois esses desempenham papéis significativos na capacidade dos veículos em detectar sua vizinhança, bem como no cumprimento dos limites de tempo necessários para assegurar uma comunicação eficaz.

## CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho abordou o tema relacionado à determinação dos atrasos nas entregas de mensagens durante o processo de detecção de vizinhança em uma rede formada por pelotões de veículos. Esse tema apresentou alguns desafios, os quais surgiram principalmente pelo fato dos pelotões serem compostos por entidades comunicantes que se movimentam constantemente, o que pode resultar em alterações na topologia da rede, na quantidade de veículos nos pelotões, assim como no tamanho das mensagens trocadas entre os líderes de pelotões. Esses aspectos afetam diretamente o tempo de entrega das mensagens, conforme observado nesta dissertação. Além disso, constatamos que a mobilidade dos veículos causa instabilidade e dinamicidade na infraestrutura, sendo as simulações em diferentes cenários essenciais para determinar os tempos de entrega das mensagens.

O estudo desses tempos de entrega de mensagens seria complicado pelo grande número de suposições de comunicação possíveis: as transmissões simultâneas causam colisões? As colisões podem ser detectadas? A recepção da mensagem pode sofrer algum tipo de interferência em função das distâncias geográficas ou por critérios mais complexos, como a relação sinal-ruído? E assim por diante. Para lidar com este problema, optamos por utilizar uma camada MAC abstrata, que abstrai os detalhes de baixo nível, como as suposições de comunicação possíveis. Essa abstração permitiu considerar os atrasos nas entregas de mensagens como limites de tempo abstratos, chamados de tempos limite de progresso. Isto possibilitou que o estudo se concentrasse diretamente na análise dos tempos limite de progresso, observando como as variações nos cenários, como o tamanho das mensagens, a quantidade de veículos na rede e a periodicidade do envio de mensagens, influenciavam esses tempos.

Nas simulações realizadas, utilizamos um protocolo simples desenvolvido nesta dissertação, com o objetivo de assegurar que as informações trocadas entre pelotões vizinhos estivessem corretas e atualizadas. Aplicamos a verificação de modelos utilizando o verificador PRISM para verificar a correção do protocolo.

As informações obtidas por meio da simulação de cenários utilizando a plataforma OMNET++ em conjunto com os simuladores de rede veicular VEINS e de tráfego rodoviário SUMO podem ser úteis para o desenvolvimento de sistemas eficientes de detecção de vizinhança. O processo de detecção de vizinhança neste trabalho retratou a comunicação entre pelotões sem perda de generalidade já que pode ser aplicado na comunicação e gerenciamento entre dois veículos. Os experimentos realizados revelaram que o aumento no tamanho das mensagens e no número de

veículos utilizando o canal de comunicação resultam em um maior tempo limite de progresso. Além disso, foi observado que a redução do intervalo entre os envios de mensagens também ocasionou um aumento no tempo limite de progresso. Portanto, esta dissertação contribui ao determinar os tempos limite de progresso para a entrega de mensagens em redes veiculares *ad hoc* sujeitas a diversas perturbações no canal de comunicação em diferentes cenários.

## 6.1 LIMITAÇÕES

A principal limitação encontrada nesta dissertação foi a simulação de cenários simplificados para análise dos tempos limite de progresso. Apesar de termos realizado diversas simulações, variando valores de parâmetros como quantidade de pelotões, tempo de espera, porcentagem de perda de mensagens e atrasos na entrega, não foi possível realizar a simulação em um ambiente veicular real.

Outra limitação foi a não utilização da extensão para simulação de pelotão Plexe (RANGO; RAIMONDO; AMENDOLA, 2019). Essa extensão do VEINS poderia ter sido incorporada ao estudo para aprimorar a compreensão dos sistemas de condução autônoma em formação de pelotão. A ausência dessa extensão limitou a análise dos tempos limite de progresso em situações específicas, como a variação dos tempos durante o processo de comunicação entre os líderes de pelotão e a interferência que os membros dos pelotões poderiam causar na comunicação entre líderes. A simulação de pelotão através do Plexe poderia ter proporcionado uma visão mais abrangente e realista do desempenho dos veículos autônomos em uma formação de pelotão.

## 6.2 TRABALHOS FUTUROS

Diferentes possibilidades para trabalhos futuros poderão ser exploradas. Por exemplo, pode-se realizar uma avaliação dos tempos limite de progresso que inclua também a segurança na comunicação, considerando seus custos adicionais. Isso pode incluir o estudo de técnicas de criptografia e autenticação para garantir a confidencialidade e integridade das mensagens trocadas entre os nós da rede. Outra possibilidade é avaliar a viabilidade da implementação de protocolos de roteamento seguros em redes de pelotões *ad hoc*, levando em consideração as limitações de recursos dos nós da rede veicular e os tempos limite de progresso adequados.

Outro trabalho futuro que pode ser explorado é a otimização dos tempos limite de progresso em redes de pelotões. Isso pode ser alcançado através da utilização de algoritmos de otimização para encontrar valores que levem a um melhor desempenho e menor consumo de recursos. Além disso, pode-se investigar a possibilidade de reconfiguração dinâmica de pelotões para melhorar o desempenho da rede.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ABASSI, R. Vanet security and forensics: Challenges and opportunities. *Wiley Interdisciplinary Reviews: Forensic Science*, Wiley Online Library, v. 1, n. 2, p. e1324, 2019.
- AHMED, E.; GHARAVI, H. Cooperative vehicular networking: A survey. *IEEE Transactions on Intelligent Transportation Systems*, v. 19, n. 3, p. 996–1014, 2018.
- AKBAR, M. et al. Evaluation of ieee 802.11n for multimedia application in vanet. *Procedia Computer Science*, v. 32, p. 953–958, 2014. ISSN 1877-0509. The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014). Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050914007170>.
- ALLEN, E.; EDMUND, C. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, v. 2, n. 3, p. 241 – 266, 1982. ISSN 0167-6423. Disponível em: <http://www.sciencedirect.com/science/article/pii/0167642383900175>.
- ALMEIDA, T. et al. Análise de desempenho do ieee 802.11 p: Simulações versus experimentos reais. In: SBC. *Anais Principais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. [S.l.], 2018.
- ALVES, R. et al. Redes veiculares: Princípios, aplicações e desafios. In: \_\_\_\_\_. [S.l.: s.n.], 2009. p. 199–254.
- AMEEN, H. A. et al. A review on vehicle to vehicle communication system applications. *Indonesian Journal of Electrical Engineering and Computer Science*, v. 18, n. 1, p. 188–198, 2020.
- AMOOZADEH, M. et al. Platoon management with cooperative adaptive cruise control enabled by vanet. *Vehicular Communications*, v. 2, n. 2, p. 110–123, 2015. ISSN 2214-2096. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2214209615000145>.
- ARANTES, L. et al. Partition participant detector with dynamic paths in mobile networks. In: *2010 Ninth IEEE International Symposium on Network Computing and Applications*. [S.l.: s.n.], 2010. p. 224–228.
- ARENA, F.; PAU, G.; SEVERINO, A. A review on ieee 802.11 p for intelligent transportation systems. *Journal of Sensor and Actuator Networks*, MDPI, v. 9, n. 2, p. 22, 2020.
- BAIER, C.; KATOEN, J.-P. *Principles of model checking*. [S.l.]: MIT press, 2008.
- BARIAH, L. et al. Recent advances in vanet security: a survey. In: IEEE. *2015 IEEE 82nd vehicular technology conference (VTC2015-fall)*. [S.l.], 2015. p. 1–7.
- BEHRISCH, M. et al. Sumo—simulation of urban mobility: an overview. In: THINKMIND. *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. [S.l.], 2011.

BILSTRUP, K. et al. Evaluation of the ieee 802.11p mac method for vehicle-to-vehicle communication. In: . [S.l.: s.n.], 2008. p. 1 – 5.

BRUNO, F. et al. A probabilistic model checking analysis of vehicular ad-hoc networks. In: *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. [S.l.: s.n.], 2015. p. 1–7. ISSN 1550-2252.

