



Universidade Federal da Bahia
Instituto de Computação & Escola Politécnica

Programa de Pós-Graduação em Mecatrônica

**PROTOCOLO DE COMUNICAÇÃO EM
GRUPO CONFIÁVEL E DE TEMPO REAL
PARA APLICAÇÕES DE COORDENAÇÃO
VEICULAR SOBRE REDES VEICULARES AD
HOC**

Marcela Alves Pereira

DISSERTAÇÃO DE MESTRADO

Salvador
15 de Setembro de 2023

MARCELA ALVES PEREIRA

**PROTOCOLO DE COMUNICAÇÃO EM GRUPO CONFIÁVEL E
DE TEMPO REAL PARA APLICAÇÕES DE COORDENAÇÃO
VEICULAR SOBRE REDES VEICULARES AD HOC**

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

Orientador: Alirio Santos de Sá

Salvador
15 de Setembro de 2023

P436 Pereira, Marcela Alves.

Protocolo de comunicação em grupo confiável e de tempo real para aplicações de coordenação veicular sobre redes veiculares ad hocs / Marcela Alves Pereira. – Salvador, 2023.

95f.: il.

Orientador: Prof. Dr. Alirio Santos de Sá.

Dissertação (mestrado) – Universidade Federal da Bahia. Instituto de Computação e Escola Politécnica, 2023.

1. Redes veiculares ad hocs. 2. Protocolo de comunicação em grupo. 3. Sistemas distribuídos. I. de Sá, Alirio Santos. II. Universidade Federal da Bahia. III. Título.

CDD: 003.78

TERMO DE APROVAÇÃO

Marcela Alves Pereira

Protocolo de Comunicação em Grupo Confiável e de Tempo Real para Aplicações de Coordenação Veicular sobre Redes Veiculares Ad Hoc.

Esta Dissertação de Mestrado foi julgada adequada à obtenção do título de Mestre em Mecatrônica e aprovada em sua forma final pelo Programa de Pós-Graduação em Mecatrônica da Universidade Federal da Bahia.

Salvador, 15 de setembro de 2023.

Documento assinado digitalmente
 ALIRIO SANTOS DE SA
Data: 15/09/2023 18:08:53-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Alírio Santos de Sá
(UFBA)

Documento assinado digitalmente
 RAIMUNDO JOSE DE ARAUJO MACEDO
Data: 20/09/2023 06:13:29-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Raimundo José de Araujo Macedo
(UFBA)

Documento assinado digitalmente
 WEVERTON LUIS DA COSTA CORDEIRO
Data: 19/09/2023 14:03:39-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Weverton Luis da Costa Cordeiro
(UFRGS)

Dedico este trabalho à minha família, em especial aos meus pais: Romenildo Freire Pereira (in memoriam), Manoel Messias Vieira Santos e Maria Maura Alves Pereira. Expresso todo o meu amor e gratidão por tudo o que fizeram por mim ao longo da minha vida. Desejo ter sido merecedora do esforço dedicado por vocês em todos os aspectos, especialmente no que diz respeito à minha formação.

AGRADECIMENTOS

A Deus por ter me fortalecido durante este projeto de pesquisa, reestabelecendo a minha saúde e as minhas forças para chegar até o final.

À minha família, pelo apoio que sempre me deram ao longo de toda a minha vida. Em especial, ao meu filho Davi Pereira Roque e à minha filha de coração, Érica Silva Neves, que sempre compreenderam as minhas ausências devido às viagens e aos estudos.

Ao meu Pai, Romenildo Freire Pereira (*in memoriam*) por ter me ensinado, desde pequena, o valor do estudo e da perseverança.

Aos meus pais, Maria Maura Alves Pereira e Manoel Messias Vieira Santos, pelo incentivo e apoio em todos os momentos da minha vida e por terem sido, muitas vezes, o porto seguro para mim e para o meu filho. Além de terem trabalhado e lutado que eu tivesse oportunidade de me instruir.

À Margarete Sá, por se preocupar comigo, por acolher a minha família, por me ensinar conteúdos tão complexos, além de me incentivar nos momentos mais difíceis.

Ao meu colega Luís Carvalho que, em uma palestra, me fez compreender que ultrapassar o tempo de um curso não significava derrota, mas, sim, uma oportunidade para crescer e aprender a valorizar os desafios impostos pela vida.

Ao meu amigo e parceiro Cleber Brito Santos pelos inúmeros momentos de apoio e colaboração.

A todos os colegas e professores da UFBA, pelos ensinamentos e momentos de descontração.

À minha amiga e irmã de coração, Maria Sousa Amaral, pelos incentivos e orações nos momentos mais difíceis da pesquisa.

Aos meus grandes mestres Wilton Lacerda e Marco Antônio Andrade pelo incentivo e confiança. Sem vocês, eu não teria ingressado neste mestrado.

À Amanda Ferraz pelas orações, carinho, incentivo e amizade nestes anos de pesquisa e de trabalho na docência.

Ao IFBA, a todos os professores da minha área e aos alunos pela colaboração e paciência com o meu processo de pesquisa.

Ao Miro Conceição pelo apoio e incentivo nos momentos mais difíceis da pesquisa e da minha vida pessoal. Sem o seu companheirismo, carinho e dedicação, este sonho não seria realizado.

Em especial, ao meu orientador Alirio Sá, que apesar da intensa rotina de sua vida acadêmica, acreditou em mim, quando eu estava passando por um momento delicado da minha saúde, para desenvolver o trabalho de pesquisa. Jamais esquecerei a sua paciência, dedicação, sugestões e conselhos.

Por fim, agradeço a todos os meus amigos e pessoas que de alguma maneira contribuíram para que este trabalho fosse concluído.

RESUMO

As redes veiculares *Ad Hoc* (VANETs, *Vehicular Ad hoc Networks*) constituem um tipo específico de redes móveis cujos nós são veículos capazes de enviar e receber mensagens diretamente entre si usando enlaces de comunicação sem fio. Essas redes oportunizam a criação de diversas aplicações distribuídas sem a dependência de uma infraestrutura específica para a comunicação entre os veículos. Em ambientes urbanos e rodovias, as VANETs têm sido vistas como uma oportunidade na construção de Sistemas de Transporte Inteligentes (ITS, *Intelligent Transportation Systems*), voltados não apenas para a melhoria da fluidez, segurança e bem-estar no trânsito, mas também na redução da emissão de poluentes. Todavia, muitas aplicações distribuídas para ITS requerem que grupos de veículos coordenem suas ações de forma consistente, confiável e com restrições temporais. Contudo, em VANETs constituídas em ambientes urbanos e rodovias, os veículos trafegam por vias que demarcam um padrão de mobilidade e que apresentam uma estrutura de comunicação altamente dinâmica. Portanto, as VANETs são propensas a mudanças constantes em sua topologia, desconexões frequentes na comunicação entre veículos e perdas de mensagens nos meios de comunicações. Por isso, atender aos requisitos de coordenação veicular em aplicações distribuídas de ITS baseado em VANETs é um desafio. Nesse contexto, este trabalho propõe o Protocolo para Comunicação Veicular Confiável Baseado em Blocos Causais (*VCBP*, *Vehicular Causal Block Protocol*) para dar suporte aos requisitos de coordenação veicular em aplicações distribuídas de ITS baseados em VANETs. O protocolo de comunicação em grupo proposto oferece facilidade na difusão de mensagens com garantias de entrega confiável, com restrições temporais e ordenação causal e total das mensagens. O VCBP foi implementado e avaliado em um ambiente de simulação de redes veiculares e o seu desempenho foi comparado com outros protocolos disponíveis na literatura. Os resultados mostram que o *VCBP* oferece mais garantias na entrega das mensagens, com uma taxa de entrega de mensagens semelhante ou superior às demais abordagens avaliadas.