CABALLERO-GIL, C.; CABALLERO-GIL, P.; MOLINA-GIL, J. Using groups to reduce communication overhead in vanets. *AP2PS 2010 - 2nd International Conference on Advances in P2P Systems*, p. 6, 01 2010.

CHEN, L.; BIAN, K. Neighbor discovery in mobile sensing applications: A comprehensive survey. *Ad Hoc Networks*, Elsevier, v. 48, p. 38–52, 2016.

CHOCKLER, G.; KEIDAR, I.; VITENBERG, R. Group communication specifications: A comprehensive study. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 33, n. 4, p. 427–469, dez. 2001. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/503112.503113>.

CLARKE, E. M. *The Birth of Model Checking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. 1–26 p. ISBN 978-3-540-69850-0. Disponível em: [https://doi.org/10.1007/978-3-540-69850-0\\_1](https://doi.org/10.1007/978-3-540-69850-0_1).

CORNEJO, A.; VIQAR, S.; WELCH, J. Reliable neighbor discovery for mobile ad hoc networks. *Ad Hoc Networks*, v. 12, p. 259–277, 2014. ISSN 1570-8705. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1570870512001564>.

COULOURIS, G. et al. *Distributed Systems: Concepts and Design*. 5th. ed. USA: Addison-Wesley Publishing Company, 2011. ISBN 0132143011, 9780132143011.

CUNHA, F. et al. Data communication in vanets: Survey, applications and challenges. 12 2016.

DENG, D.-J. et al. A collision alleviation scheme for ieee 802.11 p vanets. *Wireless Personal Communications*, Springer, v. 56, n. 3, p. 371–383, 2011.

DEVORE, J. L. *Probabilidade e estatística para engenharia e ciências*. [S.l.]: Cengage Learning Edições Ltda., 2010.

DIANGE, Y. et al. Intelligent and connected vehicles: Current status and future perspectives. *Science China Technological Sciences*, Springer, v. 61, n. 10, p. 1446–1471, 2018.

DU, J.; WANG, S.; ZHANG, B. Vehicle density and signal to noise ratio based broadcast backoff algorithm for vanets. *Ad Hoc Networks*, v. 99, p. 102071, 2020. ISSN 1570-8705. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1570870519309333>.

ERIKA, A. et al. Dtmc model checking by scc reduction. In: *2010 Seventh International Conference on the Quantitative Evaluation of Systems*. [S.l.: s.n.], 2010. p. 37–46.

FALL, K. Messaging in difficult environments. *Intel Research Berkeley, IRB-TR-04-019*, 2004.

FERNANDES, P.; NUNES, U. Platooning of autonomous vehicles with intervehicle communications in sumo traffic simulator. In: *13th International IEEE Conference on Intelligent Transportation Systems*. [S.l.: s.n.], 2010. p. 1313–1318. ISSN 2153-0017.

- FILIERI, A.; GHEZZI, C.; TAMBURRELLI, G. Run-time efficient probabilistic model checking. In: *Proceedings of the 33rd International Conference on Software Engineering*. New York, NY, USA: ACM, 2011. (ICSE '11), p. 341–350. ISBN 978-1-4503-0445-0. Disponível em: <http://doi.acm.org/10.1145/1985793.1985840>.
- FKAIER, S.; KHALGUI, M.; FREY, G. Modeling methodology for reconfigurable distributed systems using transformations from gr-uml to gr-tnces and iec 61499. In: *ENASE*. [S.l.: s.n.], 2021. p. 221–230.
- GALAVIZ-MOSQUEDA, G. et al. Reliable multihop broadcast protocol with a low-overhead link quality assessment for its based on vanets in highway scenarios. *The Scientific World Journal*, v. 2014, p. 1–12, 07 2014.
- HADZILACOS, V.; TOUEG, S. *A Modular Approach to Fault-Tolerant Broadcasts and Related Problems*. USA, 1994.
- HAIKIA, P. et al. Performance analysis of IEEE 802.11p DCF for multiplatoonig communications with autonomous vehicles. *IEEE Transactions on Vehicular Technology*, IEEE, v. 66, n. 3, p. 2485–2498, March 2017. ISSN 0018-954. Disponível em: <https://doi.org/10.1109/TVT.2016.2571696>.
- HAN, C. et al. Analytical study of the iee 802.11 p mac sublayer in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 13, n. 2, p. 873–886, 2012.
- HASAN, F. et al. Time synchronization in vehicular ad-hoc networks: A survey on theory and practice. *Vehicular Communications*, v. 14, p. 39 – 51, 2018. ISSN 2214-2096. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2214209618300421>.
- HASSANABADI, B.; VALAEE, S. Reliable periodic safety message broadcasting in vanets using network coding. *IEEE Transactions on Wireless Communications*, v. 13, n. 3, p. 1284–1297, 2014.
- JIA, D. et al. A survey on platoon-based vehicular cyber-physical systems. *IEEE Communications Surveys Tutorials*, v. 18, n. 1, p. 263–284, 2016.
- JIN, L.; ZHANG, G.; ZHU, X. Formal analysis and evaluation of the back-off procedure in iee802. 11p vanet. *Modern Physics Letters B*, World Scientific, v. 31, n. 19-21, p. 1740063, 2017.
- KHAIRNAR, V. D.; KOTTECHA, K. Performance of vehicle-to-vehicle communication using iee 802.11 p in vehicular ad-hoc network environment. *arXiv preprint arXiv:1304.3357*, 2013.
- KRAUSS, S. Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics. 1998.
- KUHN, F.; LYNCH, N.; NEWPORT, C. The abstract mac layer. In: SPRINGER. *International Symposium on Distributed Computing*. [S.l.], 2009. p. 48–62.
- KWIATKOWSKA, M.; NORMAN, G.; PARKER, D. Quantitative analysis with the probabilistic model checker prism. *Electronic Notes in Theoretical Computer Science*, v. 153, n. 2, p. 5 – 31, 2006. ISSN 1571-0661. Proceedings of the Third Workshop on Quantitative Aspects of Programming Languages (QAPL 2005). Disponível em: <http://www.sciencedirect.com/science/article/pii/S157106610600257X>.

LABERTEAUX, K.; HARTENSTEIN, H. *VANET: vehicular applications and inter-networking technologies*. [S.l.]: John Wiley & Sons, 2009.

LEE, M.; ATKISON, T. Vanet applications: Past, present, and future. *Vehicular Communications*, v. 28, p. 100310, 2021. ISSN 2214-2096. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2214209620300814>.

LI, J.; CHIGAN, C. Achieving robust message dissemination in vanet: Challenges and solution. In: CITESEER. *IEEE intelligent vehicles symposium (IV)*. [S.l.], 2011. p. 845–850.

LOPEZ, P. A. et al. Microscopic traffic simulation using sumo. In: *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. Disponível em: <https://elib.dlr.de/124092/>.

MAHAJAN, A. et al. Urban mobility models for vanets. In: *2nd IEEE International Workshop on Next Generation Wireless Networks*. [S.l.: s.n.], 2006. v. 33, p. 38.

MARTA, K.; GETHIN, N.; PARKER, D. The prism benchmark suite. In: *International Conference on Quantitative Evaluation of Systems*. Los Alamitos, CA, USA: IEEE Computer Society, 2012. p. 203–204. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/QEST.2012.14>.

MARTA, K.; GETHIN, N.; PARKER, D. The prism benchmark suite. In: *International Conference on Quantitative Evaluation of Systems*. Los Alamitos, CA, USA: IEEE Computer Society, 2012. p. 203–204. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/QEST.2012.14>.

MEZGHANI, F. et al. Interest-based forwarding for satisfying user preferences in vehicular networks. In: SPRINGER. *International Conference on Ad Hoc Networks*. [S.l.], 2015. p. 3–14.

MIHAIL, S.; MARIA, K. Inter-vehicle communication systems: a survey. *IEEE Communications Surveys Tutorials*, v. 10, n. 2, p. 88–105, Second 2008. ISSN 1553-877X.

MUHAMMAD, S. J.; ZHANG, S.; DYU, V. Timely and efficient multihop broadcast scheme for reliable inter-vehicular communication. In: IEEE. *2018 24th International Conference on Automation and Computing (ICAC)*. [S.l.], 2018. p. 1–7.

NGUYEN, V. et al. Mac protocols with dynamic interval schemes for vanets. *Vehicular Communications*, Elsevier, v. 15, p. 40–62, 2019.

OMAR, H. A.; ZHUANG, W.; LIT, L. Delay analysis of vemac supporting periodic and event-driven safety messages in vanets. In: *2013 IEEE Global Communications Conference (GLOBECOM)*. [S.l.: s.n.], 2013. p. 4621–4626.

OUCHANI, S. Ensuring the functional correctness of iot through formal modeling and verification. In: ABDELWAHED, E. H. et al. (Ed.). *Model and Data Engineering*. Cham: Springer International Publishing, 2018. p. 401–417.

PARK, P. et al. Wireless network design for control systems: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 2, p. 978–1013, 2017.

PAULO, O. F. d. C. *Revisão de modelos CTL*. 70 p. Dissertação (Mestrado) — Universidade de São Paulo, São Paulo, 2010.

- PENG, H. et al. Performance analysis of ieee 802.11p dcf for multiplatooning communications with autonomous vehicles. *IEEE Transactions on Vehicular Technology*, v. 66, n. 3, p. 2485–2498, 2017.
- PITKANEN, M.; KERANEN, A.; OTT, J. Message fragmentation in opportunistic dtns. In: IEEE. *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*. [S.l.], 2008. p. 1–7.
- QURESHI, K.; ABDULLAH, H. Topology based routing protocols for vanet and their comparison with manet. *Journal of Theoretical and Applied Information Technology*, v. 58, n. 3, p. 707–715, 2013.
- RAJESH, M.; GNANASEKAR, J. Consistently neighbor detection for manet. In: IEEE. *2016 International Conference on Communication and Electronics Systems (ICCES)*. [S.l.], 2016. p. 1–9.
- RAMANATHAN, R. et al. An empirical study on mac layer in ieee 802.11 p/wave based vehicular ad hoc networks. *Procedia computer science*, Elsevier, v. 143, p. 720–727, 2018.
- RANGO, F. D.; RAIMONDO, P.; AMENDOLA, D. Extending sumo and plexe simulator modules to consider energy consumption in platooning management in vanet. In: IEEE. *2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. [S.l.], 2019. p. 1–9.
- REN, Z.; LI, W.; YANG, Q. Location verification for vanets routing. In: *2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*. [S.l.: s.n.], 2009. p. 141–146.
- ROBINSON, T.; COELINGH, E. Operating platoons on public motorways: An introduction to the sartre platooning programme. 12 2018.
- ROY, A.; ACHARYA, T.; DASBIT, S. Quality of service in delay tolerant networks: A survey. *Computer Networks*, Elsevier, v. 130, p. 121–133, 2018.
- SANGUESA, J. A. et al. A survey and comparative study of broadcast warning message dissemination schemes for vanets. *Mob. Inf. Syst.*, v. 2016, p. 8714142:1–8714142:18, 2016.
- SHAH, S. A. A. et al. Coverage differentiation based adaptive tx-power for congestion and awareness control in vanets. *Mobile Networks and Applications*, Springer, v. 23, n. 5, p. 1194–1205, 2018.
- SHAKEEL, S. M. et al. Experimental evaluation of safety beacons dissemination in vanets. *Procedia Computer Science*, Elsevier, v. 56, p. 618–623, 2015.
- SOMMER, C. et al. Veins: The open source vehicular network simulation framework. In: *Recent advances in network simulation*. [S.l.]: Springer, 2019. p. 215–252.
- STIBOR, L.; ZANG, Y.; REUMERMAN, H.-J. Evaluation of communication distance of broadcast messages in a vehicular ad-hoc network using ieee 802.11 p. In: IEEE. *2007 IEEE Wireless Communications and Networking Conference*. [S.l.], 2007. p. 254–257.

- SUN, N. *Performance study of ieee 802.11 p for vehicle to vehicle communications using opnet: a thesis presented in partial fulfilment of the requirements for the degree of master of engineering in telecommunications and network at massey university, auckland, new zealand*. Tese (Doutorado) — Massey University, 2011.
- SUN, W. et al. Analytical study of the ieee 802.11 p edca mechanism. In: IEEE. *2013 IEEE Intelligent Vehicles Symposium (IV)*. [S.l.], 2013. p. 1428–1433.
- VALÉRIO, R.; PEDRO, S.; FRANCISCO, V. A group membership protocol for communication systems with both static and dynamic scheduling. In: . [S.l.: s.n.], 2006. p. 22 – 31.
- VARGA, A.; HORNIG, R. An overview of the omnet++ simulation environment. In: *1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems*. [S.l.: s.n.], 2010.
- WANG, J.; LIU, J.; KATO, N. Networking and communications in autonomous driving: A survey. *IEEE Communications Surveys and Tutorials*, IEEE, v. 21, n. 2, p. 1243–174, 2019. ISSN 1553-877x. Disponível em: <https://doi.org/10.1109/COMST.2018.2888904>.
- YANG, Q.; ZHENG, J.; SHEN, L. Modeling and performance analysis of periodic broadcast in vehicular ad hoc networks. In: . [S.l.: s.n.], 2011. p. 1–5.
- YU, D. et al. Implementing the abstract mac layer in dynamic networks. *IEEE Transactions on Mobile Computing*, v. 20, n. 5, p. 1832–1845, 2021.
- ZHU, W. et al. A collision avoidance mechanism for emergency message broadcast in urban vanet. In: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. [S.l.: s.n.], 2016. p. 1–5.

## Apêndice

# A

## SIMULADORES

### A.1 SUMO

O SUMO é um simulador de tráfego urbano de código aberto que é capaz de simular a movimentação de diferentes tipos de veículos, que se movimentam em diferentes velocidades e suporta múltiplas faixas. O SUMO não é apenas um simulador de tráfego, mas um conjunto de aplicações que ajuda na preparação da execução da simulação do tráfego urbano. Sua interface pode ser verificada na Figura A.1



**Figura A.1** Simulador SUMO

Tanto a representação da via quanto as demandas de tráfego são exigidas pela simulação e são importadas ou geradas usando diferentes fontes. Existem algumas formas de gerar mapas no SUMO. Uma delas consiste na construção do código de forma manual, na qual é possível selecionar a quantidade de interseções e a distância entre elas.

Para a criação do cenário de simulação foram configurados alguns arquivos com extensão .xml específicos. Essa configuração foi realizada através da criação dos seguintes arquivos .xml: .net e .rou. O arquivo de extensão .net é responsável pelas informações do mapa onde foram realizadas as simulações. O arquivo de extensão .rou reúne informações dos dados básicos dos

veículos, tais como comprimento, aceleração, velocidade máxima e outros, além das rotas a serem seguidas por eles. Por fim, há o arquivo de configuração `sumo.cfg`, que contém a configuração da simulação formada pelos arquivos anteriormente descritos. De posse dos arquivos mencionados, foi possível executar a simulação e visualizar o movimento dos veículos por meio da interface gráfica do SUMO.

Para a configuração das demandas do tráfego e características da pista e dos pelotões, editamos o arquivo com a extensão `rou.xml`. Inicialmente, definimos os veículos com uma rota que lhes pertence. Três atributos foram definidos neste arquivo: *vType*, *route* e *flow*. No atributo *vType* definimos a cor dos veículos utilizando o sistema de cor aditivada RGB, *Red*, *Green*, *Blue* e nomeamos o veículo com um id. Além disso, determinamos a velocidade dos veículos em m/s, o espaço entre o líder do pelotão e o veículo à frente. Por fim, configuramos a aceleração e a desaceleração. No atributo *flow* definimos a rota explicitamente. Além disso, definimos a quantidade de veículos por pelotão, o tempo em que cada veículo é inserido na pista para iniciar a troca de mensagens, a faixa da pista em que o pelotão será inserido como inicialmente (considera-se que os pelotões podem mudar de faixa durante o percurso) e por fim, a faixa em que o pelotão permanecerá. Por fim, no atributo *route* é definido a rota pela qual o pelotão irá trafegar.