Palavras-chave: Redes Veiculares Ad Hoc, Sistemas Distribuídos, Comunicação em Grupo, Tempo Real, Difusão Confiável

ABSTRACT

Vehicular Ad hoc Networks (VANETs) constitute a specific type of mobile ad hoc network in which nodes are vehicles capable of sending and receiving messages directly between each other using wireless communication links, enabling the creation of several distributed applications without the dependence on a specific infrastructure for vehicle communication. In urban and highway environments, VANETs have been seen as an opportunity in the development of Intelligent Transportation Systems (ITS), aiming not only to improve traffic flow, security, and well-being but also to reduce pollutant emissions. However, many distributed applications for ITS require groups of vehicles to coordinate their actions consistently, reliably, and with temporal constraints. Nevertheless, in VANETs formed in urban and highway environments, vehicles travel along pathways exhibiting a mobility pattern and highly dynamic communication structure. Therefore, network topology changes, communication link disconnections, and message losses are frequent in a VANET. Hence, meeting the requirements for vehicle coordination in distributed ITS applications based on VANETs represents a challenge. In this context, this work proposes the Vehicular Causal Block Protocol (VCBP) to support the vehicle coordination requirements in distributed ITS applications based on VANETs. The proposed group communication protocol offers the facility of message multicast with guarantees of reliable delivery, temporal constraints, and causal and total ordering of messages. VCBP has been implemented and evaluated in a vehicular network simulation environment, and its performance has been compared with other protocols available in the literature. The results show that VCBP can provide enhanced message delivery guarantees with a message delivery rate similar to or better than other evaluated approaches.

Keywords: Vehicular Ad Hoc Networks, Distributed Systems, Group Communication, Real-time, Reliable Multicast

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Visão Geral	1
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	2
1.2 Publicações	3
1.3 Organização da Dissertação	3
Capítulo 2—Redes Veiculares: Conceitos Básicos, Padrões e Aplicações	5
2.1 Introdução	5
2.2 Redes Veiculares Ad Hoc	6
2.2.1 Arquitetura	6
2.2.2 Padrões de Comunicação	7
2.2.3 Protocolos de Comunicação	9
2.3 Aplicações baseadas em Redes Veiculares	11
2.3.1 Pelotões Veiculares	13
2.4 Sistemas de Transportes Inteligentes	14
2.4.1 Arquiteturas de ITS	14
Capítulo 3—Comunicação em grupo em Redes Não Confiáveis	17
3.1 Introdução	17
3.2 Aspectos Básicos sobre Comunicação em Grupo	17
3.2.1 Classificação dos Grupos em Sistemas Distribuídos	17
3.2.2 Serviço Membership	18
3.2.3 Serviço de Multicast	18
3.3 Comunicação em Grupo em Móveis sem Fio	20
3.3.1 Δ -CBP: Protocolo para Redes Móveis Sem Fio	20
3.3.2 Protocolos para Redes Veicular Ad Hocs	21
3.3.2.1 <i>M-RBP</i>	21
3.3.2.2 RNP.	23
3.3.2.3 FSBP.	23
3.3.2.4 Vertigo.	25
3.3.3 Protocolos para Redes Veiculares Infraestruturadas	25

Capítulo 4—VCBP: Protocolo de Comunicação em Grupo para VANETs	27
4.1 Introdução	27
4.2 Trabalhos Relacionados	30
4.3 Modelo de Sistema	31
4.4 Visão Geral do Modelo de Blocos Causais	33
4.5 Visão Geral do VCBP	34
4.5.1 Garantindo Integridade, Ordem Causal, Ordem Total e Restrição Temporal na Entrega de Mensagens	35
4.5.2 Minimizando Perdas de Mensagens nos Canais de Comunicação	35
4.5.3 Garantindo Acordo na Entrega das Mensagens	37
4.6 Detalhamento da Troca de Mensagens do VCBP	37
4.7 Detalhamento do Algoritmo Proposto no VCBP	41
4.7.1 Principais Tarefas Implementadas no VCBP	41
4.7.2 Principais Operações do VCBP	42
4.7.3 Funções e Procedimentos Auxiliares Importantes para o VCBP	44
Capítulo 5—Avaliação de Desempenho	47
5.1 Descrição do Ambiente de Simulação	47
5.2 Configuração dos Protocolos de Comunicação em Grupo	49
5.3 Métricas de Desempenho	50
5.4 Experimentos	51
5.4.1 Cenário 1: Comunicação Veicular sem Interferências	51
5.4.2 Cenário 2: Comunicação Veicular com Interferência	53
Capítulo 6—Considerações Finais	57
Apêndice A—Acoplamento dos Simuladores	65
Apêndice B—Diagramas De Classe	67
B.1 RCBP	67
B.2 VCBP	68
Apêndice C—Códigos-fonte do VCBP	69
C.1 Função Init	69
C.2 Função Beaconing	70
C.3 Função handleSelfMsg	70
C.4 Função rmcst	71
C.5 Função receive	72
C.6 Função retransmission	73
C.7 Função delivery	74