O arquivo com a extensão `net.xml` é dividido em *edge*, *requests*, *junction* e *connection*. Cada *edge* inclui as definições das pistas das quais consiste. Em nossa simulação, utilizamos dois tipos de pistas: uma que representa um mapa real de uma cidade do estado da Baviera, Alemanha, e a outra é uma pista reta com duas faixas.

No caso da pista reta no *junction*, são definidos os cruzamentos. Definimos que as duas pistas com duas faixas se cruzam. No *connection*, são descritas quais faixas de saída podem ser alcançadas a partir de uma faixa de entrada. Além disso, é fornecida a primeira faixa a ser usada para passar pela interseção. Informações adicionais, como o uso de semáforos, não são definidas.

## A.2 OMNET++

O OMNET++, *Objective Modular Network Testbed in C++* é um framework de simulação de eventos discretos orientado a objeto, de código aberto, que provê uma infraestrutura para construção de simulações. Essa infraestrutura conta com uma arquitetura de componentes para modelos de simulação, que são montados a partir de módulos. Os módulos podem ser conectados entre si através de portas e combinados para formar módulos compostos. As conexões são criadas dentro de um único nível de hierarquia de módulos. Um submódulo pode ser conectado com outro ou com o módulo composto. Cada modelo de simulação é uma instância de um tipo de módulo composto.

O simulador OMNET++ utiliza a linguagem proprietária NED, *Network Description*, que é baseada em C++, como ferramenta para modelar a topologia da rede e dos nós (veículos). Basicamente, a linguagem NED permite a definição de módulos simples, complexos e de rede, sendo estes últimos os que contém os componentes e especificações de uma rede de comunicações, como a quantidade dos veículos. Os eventos ocorrem dentro de módulos simples, os quais são implementados em C++ com a biblioteca de classes OMNET++.

Nos módulos, os algoritmos e as propriedades dos módulos iniciam ou finalizam em cada mudança de estado. Uma vez configurados todos os módulos, será necessária a criação do arquivo `.ini` que permitirá o início da simulação. Neste arquivo são descritos tanto os parâmetros próprios da simulação quanto o número de interações, o tempo de duração, os valores dos atributos dos

módulos e toda a topologia do modelo de rede.

### A.3 VEINS

VEINS (*VEhIcular Network Simulation*) é um simulador de redes veiculares que permite a interação dinâmica entre outros dois simuladores: OMNET++ e SUMO (VARGA; HORNIG, 2010; SOMMER et al., 2019; BEHRISCH et al., 2011). Essa interação é fundamental para a execução da simulação de uma rede veicular, uma vez que um simulador de rede, a exemplo do OMNET++, é utilizado para modelar configurações de redes de computadores, porém o suporte à mobilidade é limitado a padrões de mobilidade simples que não podem ser utilizados para experiências em cenários de VANETs. Por outro lado, o simulador de mobilidade de tráfego rodoviário em combinação com um simulador de rede possibilita uma avaliação mais realista. Assim, a influência das redes veiculares sobre o tráfego rodoviário pode ser modelada e interações complexas entre os dois domínios examinados.

Os simuladores OMNeT++ e SUMO são interligados por meio de um *socket TCP*, e assim seguido de um protocolo de padronização, TraCI (*Traffic Control Interface*), com isto é possível que a simulação seja acoplada pela simulação de rede e pela simulação de tráfego. Os movimentos realizados na simulação de tráfego pelos veículos no SUMO são refletidos em movimentos dos nós contidos na simulação de rede no OMNeT++ ((LOPEZ et al., 2018)). A arquitetura utilizada nesse acoplamento é cliente-servidor, onde o cliente enviou comandos ao servidor para controlar o estado da simulação. O cliente e o servidor são, respectivamente, o OMNET++ e o SUMO.

A Figura A.2 ilustra a estrutura de acoplamento dos simuladores OMNET++ e SUMO. O modo básico de acoplamento do OMNET++ e do SUMO consiste no armazenamento ordenado, envio e execução de todos os comandos realizados no período

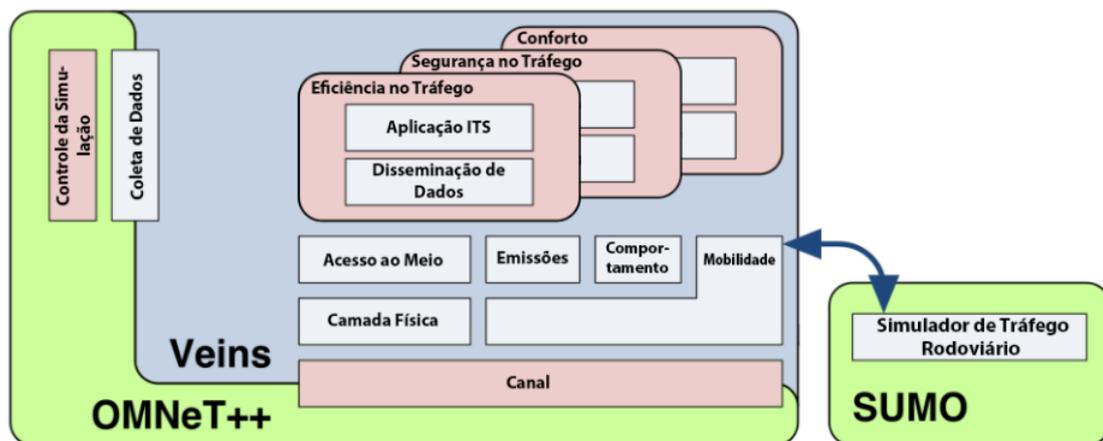


Figura A.2 Arquitetura do VEINS e conexões com o OMNET++ e SUMO

de simulação, o que garante a execução de tais comandos. Durante a execução da simulação, os módulos de comunicação trocam comandos por meio de conexões TCP. O OMNET++ é um simulador baseado em eventos, portanto, lida com a mobilidade, agendando movimentos de nós em intervalos regulares. Tal característica favorece a abordagem do SUMO que, assim como o OMNET++, avança o tempo de simulação em passos discretos. Em cada passo, o OMNET++ então envia todos os comandos armazenados para o SUMO e dispara o próximo passo correspondente da simulação de tráfego rodoviário.

Algumas características importantes desse simulador são:

- 100% de código aberto, o que oferece extensibilidade irrestrita;
- Baseado em modelos totalmente detalhados dos padrões IEEE 802.11p, incluindo a operação multicanal, acesso ao canal QoS, ruídos e efeitos de interferência;
- Oferecimento de uma ampla gama de métricas de avaliação, incluindo quantidade de mensagens enviados e perdidos, tempo de viagem, distância total percorrida, entre outros.

## NOÇÕES SOBRE VERIFICAÇÃO DE MODELOS

Para verificar o sistema, seu comportamento deve ser modelado em um sistema de estados finitos, normalmente a estrutura de Kripke. Na prática, várias linguagens de alto nível são utilizadas para a modelagem, sendo uma delas a linguagem de modelagem PRISM (MARTA; GETHIN; PARKER, 2012b).

Em uma estrutura Kripke, um estado captura o valor de cada variável do modelo em um instante de tempo específico. As transições entre estados, representam os possíveis comportamentos do sistema à medida que o tempo avança. O conjunto de pares que formam as transições é chamado de relação de transição. Uma sequência de estados que está de acordo com a relação de transição é chamada de caminho.

O comportamento das estruturas de Kripke pode ser descrito através de lógicas temporais. Lógica temporal é um formalismo que descreve a ordenação de eventos no tempo, sem introduzir o tempo explicitamente (CLARKE, 2008). É uma extensão da lógica proposicional tradicional, com operadores que se referem ao comportamento dos sistemas ao longo do tempo. Uma lógica temporal muito utilizada na verificação de modelos é a lógica de árvore de computação (CTL, *Computation Tree Logic*).