LISTA DE FIGURAS

2.1	Cenário de uma rede Veicular (ALVES et al., 2009)	7
2.2	Alocação de espectro para aplicações DSRC (ALVES et al., 2009)	8
2.3	Pilha de protocolos WAVE (ALVES et al., 2009)	9
2.4	Intervalo de sincronização dos canais CCH e SCH. Adaptado de Chen, Jiang e Delgrossi (2009))	10
2.5	Colisões laterais na rodovia. Adaptado de Hartenstein e Laberteaux (2009)	11
2.6	Redes veiculares ad hocs usadas em sistemas de transporte de emergência. Adaptado de Buchenscheit et al. (2009)	12
2.7	Vários pelotões no cenário de uma rodovia. Adaptado de Jia et al. (2015)	13
3.1	Sobreposição de grupos em uma vizinhança (MAXEMCHUK, 2009).	23
3.2	Dois exemplos de processos abortados (GU; LIN; MAXEMCHUK, 2015)	24
4.1	Exemplos de carros de um pelotão realizando ultrapassagem com colisão em virtude da perda de mensagens e da falta de visibilidade. Adaptado de (HARTENSTEIN; LABERTEAUX, 2009)	28
4.2	Exemplo de carros de um pelotão realizando ultrapassagem segura a partir da utilização do VCBP. Adaptado de (HARTENSTEIN; LABERTEAUX, 2009)	29
4.3	Exemplo de matriz de blocos com 6 processos (MACÊDO, 1994)	34
4.4	Disseminação da Primeira Mensagem do Grupo Utilizando o VCBP	38
4.5	Disseminação da Segunda Mensagem do Grupo Utilizando o VCBP	39
4.6	Disseminação da Terceira Mensagem do Grupo Utilizando o VCBP	40
4.7	Disseminação da Quarta Mensagem do Grupo Utilizando o VCBP	40
5.1	Desempenho dos protocolos em termos da latência de recebimento e da taxa de recebimento de mensagens nos experimentos sem interferência na comunicação veicular	51
5.2	Desempenho dos protocolos em termos da latência de entrega e da taxa de entrega de mensagens nos experimentos sem interferência na comunicação veicular	52
5.3	Desempenho dos protocolos em termos da taxa de mensagens recebidas, da taxa de mensagens entregues, da latência de recebimento de mensagens e da latência de entrega de mensagens nos experimentos com interferência na comunicação veicular e pelotões compostos por 2 veículos	53

5.4	Desempenho dos protocolos em termos da taxa de mensagens recebidas, da taxa de mensagens entregues, da latência de recebimento de mensagens e da latência de entrega de mensagens nos experimentos com interferência na comunicação veicular e pelotões compostos por 4 veículos	53
5.5	Desempenho dos protocolos em termos da taxa de mensagens recebidas, da taxa de mensagens entregues, da latência de recebimento de mensagens e da latência de entrega de mensagens nos experimentos com interferência na comunicação veicular e pelotões compostos por 8 veículos	54
A.1	Visão geral da estrutura de simulação acoplada do VEINS Adaptado de (SOMMER; GERMAN; DRESSLER, 2010)	65
B.1	Diagrama de Classe do RCBP	67
B.2	Diagrama de Classe do VCBP	68

LISTA DE TABELAS

4.1	Avaliação dos Requisitos.	31
5.1	Parâmetros de configuração de rede	48
5.2	Parâmetros de configuração do serviço de <i>Beaconing</i>	48
5.3	Parâmetros de configuração de mobilidade	49
5.4	Parâmetros de configuração dos protocolos de comunicação em grupo	50

INTRODUÇÃO

1.1 VISÃO GERAL

O crescente aumento da frota de veículos nas vias urbanas e rodovias impõe desafios na prevenção de acidentes, na melhoria do tráfego e na redução da emissão de poluentes. O enfrentamento desses desafios requer a integração de políticas públicas, políticas de transporte eficientes e inovações tecnológicas.

Dentre as políticas de trânsito, destacam-se as recomendações efetuadas pela Organização das Nações Unidas (ONU) para a implementação de ações com o objetivo de aumentar a segurança no trânsito em todos os países membros. Essa iniciativa foi formulada em decorrência do alto índice de acidentes de trânsito. A redução dos acidentes graves foi uma das principais motivações para estas recomendações porque a cada ano morrem, em média, 1.3 milhão de pessoas em todo mundo e cerca de 50 milhões de pessoas, que sobrevivem dos acidentes, ficam feridas ou com alguma seqüela. Esses altos índices são, geralmente, decorrentes de erros humanos e da falta de coordenação em, por exemplo, cruzamentos, rotatórias, rampas, ultrapassagens, entre outros (BASTOS et al., 2020; ONU, 2021).

No período de 2011 a 2020, definido pela ONU como a *Década da Ação da Segurança do Trânsito*, a adoção de políticas públicas para reduzir os acidentes no Brasil conseguiu reduzir o número de mortes em 30% (ONU, 2021). No entanto, poucas iniciativas foram adotadas para reduzir a emissão de poluentes no trânsito e o congestionamento decorrente do aumento da frota dos veículos individuais, coletivos e de cargas – aspectos que impactam negativamente no conforto dos motoristas, na forma como estes conduzem os veículos e nos custos de transporte de pessoas e mercadorias (OLIVEIRA et al., 2020).

Os problemas de mobilidade representam um grande desafio para o Brasil. Por exemplo, o brasileiro gasta, em média, nas grandes metrópoles do país, 40 minutos para se deslocar de seu domicílio até o trabalho e, nas áreas não metropolitanas, essa média chega a 23 minutos (PEREIRA et al., 2021). A falta de investimentos suficientes para melhorar os congestionamentos e o alto custo das tarifas de manutenção das rodovias, evidenciam a necessidade de um planejamento urbano adequado e de políticas de trânsito mais eficientes.

Para mitigar os problemas de trânsito, inovações tecnológicas podem ser uma das ferramentas para garantir maior sustentabilidade, fluidez, conforto e segurança no trânsito (ASPLUND; LÖVHALL; VILLANI, 2017). Nesse sentido, as Redes Veiculares *Ad Hoc* (VANETs, *Vehicular Ad hoc Networks*) são uma oportunidade de construir aplicações distribuídas para Sistemas de Transporte Inteligentes (ITS, *Intelligent Transportation Systems*) (JIA et al., 2015; CUNHA et al., 2017). Nessas aplicações para ITS baseadas em VANET, os veículos se comunicam diretamente uns com os outros por troca de mensagens para coordenar suas ações cooperativamente – como por exemplo, mudança automática de faixa, cooperação automática e descentralizada de cruzamentos e rotatórias, formação veicular em pelotão, entre outros (ALVES et al., 2009; BUCHENSCHWEIT et al., 2009; JIA et al., 2015). Essas aplicações têm potencial para permitir que os veículos cooperem para melhorar o congestionamento, viagem a velocidades mais altas e evitem ações inseguras nas vias e rodovias, evitando, por exemplo, frenagens ou acelerações repentinas e acidentes (HULT et al., 2016; ROBINSON; CHAN; COELINGH, 2010) e resultando em maior fluidez e bem-estar no trânsito (GOLUBEVA; MAGARIL, 2013; JIA et al., 2015).