### B.1 LÓGICA DE ÁRVORE DE COMPUTAÇÃO

A noção de ramificação de tempo foi abordada por Allen e Edmund (1982) juntamente com uma lógica temporal, onde é possível expressar propriedades que definem o comportamento do sistema num futuro não determinado. Os sistemas têm como caracterização básica estados e caminhos, sendo o estado a descrição do sistema em um dado instante de tempo, ou seja, valores associados às suas variáveis naquele instante, e o caminho representa a relação entre vários estados numa linha do tempo.

Desta forma, através da lógica de árvore de computação, é possível raciocinar sobre a evolução futura do sistema. Aplicada a sistemas computacionais, a lógica temporal CTL é capaz de quantificar sobre os diversos caminhos que uma execução pode tomar, como por exemplo, se dois processos estarão utilizando simultaneamente o mesmo recurso em um dado momento, em algum dos caminhos de execução possíveis.

**Definição 2.1** (Sintaxe CTL). A linguagem de CTL é sintaticamente definida através da forma Backus Naur <sup>1</sup> com a seguinte gramática:

$$\begin{aligned} \phi ::= & \top \mid \perp \mid p \mid (\neg\phi) \mid (\phi \vee \neg\phi) \mid (\phi \wedge \neg\phi) \mid (\phi \rightarrow \neg\phi) \mid EX\phi \\ & \mid AX\phi \mid EF\phi \mid AF\phi \mid EG\phi \mid AG\phi \mid EF\phi \mid E[\phi U \phi] \mid A[\phi U \phi] \end{aligned}$$

onde  $p$  é um átomo proposicional,  $\neg$ ,  $\wedge$ ,  $\vee$  e  $\rightarrow$  são operadores lógicos clássicos e os demais constituem os operadores temporais e seus quantificadores de caminhos.

Cada operador temporal é composto por um quantificador de caminho, a saber,  $A$  e  $E$ . O quantificador  $A$  significa "ao longo de todos os caminhos" (inevitavelmente) e  $E$  significa "ao menos (existe) um caminho". Os conectivos  $X$ ,  $F$ ,  $G$  ou  $U$ , significam "próximo estado", "algum estado futuro", "todos os estados futuros (globalmente)" e "até algum estado", respectivamente. Os símbolos  $X, F, G, U$  não podem ocorrer sem serem precedidos por um  $A$  ou  $E$ ; da mesma forma, cada  $A$  ou  $E$  deve preceder um dos  $X, F, G, U$ . Para uma definição formal da semântica de CTL é necessário antes definir o conceito de estrutura de Kripke.

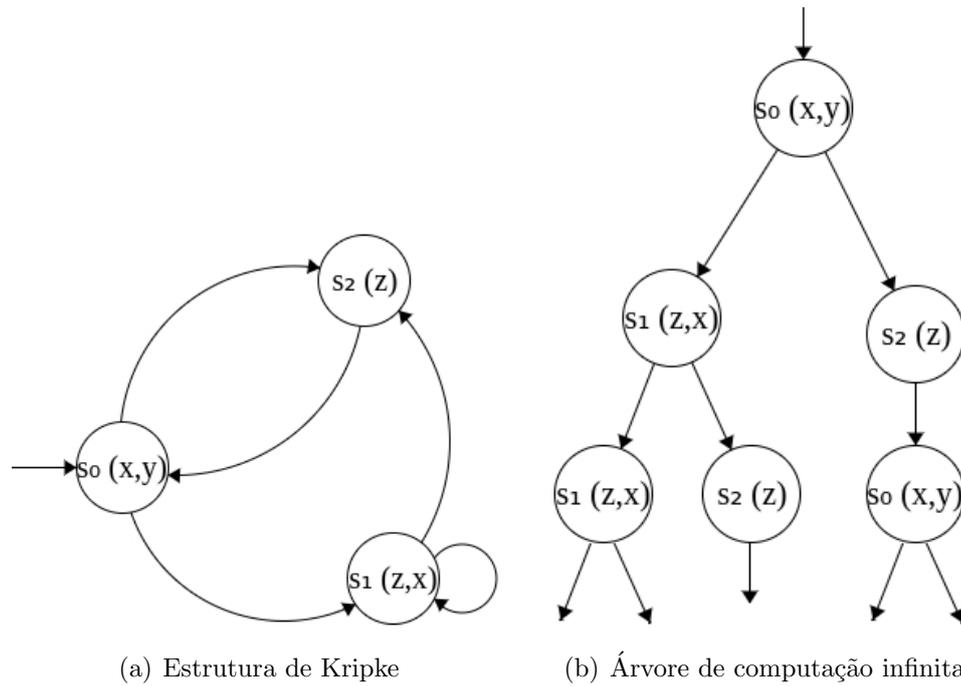
**Definição 2.2** (Semântica CTL). Uma estrutura de Kripke é uma tupla  $(S, i, R, L)$  em que  $S$  representa conjunto de estados,  $i \in S$  um conjunto de estados iniciais,  $R \subseteq S \times S$  uma relação entre estes estados. Seja  $AP$  um conjunto de propriedade atômicas,  $L : S \rightarrow AP$  é uma associação entre cada estado e um conjunto de valores proposicionais.

A Figura B.1 apresenta uma estrutura de Kripke onde  $AP = \{x, y, z\}$ ,  $S = \{s_0, s_1, s_2\}$ ,  $R = \{(s_0, s_1), (s_0, s_2), (s_1, s_2), (s_1, s_1)\}$ ,  $i = \{s_0\}$  com  $L(s_0) = \{x, y\}$ ,  $L(s_1) = \{z, x\}$  e  $L(s_2) = \{z\}$ . Conforme mostrado nesta figura a estrutura de Kripke gera uma árvore de computação infinita através do desdobramento da estrutura a partir de um estado. Na Figura B.1(a) é apresentada a árvore de computação infinita (no qual o futuro não é determinado) que é obtida através do desdobramento da estrutura de Kripke (Figura B.1(b)) a partir do estado  $s_0$ .

Considere  $\mathcal{M}$  uma estrutura de Kripke e  $s$  um estado dessa estrutura. A relação  $(\mathcal{M}, s) \models \phi$ , onde  $\phi$  é uma fórmula CTL, ocorre se e somente se  $\phi$  é satisfeita no estado  $s$  de  $\mathcal{M}$  e é definida indutivamente sobre a estrutura de  $\phi$ :

1.  $\mathcal{M}, s \models \top$  sse  $\mathcal{M}, s \not\models \perp$
2.  $\mathcal{M}, s \models p$  sse  $p \in L(s)$
3.  $\mathcal{M}, s \models \neg\phi$  sse  $\mathcal{M}, s \not\models \phi$
4.  $\mathcal{M}, s \models \phi_1 \wedge \phi_2$  sse  $\mathcal{M}, s \models \phi_1$  e  $\mathcal{M}, s \models \phi_2$
5.  $\mathcal{M}, s \models \phi_1 \vee \phi_2$  sse  $\mathcal{M}, s \models \phi_1$  ou  $\mathcal{M}, s \models \phi_2$
6.  $\mathcal{M}, s \models \phi_1 \rightarrow \phi_2$  sse  $\mathcal{M}, s \not\models \phi_1$  ou  $\mathcal{M}, s \models \phi_2$
7.  $\mathcal{M}, s \models AX\phi$  sse para todos os estados  $s_1$  tal que  $s \rightarrow s_1$  temos que  $\mathcal{M}, s_1 \models \phi$ . Assim,  $AX$  se diz: "em todo estado seguinte".
8.  $\mathcal{M}, s \models EX\phi$  sse para algum estado  $s_1$  tal que  $s \rightarrow s_1$  temos que  $\mathcal{M}, s_1 \models \phi$ . Assim,  $EX$  se diz: "em algum próximo estado".

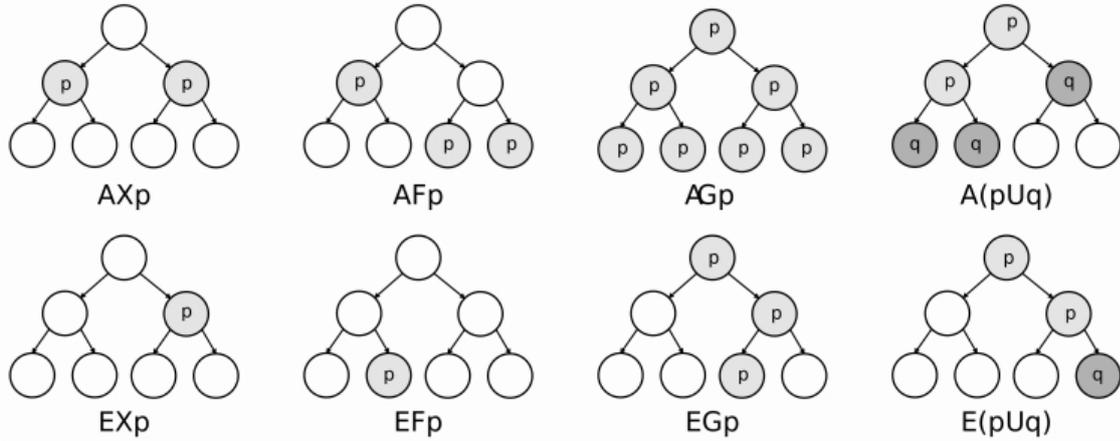
<sup>1</sup>Formalismo de Backus-Naur é uma metassintaxe utilizada para expressar gramáticas livres de contexto, sendo ela a notação mais usual para descrever gramáticas de linguagens de programação.



**Figura B.1** Estrutura de Kripke e árvore de computação infinita

9.  $\mathcal{M}, s \models AG\phi$  sse para todos os caminhos  $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \dots$ , onde  $s_1$  é igual a  $s$ , e para todos os estados  $s_i$  ao longo do caminho, temos que  $\mathcal{M}, s_i \models \phi$ . Mnemonicamente: para todos os caminhos da computação iniciados em  $s$ , a propriedade  $\phi$  é mantida globalmente.
10.  $\mathcal{M}, s \models EG\phi$  sse existe um caminho  $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \dots$ , onde  $s_1$  é igual a  $s$ , e para todos os estados  $s_i$  ao longo do caminho, temos que  $\mathcal{M}, s_i \models \phi$ . Mnemonicamente: existe um caminho que começa em  $s$  tal que  $\phi$  se mantém globalmente ao longo do caminho.
11.  $\mathcal{M}, s \models AF\phi$  sse ao longo de qualquer caminho  $s_1 \rightarrow s_2 \rightarrow s_3 \dots$ , onde  $s_1$  é igual a  $s$  existe algum  $s_i$  tal que  $\mathcal{M}, s_i \models \phi$ . Mnemonicamente: para todos os caminhos da computação começando em  $s$ , haverá algum estado futuro onde  $\phi$  será verdade.
12.  $\mathcal{M}, s \models EF\phi$  sse existe um caminho  $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \dots$ , onde  $s_1$  é igual a  $s$ , e para algum  $s_i$  ao longo do caminho, temos que  $\mathcal{M}, s_i \models \phi$ . Mnemonicamente: existe um caminho da computação começando em  $s$  tal que  $\phi$  ocorre em algum estado futuro.
13.  $\mathcal{M}, s \models A[\phi_1 U \phi_2]$  sse para todos os caminhos  $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \dots$ , onde  $s_1$  é igual a  $s$ , o caminho satisfaz  $\phi_1 U \phi_2$ , ou seja, todos os caminhos da computação que começam em  $s$  tal que  $\phi_1$  é verdade até  $\phi_2$  ser verdade.
14.  $\mathcal{M}, s \models E[\phi_1 U \phi_2]$  sse existe um caminho  $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \dots$ , onde  $s_1$  é igual a  $s$ , que satisfaz  $\phi_1 U \phi_2$ , ou seja, existe um caminho na computação que começa em  $s$  tal que  $\phi_1$  é verdade até  $\phi_2$  ser verdade.

A interpretação dos operadores é ilustrada na Figura B.2.



**Figura B.2** Operadores temporais da lógica CTL.  
**Fonte:** Adaptado de Paulo (2010)

## B.2 LÓGICA DE ÁRVORE DE COMPUTAÇÃO PROBABILÍSTICA

A lógica de árvore de computação probabilística PCTL é utilizada por muitos verificadores de modelos. PCTL é uma extensão de CTL, em que é adicionado o operador  $P$  de probabilidade. O operador probabilístico  $P$  substitui os quantificadores de caminho  $A$  e  $E$ . Este é utilizado para descrever propriedades das cadeias de Markov em tempo discreto (DTMC, *Discrete Time Markov Chains*).

DTMC é um formalismo de modelagem bem conhecido para sistemas probabilísticos, em que os estados representam possíveis configurações do sistema. As transições entre estados ocorrem em tempo discreto e têm uma probabilidade associada. Os modelos das cadeias de Markov em tempo discreto representam processos estocásticos discretos, segundo os quais a distribuição de probabilidade de estados futuros depende apenas do estado atual (FILIERI; GHEZZI; TAMBURRELLI, 2011).

Erika et al. (2010) definem DTMC como uma tupla  $M = (S, P, L, AP)$ , sendo:

- $S$  o conjunto finito não vazio de estados.
- $P : S \times S \rightarrow [0, 1]$  é uma matriz de Markov com  $\sum_{s' \in S} P(s, s') = 1$  para todos estados  $s \in S$ .
- $AP$  é o conjunto de propriedades atômicas.
- $L : S \rightarrow 2^{AP}$  é uma função que rotula cada estado com um conjunto de proposições atômicas verdadeiras naquele estado.

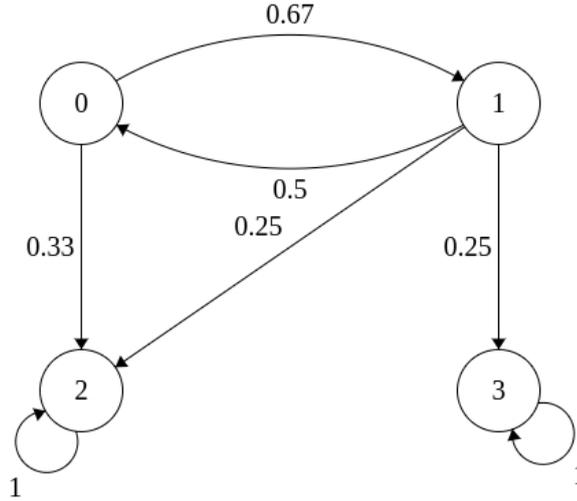
$P$  tem as seguintes propriedades:

$$0 \leq P(s, s') \text{ para todo } s, s' \in S \text{ - equação 1}$$

$$\sum_{s, s' \in S} P(s, s') = 1 \text{ para qualquer } s, s' \in S \text{ - equação 2}$$

No exemplo de um DTMC  $\mathcal{M} = (S, P, AP, L)$  da Figura B.3, é possível derivar a matriz de probabilidade  $P$ .

$$P = \begin{bmatrix} 0.0 & 0.67 & 0.33 & 0.0 \\ 0.5 & 0.0 & 0.25 & 0.25 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$



**Figura B.3** Exemplo de uma DTMC.

$P(0,1) = 0.67, P(0,2) = 0.33, P(1,0) = 0.5, P(1,2) = 0.25, P(1,3) = 0.25, P(2,2) = 1, P(3,3) = 1.0$ . De acordo a **equação 1**, todos os valores da matriz são maiores ou iguais a zero, e menores ou iguais a 1. De acordo a **equação 2**,  $P(0,1) + P(0,2) = 1$  e  $P(1,0) + P(1,2) + P(1,3) = 1$  e  $P(2,2) = 1$  e  $P(3,3) = 1$

As formulas de estado e caminho são apresentadas separadamente, conforme indicado nas definições a seguir.