Para atingir seus objetivos, muitas dessas aplicações distribuídas para ITS baseadas em VANETs requerem que grupos de veículos coordenem suas ações de forma consistente, confiável e com restrições de temporais – o que é um desafio devido à alta mobilidade e comunicação sem fio entre veículos das VANETs. Nesse contexto, este trabalho propõe o Protocolo Confiável para Comunicação Veicular Baseado em Blocos Causais (VCBP, *Vehicular Causal Block Protocol*), um protocolo de comunicação em grupo para dar suporte a coordenação distribuída entre veículos em aplicações distribuídas de ITS baseadas em VANETs. O VCBP é baseado no modelo de Bloco Causal (MACÊDO, 1994) e garante confiabilidade, atomicidade, ordem causal (LAMPOR, 1978) e restrição temporal na entrega de mensagens aos membros de um grupo veicular construído em uma VANET. O VCBP foi implementado em um ambiente simulado e avaliado utilizando pelotões veiculares de transporte de carga. O seu desempenho foi comparado a outras abordagens de comunicação em grupo similares existentes na literatura, considerando cenários com diferentes números de veículos e interferência na comunicação veicular. Os resultados mostram que, nos cenários avaliados, o protocolo proposto pode oferecer mais garantias de comunicação com uma taxa de entrega de mensagens semelhante ou superior às demais abordagens avaliadas.

1.1.1 Objetivo Geral

Esta dissertação tem como principal desafio propor uma solução de comunicação confiável para dar suporte à coordenação veicular distribuída, confiável e de tempo real em redes veiculares *ad hocs*.

1.1.2 Objetivos Específicos

A partir do objetivo geral proposto, consideram-se os seguintes objetivos específicos:

- Garantias de tempo real – o suporte de comunicação deve considerar restrições temporais na entrega das mensagens na coordenação veicular em VANET;

- Suporte a entrega confiável – a solução de comunicação deve garantir que os membros de um grupo veicular entreguem as mensagens de forma atômica e em ordem idêntica, evitando inconsistência na tomada de decisão entre os veículos;
- Suporte a tomada de decisão consistente – o suporte de comunicação devem garantir que os membros do grupo veicular entreguem as mensagens respeitando causalidade, permitindo que as ações de coordenação respeitem potenciais relações de causa-e-efeito entre si.
- Reuso de abordagens da literatura – avaliar a proposição de soluções para coordenação veicular em VANET, observando o legado de soluções existentes;
- Verificação de efetividade e eficiência da solução – apresentar uma avaliação do desempenho e da confiabilidade da solução proposta quando comparada a outras abordagens existentes na literatura, considerando os requisitos da coordenação veicular em VANETs.

1.2 PUBLICAÇÕES

PEREIRA, Marcela Alves; SÁ, Alirio Santos de. **VCBP: A reliable and real-time group communication protocol for vehicular ad hoc networks**. In: *Proceedings of the 11th Latin-American Symposium on Dependable Computing*. New York, NY, USA: Association for Computing Machinery, 2023. (LADC'2022), p. 149–154. ISBN 9781450397377. Disponível em <https://doi.org/10.1145/3569902.3570191>.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está organizada conforme descrito a seguir. O Capítulo 2 e o Capítulo 3 realizam, respectivamente, uma revisão da literatura sobre VANETs e sobre protocolos de comunicação em grupo para redes não confiáveis. O Capítulo 4 descreve o *VCBP*, o protocolo de comunicação em grupo para aplicações distribuídas de coordenação veicular baseadas em redes veiculares *ad hoc* proposto neste trabalho. O Capítulo 5 apresenta a avaliação de desempenho do *VCBP*, discutindo os resultados obtidos quando comparado com outros protocolos de comunicação em grupo existentes na literatura. Por fim, o Capítulo 6 faz algumas considerações finais e descreve propostas de trabalhos futuros.

Este capítulo apresenta os conceitos básicos das redes veiculares, seus padrões e aplicações. Além disso, são apresentados também os conceitos básicos relacionados a Sistemas de Transporte Inteligentes e a sua relação com as redes veiculares.

REDES VEICULARES: CONCEITOS BÁSICOS, PADRÕES E APLICAÇÕES

2.1 INTRODUÇÃO

Uma rede de computadores pode ser definida como um conjunto de computadores interconectados por um meio de comunicação com o objetivo de comunicar e compartilhar recursos entre si (TANENBAUM; WETHERALL, 2011). Em relação ao tipo de meio de comunicação, essas redes podem ser classificadas em redes com fio e redes sem fio.

As redes com fio utilizam cabos físicos para conectar os dispositivos, enquanto as redes sem fio utilizam ondas de rádio ou sonoras para transmitir os dados. As redes com fio têm a vantagem de serem mais estáveis e seguras, mas requerem instalação de cabos, o que aumenta o custo e limita a mobilidade. As redes sem fio, por outro lado, permitem maior mobilidade e flexibilidade, contudo são mais suscetíveis a interferências nas transmissões sem fio, desconexões entre os nós da rede e perda de mensagens nos canais de comunicação (KUROSE; ROSS, 2013). Por isso, as redes sem fio são classificadas como não confiáveis e, uma vez que as transmissões podem ser interceptadas no meio de comunicação, também são menos seguras do que as redes cabeadas.

Dentre as redes sem fio, podemos distinguir entre redes com infraestrutura e sem infraestrutura. As redes com infraestrutura contam com pontos de acesso (AP, *Access Point*) que servem como pontos de conexão centralizados para os dispositivos da rede. As redes sem infraestrutura, também conhecidas como redes *ad hoc*, não contam com pontos de acesso e os dispositivos se conectam diretamente uns aos outros para compartilhar informações (ANWER; GUY, 2014).

Um exemplo de rede *ad hoc* são as redes móveis *ad hoc* (MANET, *Mobile Ad hoc NETWORK*) cujos dispositivos móveis se conectam e desconectam continuamente (RUBINSTEIN et al., 2006). Essas redes são comumente usadas em cenários militares ou de emergência, em que a infraestrutura da rede pode estar indisponível ou inacessível. Uma

subcategoria das MANETs são as redes veiculares *ad hoc* (VANETs, *Veicular Ad hoc NETWORKS*) (ALVES et al., 2009). Estas são compostas por um conjunto de dispositivos móveis (veículos), cada um atuando como um nó que se conecta automaticamente uns com os outros, de forma arbitrária, e opera sem uma administração de uma rede hierárquica estrita (QURESHI; ABDULLAH, 2013). As VANETs são altamente dinâmicas e podem ser usadas, por exemplo, para melhorar a segurança e eficiência no tráfego rodoviário, fornecendo informações sobre as condições de trânsito e possíveis perigos na estrada.

Nas demais seções deste Capítulo apresentam em mais detalhes os aspectos básicos das VANETs e de suas aplicações.

2.2 REDES VEICULARES AD HOC

Conforme introduzido na Seção 2.1, as Redes Veiculares *Ad Hocs* constituem um tipo específico de redes móveis, cujos nós são veículos capazes de enviar, receber e encaminhar mensagens diretamente entre si (ALVES et al., 2009). Essas redes herdam algumas características específicas das MANETs convencionais, tais como o dinamismo de sua topologia de comunicação e eventuais desconexões nos canais de comunicação causadas pela mobilidade dos veículos (ALVES et al., 2009).