**Definição 1** (Fórmula de estado de PCTL). Uma fórmula de estado em PCTL pode ser definida como indicado abaixo:

$$\Phi ::= \alpha \mid \neg\Phi \mid \Phi \vee \Phi \mid \Phi \wedge \Phi \mid \mathcal{P} \triangleq p(\varphi)$$

onde  $\alpha$  é uma proposição atômica de AP. Utilizam-se operadores *booleanos*  $\neg, \vee, \wedge$ , para combinar fórmulas de estado.  $p$  é um valor de probabilidade,  $\triangleq$  representa um operador de comparação, onde pode ser substituído por  $\leq, <, \geq$  e  $>$ .  $\mathcal{P} \triangleq p(\varphi)$  indica que a probabilidade para alcançar  $\varphi$  é  $\triangleq p$ .

**Definição 2** (Fórmula de caminho de PCTL). Uma fórmula de caminho em PCTL pode ser definida como:

$$\phi ::= X\Phi \mid \Phi U^{\leq k} \Psi$$

$X \Phi$  significa que o próximo estado no caminho irá satisfazer  $\Phi$ .  $\Phi U^{\leq k} \Psi$  significa que  $\Phi$  é verdade em todos os caminhos até que  $\Psi$  seja alcançado em  $k$  mudanças de estado ou menos. Quando  $k$  não é indicado,  $k$  assume o valor  $\infty$ .

Segue abaixo um exemplo de uma fórmula PCTL:

$\alpha$  pode ser alcançado com probabilidade de 1% em pelo menos 5 saltos ao longo do caminho.  

$$P \leq 0.01(\text{true} U^{\leq 5} \alpha)$$

**Definição 3** (Relação de satisfabilidade para PCTL). Sendo  $a \in AP$  uma proposição atômica,  $\mathcal{M} = (S, P, AP, L)$  um DTMC,  $s \in S$  um estado,  $\Phi, \Psi$  fórmulas de estado PCTL e  $\varphi$  fórmula de caminho. A relação de satisfação  $\models$  é definida sobre fórmulas de estado e fórmulas de caminho como segue:

- $\mathcal{M}, s \models \text{true}$  para todo  $s \in S$  - Significa que a fórmula é satisfeita em qualquer estado  $s$ .
- $\mathcal{M}, s \models a$  sse  $a \in L(s)$  - Uma fórmula atômica  $a$  é satisfeita em um estado  $s$  se e somente se o rótulo  $a$  estiver presente no conjunto de rótulos  $L(s)$  associado a esse estado  $s$ .
- $\mathcal{M}, s \models \Phi \wedge \Psi$  sse  $s \models \Phi \wedge s \models \Psi$ . - A conjunção de duas fórmulas  $\Phi$  e  $\Psi$  é satisfeita em um estado  $s$  se e somente se ambas as fórmulas  $\Phi$  e  $\Psi$  forem satisfeitas em  $s$ .
- $\mathcal{M}, s \models \neg\Phi$  sse  $s \not\models \Phi$  - A negação de uma fórmula  $\Phi$  é satisfeita em um estado  $s$  se e somente se  $\Phi$  não é satisfeita em  $s$ .
- $\mathcal{M}, s \models \mathcal{P} \triangleq p(\varphi)$  sse  $Pr\{\pi \in Paths(s) | \pi \models \varphi\} \geq p$  - A relação de satisfação é expressa como  $\mathcal{M}, s \models \mathcal{P} \triangleq p(\varphi)$ , que significa que a propriedade  $\mathcal{P} \triangleq p(\varphi)$  é satisfeita no estado  $s$  do modelo  $\mathcal{M}$  se e somente se a probabilidade de que exista um caminho  $\pi$  a partir de  $s$ , tal que  $\pi \models \varphi$ , seja maior ou igual a  $p$ .

## MODELAGEM DO PBCP EM PRISM

O Protocolo Básico de Comunicação entre Pelotões foi desenvolvido com o propósito de identificar pelotões vizinhos em uma Rede Veicular *Ad Hoc*. A modelagem desse protocolo utiliza a linguagem PRISM e o modelo probabilístico DTMC (Cadeia de Markov de Tempo Discreto). Para simplificar essa especificação, consideramos dois líderes, visando analisar o comportamento da detecção de vizinhança com diferentes taxas de perdas de mensagens.

### C.1 MODELAGEM DO PROTOCOLO EM PRISM

```

1  - // MODELO PROBABILISTICO UTILIZADO – DISCRETE TIME MARKOV CHAIN.
2  -     dtmc
3  -
4  - // FORMULAS UTILIZADAS PARA: DEFINIR A MUDANCA DE ACAO DE ENVIO E
5  - // RECEBIMENTO DAS MENSAGENS; DEFINIR QUAL LIDER IRA EXECUTAR
6  - // A ACAO E INFORMAR ATUALIZAR A VISAO DO LIDER, RESPECTIVAMENTE
7  -
8  -     formula mudaAcao = mudancaDoTurnoDoLider=L;
9  -     formula turnoDoLider = mudancaDoTurnoDoLider=i;
10 -     formula dadosDaVisao = mudancaDoTurnoDoLider=1 ? 1:2;
11 -
10 - // CONSTANTES USADAS PARA CONFIGURAR O COMPORTAMENTO DO SISTEMA.
11 -     const double taxaDePerdaDeMensagens;
12 -     const int tempoLimite= 1;
13 -     const int L = 3;
14 -     const int i = 1;
15 -     const int j = 2;
16 -     const int enviar = 0;
17 -     const int receber = 1;
18 -     const int semInformacao = 0;
19 -     const int contSync = 10;
20 -
21 -     global tarefa : [enviar..receber] init 0;

```

```

22 -     global mudancaDoTurnoDoLider : [1..L] init 1;
23 -     global contador : [1..11] init 1;
24 -     global verificaVizinhanca : bool init false;
25 -     global bidirecional : bool init false;
26 -     global etapaManutencao : bool init false;
27 -     global variavelDeGiro : [0..1] init 0;
28 -     global vizinhancaDetectada : bool init false;
29 -
30 -     module lider_i
31 -         cronometro_i : [0..2] init 0;
32 -         visoesAcordadas_i : bool init false;
33 -
34 -     // VARIAVEIS UTILIZADAS PARA DEFINIR O QUE UM LIDER DE PELOTAO SABE DE SUA
35 -     // VISAO E DA VISAO DO OUTRO PELOTAO
36 -
37 -         visao_i : [0..2] init 0;
38 -         visoesConhecidas_i : [0..2] init 0;
39 -         visaoRecebida_i : [0..2] init 0;
40 -         visoesConhecidasRecebidas_i : [0..2] init 0;
41 -
42 -     // TAREFA T1 - RESPONSAVEL PELO ENVIO DE MENSAGENS. QUANDO O LIDER ESTA NO
43 -     // SEU TURNO (turnodolider) E ENVIA MENSAGEM (tarefa=enviar) AS SEGUINTE
44 -     // ATUALIZACOES SAO REALIZADAS: A VISAO DO LIDER E ATUALIZADA PARA A VISAO
45 -     // ATUAL (visao_i'=dadosdavisao), AS VISOES CONHECIDAS SAO ATUALIZADAS COM
46 -     // AS VISOES RECEBIDAS (visoesconhecidas_i'=visaorecebida_i) E A MUDANCA
47 -     // DO TURNO DO LIDER E ATUALIZADA
48 -     // (mudancadoturnodolider=mudancadoturnodolider+1).
49 -
50 -         [] turnoDoLider & tarefa=enviar
51 -         ->
52 -         (visao_i'=dadosDaVisao) & (visoesConhecidas_i'=visaoRecebida_i) &
53 -         (mudancaDoTurnoDoLider'=mudancaDoTurnoDoLider+1);
54 -
55 -     // TAREFA 2 - RESPONSAVEL PELO RECEBIMENTO DE MENSAGENS. QUANDO O LIDER
56 -     // ESTA NO SEU TURNO (turnoDoLider) E RECEBE MENSAGEM (tarefa=receber) E O
57 -     // CRONOMETRO NAO ATINGIU O LIMITE DE TEMPO (cronometro_i<tempoLimite),
58 -     // COM UMA PROBABILIDADE (1-taxaDePerdaDeMensagens) AS SEGUINTE
59 -     // ATUALIZACOES SAO REALIZADAS: O CRONOMETRO E ATUALIZADO
60 -     // (cronometro_i'=cronometro_i+1), A VISAO RECEBIDA E ATUALIZADA COM A VISAO
61 -     // DO LIDER VIZINHO (visaoRecebida_i' = visao_j) E AS VISOES CONHECIDAS QUE
62 -     // FORAM RECEBIDAS E ATUALIZADA COM A INFORMACOES DE VISOES CONHECIDAS
63 -     // (visoesConhecidasRecebidas_i'=visoesConhecidas_j). COM UMA PROBABILIDADE
64 -     // (taxaDePerdaDeMensagens) AS VARIAVEIS (visaoRecebida_i) E
65 -     // (visoesConhecidasRecebidas_i) NAO SAO ATUALIZADAS, INDICANDO PERDA
66 -     // DE MENSAGENS.
67 -
68 -         [] turnoDoLider & tarefa=receber & cronometro_i<tempoLimite