O padrão de mobilidade das redes veiculares *ad hoc* é uma outra especificidade que está relacionada às velocidades dos nós móveis. Ao contrário das MANETs, os veículos podem se deslocar a velocidades muito altas, possivelmente em direções contrárias, o que diminui a janela de tempo durante a qual os veículos podem se comunicar diretamente. Outra característica específica dessas redes móveis é a alta autonomia de energia, uma vez que as baterias são recarregadas constantemente durante o movimento do veículo (BOUKERCHE et al., 2008). Além disso, cada vez mais veículos dispõem de recursos tecnológicos, como o receptores GPS e sensores.

Para contextualizar o cenário das redes veiculares, que é objeto de estudo deste trabalho, a próxima Seção 2.2.1 abordará a arquitetura das redes veiculares, colocando em perspectiva as VANETs. Os padrões e os protocolos utilizados na comunicação entre os veículos são discutidos, respectivamente, na Seção 2.2.2 e na Seção 2.2.3. Por fim, na Seção 2.3 são apresentadas diferentes classes de aplicações que utilizam redes veiculares.

2.2.1 Arquitetura

A arquitetura das redes veiculares define a forma como os nós se organizam e se comunicam. Atualmente, existem três arquiteturas principais: *ad hoc* puro, infraestruturada ou híbrida (ALVES et al., 2009). A Figura 2.1 apresenta um cenário com as diversas arquiteturas das redes veiculares, ilustrando a comunicação entre um veículo e um equipamento da via (modo infraestruturado), entre veículos (*ad hoc*) e entre veículos e os equipamentos da via (modo híbrido) (ALVES et al., 2009).

Na arquitetura *ad hoc*, os veículos comunicam-se, uns com os outros, sem qualquer suporte externo ou elemento centralizador. Nestas, os veículos funcionam como roteadores e encaminham tráfego através de múltiplos saltos. Embora essa seja a configuração mais simples, por não exigir nenhum tipo de infraestrutura, ela tem como principal des-

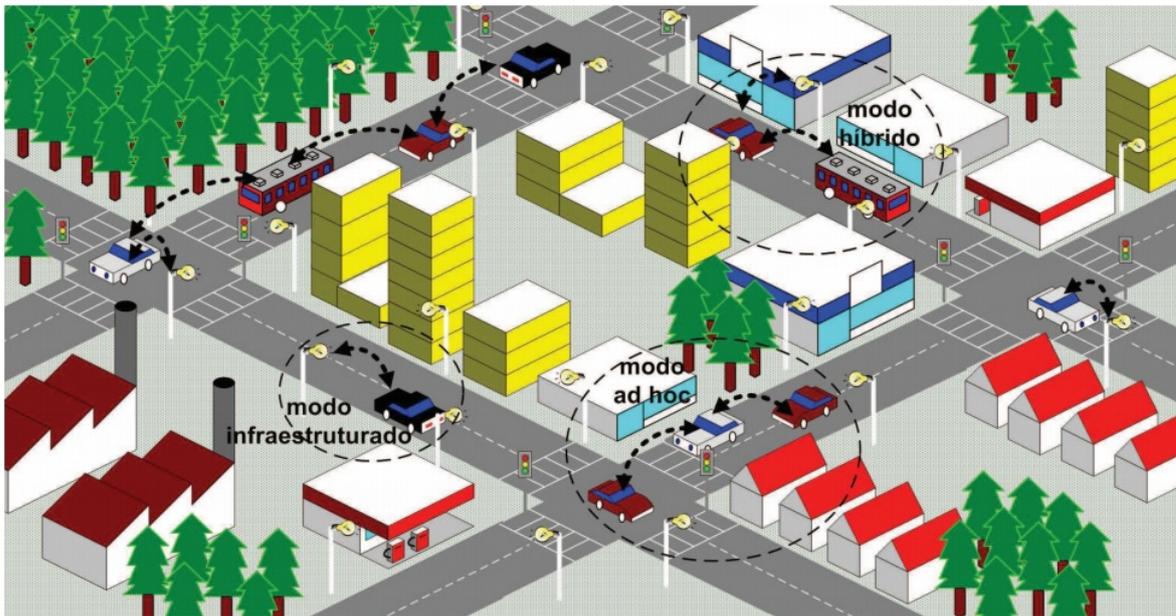


Figura 2.1: Cenário de uma rede Veicular (ALVES et al., 2009)

vantagem a conectividade da rede que depende da densidade e do padrão de mobilidade dos veículos. Estas redes são denominadas de redes *V2V* (*Vehicle-to-Vehicle*) (ALVES et al., 2009).

Na arquitetura infraestruturada, também conhecida como rede *V2I* (*Vehicle-to-Infrastructure*), os nós são distribuídos ao longo das ruas ou estradas. Esses nós funcionam como pontos de acesso para os veículos. Eles centralizam todo o tráfego da rede, servindo como pontos intermediários das comunicações. A vantagem do modo infraestruturado é o aumento da conectividade e a possibilidade da comunicação com outras redes, como a Internet. A conectividade da rede, entretanto, só é garantida mediante um número grande de elementos fixos, o que pode elevar os custos da rede.

A arquitetura híbrida é uma solução intermediária entre as arquiteturas *ad hoc* e infraestruturada. Na arquitetura híbrida, uma infraestrutura mínima é utilizada para aumentar a conectividade da rede e prover serviços como os de interconexão com outros tipos de rede de comunicação.

2.2.2 Padrões de Comunicação

Uma das primeiras tentativas de padronização das VANETs ocorreu em 1999, nos Estados Unidos, quando a *Federal Communications Commission* (FCC) alocou 75 MHz do espectro de frequências, na faixa de 5,85 GHz a 5,925 GHz, para aplicações que utilizam comunicação dedicada de curta alcance (DSRC, *Dedicated Short Range Communications*) (ALVES et al., 2009). Nesse padrão, as frequências são divididas em sete canais de 10 MHz, como mostrado na Figura 2.2. O canal 178 é o canal de controle (CCH, *Control Channel*), que é reservado para transmitir mensagens de controle, denominados de *beacons*. Os canais 172 e 184, ilustrados nas extremidades da Figura 2.2, são reservados para usos

especiais. Os demais são canais de serviço (SCH, *Service CHannel*) (ALVES et al., 2009).