```

```

48 -     ->
49 -     (1-taxaDePerdaDeMensagens) : (cronometro_i'=cronometro_i+1) &
50 -     (visaoRecebida_i' = visao_j) &
51 -     (visoesConhecidasRecebidas_i'=visoesConhecidas_j) &
52 -
53 -     (mudancaDoTurnoDoLider'=mudancaDoTurnoDoLider+1)
54 -     +
55 -     (taxaDePerdaDeMensagens) : (cronometro_i'=cronometro_i+1) &
56 -     (mudancaDoTurnoDoLider'=mudancaDoTurnoDoLider+1);
57 -
58 -     [] turnoDoLider & tarefa=receber & cronometro_i=tempoLimite &
59 -     etapaManutencao=false
60 -     ->
61 -     (1-taxaDePerdaDeMensagens) : (etapaManutencao'=true) &
62 -     (visaoRecebida_i' = visao_j) &
63 -     (visoesConhecidasRecebidas_i'=visoesConhecidas_j)
64 -     +
65 -     (taxaDePerdaDeMensagens) : (etapaManutencao'=true);
66 -
67 - // ETAPA DE MANUTENCAO. QUANDO O LIDER ESTA NA ETAPA DE MANUTENCAO
68 - // (etapaManutencao=true ) E A VISAO CONHECIDA RECEBIDA POSSUI A
69 - // INFORMACAO DA VISAO DO LIDER DESTINATARIO, A VARIAVEL
70 - // vizinhancaDetectada E ATUALIZADA COM VALOR BOLEANO TRUE INDICANDO
71 - // QUE HOVE DETECCAO DE VIZINHANCA (vizinhancaDetectada'=true ).
72 -
73 -     [] turnoDoLider & etapaManutencao=true &
74 -     visoesConhecidasRecebidas_i=visao_i
75 -     ->
76 -     (vizinhanca_i' = true) & (etapaManutencao'=false) &
77 -     (mudancaDoTurnoDoLider'=mudancaDoTurnoDoLider+1) &
78 -     (vizinhancaDetectada'=true) & (cronometro_i'=0);
79 -
80 -     [] turnoDoLider & etapaManutencao=true &
81 -     visoesConhecidasRecebidas_i!=visao_i
82 -     ->
83 -     (visoesAcordadas_i' = false) & (etapaManutencao'=false) &
84 -     (mudancaDoTurnoDoLider'=mudancaDoTurnoDoLider+1) &
85 -     (vizinhancaDetectada'=true) & (cronometro_i'=0);
86 -
87 -
88 -     [] turnoDoLider & tarefa=enviar & cronometro_i=tempoLimite &
89 -     contador > contSync & verificaVizinhanca=false
90 -     ->
91 -     (mudancaDoTurnoDoLider'=mudancaDoTurnoDoLider+1) &
92 -     (visaoRecebida_i'=semInformacao) & (visao_i'=semInformacao) &
93 -     (visoesConhecidas_i'=semInformacao) &
94 -     (visoesConhecidasRecebidas_i'=semInformacao) &

```

```

95 -      (visoesAcordadas_i != false) &
96 -      (cronometro_i != semInformacao) &
97 -      (bidirecional != false);
98 -
99 - endmodule
100 -
101 - module mudancaDeAcao
102 -
103 - // UTILIZADO PARA A REALIZACAO DE MUDANCA DE ESTADOS
104 -
105 -      [] mudaAcao & tarefa = enviar & variavelDeGiro=0 &
106 -      verificaVizinhanca=false
107 -      ->
108 -      (tarefa ' = receber) &
109 -      (mudancaDoTurnoDoLider'=i);
110 -
111 -      [] mudaAcao & tarefa = enviar & variavelDeGiro=1 &
112 -      contador <= contSync
113 -      ->
114 -      (verificaVizinhanca != false) &
115 -      (mudancaDoTurnoDoLider'=i) &
116 -      (variavelDeGiro '=0);
117 -      [] mudaAcao & tarefa = enviar & contador > contSync
118 -      ->
119 -      (mudancaDoTurnoDoLider'=i) & (contador '=1);
120 -
121 -      [] mudaAcao & tarefa = receber & verificaVizinhanca=false &
122 -      vizinhancaDetectada=false & contador <=10
123 -      ->
124 -      (tarefa '=enviar) & (mudancaDoTurnoDoLider'=i);
125 -
126 -      [] mudaAcao & tarefa = receber & verificaVizinhanca=false &
127 -      vizinhancaDetectada=false & contador >10
128 -      ->
129 -      (tarefa '=enviar) & (mudancaDoTurnoDoLider'=i) & (contador '=1);
130 -
131 - // UTILIZADO PARA CONTABILIZACAO DAS RECOMPENSAS
132 -
133 -      [] mudaAcao & visoesAcordadas_i=true &
134 -      visoesAcordadas_j=true & vizinhancaDetectada=true
135 -      ->
136 -      (bidirecional '=true) & (vizinhancaDetectada '=false) &
137 -      (verificaVizinhanca '=true);
138 -
139 -      [] bidirecional=true & vizinhancaDetectada=false &
140 -      contador <= contSync & verificaVizinhanca=true
141 -      ->

```

```

142-     ( tarefa !=enviar ) & ( mudancaDoTurnoDoLider'=i ) &
143-     ( contador !=contador+1 ) & ( bidirecional !=false ) &
144-     ( verificaVizinhanca !=false );
145-
146-     [] mudaAcao & ( visoesAcordadas_i!=true |
147-     visoesAcordadas_j!=true ) & vizinhancaDetectada=true
148-     ->
149-     ( bidirecional !=false ) & ( vizinhancaDetectada !=false ) &
150-     ( verificaVizinhanca !=true );
151-
152-     [] bidirecional=false & vizinhancaDetectada=false &
153-     contador <= contSync & verificaVizinhanca=true
154-     ->
155-     ( tarefa !=enviar ) & ( mudancaDoTurnoDoLider'=i ) &
156-     ( contador !=contador+1 ) & ( verificaVizinhanca !=false );
157-
158-     endmodule
159-
160- // MODULO QUE PERMITE MODELAR A INTERACAO ENTRE DOIS LIDERES.
161-
162-     module lider_j=lider_i
163-     [ i=j, cronometro_i=cronometro_j,
164-     visaoRecebida_i=visaoRecebida_j,
165-     visaoRecebida_j=visaoRecebida_i,
166-     visao_j=visao_i, visao_i=visao_j,
167-     visoesConhecidas_i=visoesConhecidas_j,
168-     visoesConhecidas_j=visoesConhecidas_i,
169-     visoesAcordadas_j=visoesAcordadas_i,
170-     visoesAcordadas_i=visoesAcordadas_j,
171-     visoesConhecidasRecebidas_i=visoesConhecidasRecebidas_j ]
172-     endmodule
173-     rewards   bidirecional=true : 1; endrewards

```