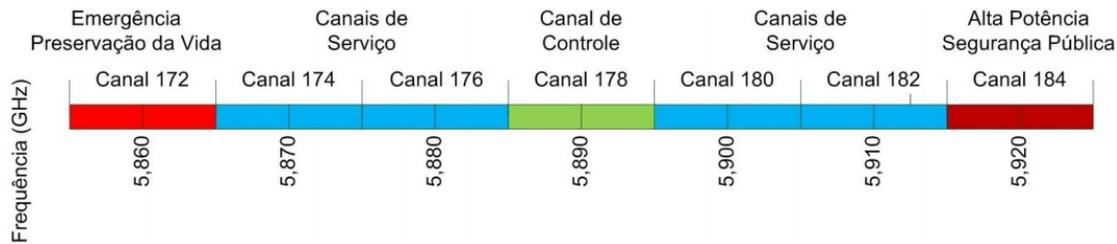


Figura 2.2: Alocação de espectro para aplicações DSRC (ALVES et al., 2009)

Desde 2004, o IEEE (*Institute of Electrical and Electronics Engineers*) vem padronizando a comunicação em redes veiculares. O padrão conhecido como IEEE 802.11p WAVE (*Wireless Access in the Vehicular Environment*) é uma modificação do padrão 802.11 com o objetivo de adicionar suporte a redes locais sem-fio em redes veiculares. Esse padrão é dividido em documentos e tem suas principais aplicações voltadas para a arquitetura V2V. O documento IEEE 802.11p define as diferenças específicas do controle de acesso ao meio em ambientes de comunicação WAVE (JIANG; DELGROSSI, 2008; ALVES et al., 2009). Em síntese, os principais documentos do padrão WAVE, família IEEE 1609, são (JIANG; DELGROSSI, 2008; ALVES et al., 2009):

- IEEE P1609.0: descreve a arquitetura WAVE, definindo o funcionamento dos padrões e os serviços necessários para que os dispositivos possam se comunicar, utilizando os múltiplos canais DSRC em um ambiente de alta mobilidade;
- IEEE P1609.1: está relacionado com o gerenciamento de recursos, descrevendo os componentes básicos da arquitetura WAVE, as mensagens de comando, os formatos para armazenamento dos dados e os tipos de dispositivos que podem ser suportados;
- IEEE P1609.2: refere-se aos serviços de segurança para as aplicações e o gerenciamento de mensagens;
- IEEE P1609.3: descreve os serviços da unidade de gerenciamento WAVE (WME, *WAVE Management Entity*) para as camadas de rede e de transporte, como o roteamento e endereçamento com suporte a troca de mensagens seguras.
- IEEE P1609.4: descreve as operações em múltiplos canais que utilizam o protocolo 802.11p (controle de acesso ao meio e camada física) para a arquitetura WAVE.

De modo geral, os documentos do padrão WAVE são especificados para prover um conjunto padronizado de interfaces para que diferentes fabricantes de automóveis possam prover comunicações V2V e V2I, a partir das unidades de bordo dos veículos (OBUs, *On-Board Units*) e das unidades de acostamento da via (RSUs, *Road Side Unit*). Uma OBU é um componente embarcado nos veículos que funciona enquanto estes transitam. Essas unidades realizam diversas funções, tais como o roteamento entre os nós da rede, o controle de congestionamento na rede e a transferência de mensagens confiáveis. As

entidades RSUs são componentes instalados em postes de luz, semáforos, em margens das vias, entre outros.

2.2.3 Protocolos de Comunicação

A arquitetura WAVE possui duas pilhas de protocolos (Figura 2.3): uma padrão da Internet, baseada no IPv6; e outra baseada no protocolo WSMP (*Wave Short Message Protocol*), projetado para a comunicação em ambientes veiculares. A razão para ter duas pilhas de protocolo é a necessidade de suportar comunicações de alta prioridade e sensíveis ao tempo – necessárias às aplicações de segurança do ambiente veicular –, como também aplicações não tão exigentes, como as transações que utilizam as transmissões dos protocolos, como TCP/UDP.

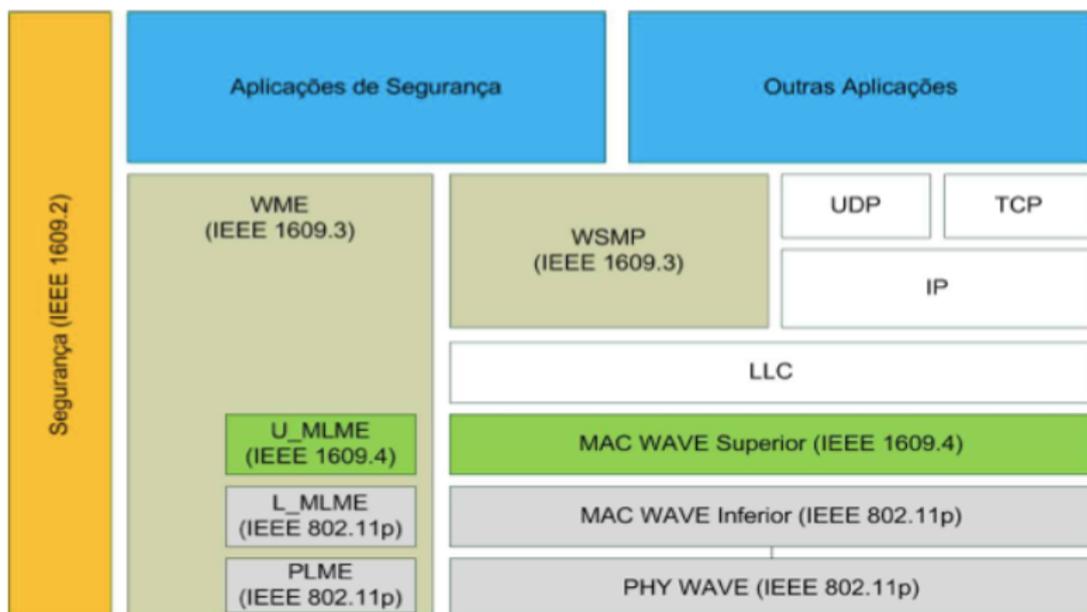


Figura 2.3: Pilha de protocolos WAVE (ALVES et al., 2009)

As mensagens do WSMP podem ser enviadas em qualquer dos canais DSRC, enquanto *datagramas* IP só podem ser enviados nos canais de serviço (SCHs). Além desses tipos de tráfego, quadros de gerenciamento são enviados no canal de controle (CCH), sendo seus formatos definidos no padrão IEEE 1609.4. O protocolo WSMP permite que as aplicações controlem diretamente características da camada física, como o canal e a potência de transmissão utilizados para enviar mensagens. A aplicação emissora também fornece o endereço MAC do dispositivo de destino, podendo também usar um endereço de difusão.

O canal de comunicação na arquitetura WAVE utiliza o canal de controle (CCH) e múltiplos canais de serviço (SCH). O canal CCH pode transmitir periodicamente mensagens curtas de status (*beacons*) que são usados, dentre outras coisas, para anunciar periodicamente a presença de veículos vizinhos. De modo geral, cada veículo pode inserir nessas mensagens informações sobre o seu status, posição, velocidade e direção. Esses dados são úteis para aplicações cooperativas, como prevenção de colisão, assistência ao

motorista e controle de cruzeiro (CAMPOLO et al., 2010).

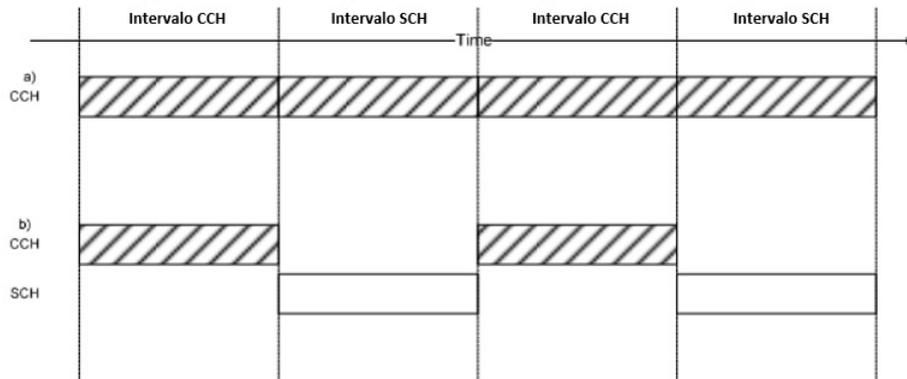


Figura 2.4: Intervalo de sincronização dos canais CCH e SCH. Adaptado de Chen, Jiang e Delgrossi (2009))

Alguns dispositivos WAVE possuem apenas uma interface de rádio, podendo comunicar em apenas um canal por vez. Assim, é necessário um mecanismo de sincronização para garantir a interoperabilidade com dispositivos com vários rádios e que podem operar simultaneamente em vários canais. O IEEE 1609.4 descreve o conceito de intervalos de canais alternados, como ilustrado na Figura 2.4. Essa figura apresenta quatro intervalos, correspondentes a dois canais de controle e dois canais de serviço, em dois cenários distintos. No primeiro cenário (a), o canal de controle (CCH) é utilizado continuamente, não existindo chaveamento com o canal de serviço (SCH). No segundo cenário (b), há uma alternância no uso desses canais de comunicação: na primeira janela de tempo de 50ms , o canal de controle é utilizado, em seguida o canal de serviço é utilizado, e assim sucessivamente. O conceito geral é que cada intervalo de sincronização tenha uma duração de 100ms . Quando o canal CCH é utilizado exclusivamente, ele ocupa todo o período de sincronização para enviar e receber mensagens. No entanto, quando há a necessidade de utilizar os canais SCH e CCH, é necessário dividir o intervalo de sincronização, reservando 50ms para o CCH e 50ms para o SCH. São dez intervalos de sincronização por segundo. O início de um intervalo CCH é sincronizado entre os veículos usando o sistema de navegação global por satélite (e.g. GPS) (CHEN; JIANG; DELGROSSI, 2009; CAMPOLO et al., 2010).

Para permitir o controle de acesso ao meio de comunicação sem fio em um mesmo canal, o protocolo IEEE 802.11p WAVE utiliza o Controle de Acesso Múltiplo com Prevenção de Colisão (CSMA/CA, *Carrier Sense Multiple Access with Collision Avoidance* (MA; CHEN, 2008)). Assim, cada veículo que deseja transmitir verifica o canal de comunicação. Se o meio estiver livre, o veículo acessará o canal. Caso contrário, se o canal estiver ocupado, o veículo irá aguardar um intervalo de tempo aleatório (intervalo de *backoff* (COMMITTEE et al., 2007)), antes de voltar a verificar o canal.

2.3 APLICAÇÕES BASEADAS EM REDES VEICULARES

As aplicações de redes veiculares podem ser divididas em três classes: segurança no trânsito, entretenimento e assistência ao motorista (ALVES et al., 2009; BUCHENS-CHEIT et al., 2009).

As aplicações de segurança possuem caráter preventivo e emergencial, onde o principal desafio é divulgar rapidamente as informações para que o condutor tenha tempo para reagir. Nessa classe de aplicações, destacam-se a divulgação de informações sobre acidentes, sobre ocorrências no trânsito e sobre condições adversas de ruas e estradas. Para exemplificar a importância das aplicações de segurança, a Figura 2.5 detalha três cenários. Inicialmente, no cenário da Figura 2.5 (a), os veículos 2, 3, 1 e 4 trafegam sem comunicação. Conseqüentemente, quando um destes veículos deseja realizar uma ultrapassagem, é necessário verificar se tem veículos se deslocando na faixa pretendida. Quando não há visibilidade – e nem leitura de sensores de mapeamento de vizinhança ou estratégias e comunicação, através da troca de mensagens – da faixa na qual o veículo está se movendo, pode acontecer uma colisão, conforme ilustrado no cenário da Figura 2.5 (b). Nesse cenário, o veículo 1 não percebeu o veículo 2 se aproximando e decidiu fazer uma mudança de faixa, provocando uma colisão entre veículos. Por fim, o cenário (c) ilustra os veículos equipados com ferramentas de comunicação. Assim, o veículo 1, antes de efetuar a ultrapassagem, envia um aviso de mudança de faixa para os nós vizinhos, porém, recebe uma mensagem do veículo 2 informando que está, no momento, utilizando a faixa pretendida. Somente após a ultrapassagem do veículo 2, o veículo 1 realiza a mudança de faixa para ultrapassar o veículo número 4. Desta forma, é possível perceber como os protocolos de comunicação são necessários na implementação de aplicações veiculares inteligentes e seguras.

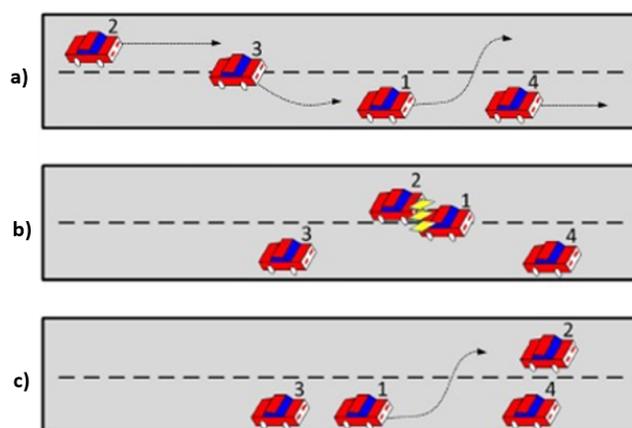
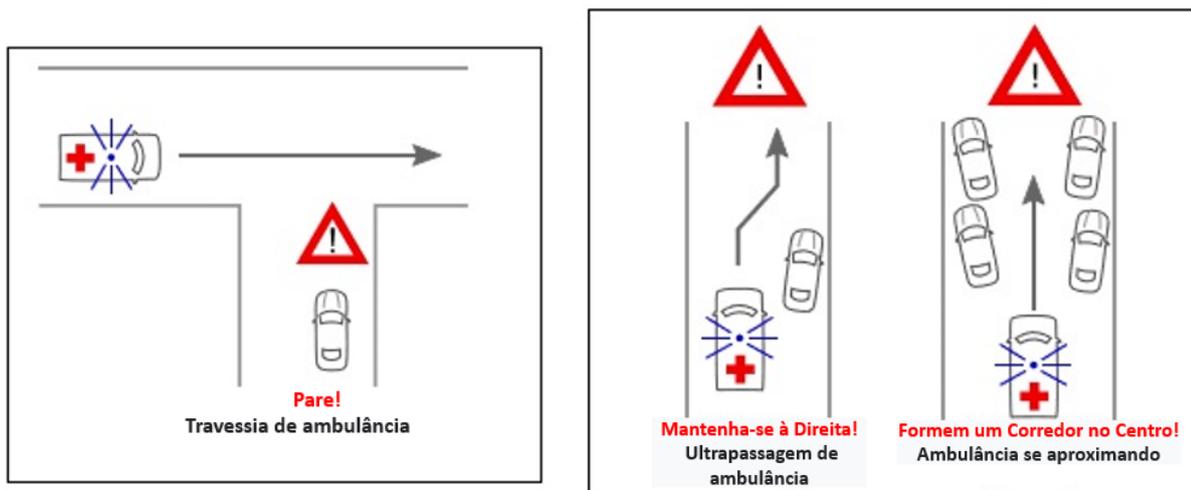


Figura 2.5: Colisões laterais na rodovia. Adaptado de Hartenstein e Laberteaux (2009)

As aplicações de entretenimento possibilitam o acesso a conteúdo de mídia, como música, filmes e programas de TV, diretamente para os dispositivos dos passageiros. Os dados podem ser transmitidos em tempo real por meio de uma rede veicular *ad hoc*, permitindo que os passageiros assistam ao conteúdo enquanto estão em movimento. O SPAWN (NANDAN et al., 2005) é uma das propostas de compartilhamento de conteúdo

que pode ser utilizada, também, para distribuir áudio e vídeo. Esse protocolo é adaptado para as redes sem fio, pois utiliza mecanismos para a descoberta de novos pares, para a seleção de pares e para o encaminhamento dos pedaços dos arquivos. O CodeTorrent (LEE et al., 2007) é outro exemplo de protocolo par-a-par (*peer-to-peer*) de compartilhamento de conteúdo em redes veiculares, que utiliza apenas a comunicação entre vizinhos próximos. O V3, uma proposta de difusão de vídeo (GUO; AMMAR; ZEGURA, 2005), realiza comunicações entre veículos sem depender de infraestrutura de comunicação na rodovia.

As aplicações de assistência ao motorista tem a finalidade de auxiliar a condução do veículo a partir da disponibilização de informações úteis. Essas informações são adquiridas a partir de serviços que podem ser oferecidos ao condutor em momentos oportunos ou podem ser de fácil acesso através de procedimentos de busca. Dentre alguns exemplos pode-se citar: aviso de estacionamentos, disseminação de informações de vias, controle de tráfego, auxílio a cruzamentos, condução conjunta de veículos, localização em mapas, aumento da visibilidade e automação da condução veicular. O TrafficView (NADEEM et al., 2004) é um exemplo de aplicação, na qual cada veículo mantém informações próprias sobre posição e velocidade, além das outras informações recebidas de outros nós da rede.



(a) Disseminação de mensagens de segurança (b) Solicitação de passagem para veículos emergenciais

Figura 2.6: Redes veiculares ad hocs usadas em sistemas de transporte de emergência. Adaptado de Buchenscheit et al. (2009)

As estratégias de disseminação de mensagens na rede V2V também estão sendo utilizadas para auxiliar os sistemas de transporte de emergência, no qual um nó emissor envia avisos sobre a condição de emergência que está ocorrendo na via, facilitando a comunicação entre os diversos outros nós que estão localizados em áreas distantes. Um exemplo destas aplicações foi proposto por Buchenscheit et al. (2009) para sincronizar a comunicação das ambulâncias com os demais veículos da rodovia (ver Figura 2.6). A partir da troca de mensagens, uma ambulância pode avisar que está se aproximando de um cruzamento e solicitar que os demais veículos parem e esperem a sua passagem para

evitar acidentes (Figura 2.6(a)). Em outra possibilidade, as ambulâncias podem solicitar aos demais veículos que liberem uma faixa para agilizar a locomoção (ver Figura 2.6(b)).

2.3.1 Pelotões Veiculares

Pelotão inteligente (*platooning*) é uma abordagem de direção cooperativa envolvendo um grupo de veículos com interesses comuns, em que um veículo segue outro e mantém uma distância pequena e quase constante do veículo anterior (JIA et al., 2015). Dentre as vantagens do *platooning*, destacam-se: a redução do congestionamento das vias, redução do consumo de combustível e o aumento da segurança do trânsito.

A implementação de um pelotão inteligente deve considerar o sistema de controle e o sistema de comunicação interveicular. O sistema de controle é responsável por acionar automaticamente os veículos. Ele se subdivide em controle longitudinal, responsável por acelerar e frear, e controle lateral, que cuida da direção.

Um pelotão deve considerar que o veículo pode conter um condutor em todos os momentos, mas também pode considerar que os veículos automatizados podem utilizar estratégias de direção cooperativa para controlar e gerenciar o pelotão, incluindo inserção de novos veículos, fusão, divisão, manutenção, entre outras.

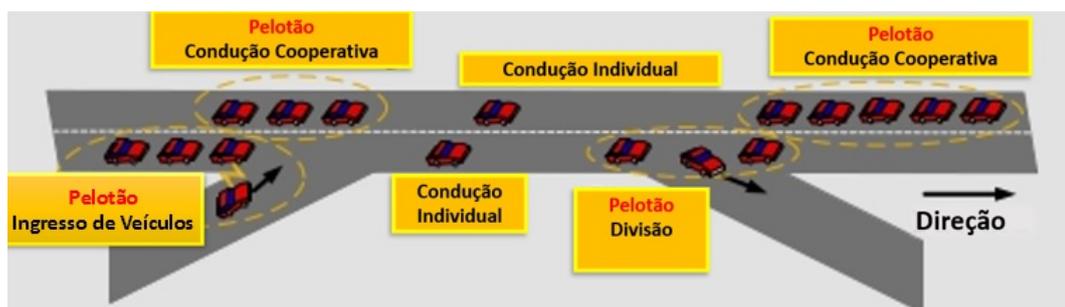


Figura 2.7: Vários pelotões no cenário de uma rodovia. Adaptado de Jia et al. (2015)

A Figura 2.7 ilustra as principais operações realizadas por um pelotão na faixa esquerda de uma via, que contém diversos carros individuais e quatro pelotões. Inicialmente, o primeiro pelotão possui 3 veículos com condução cooperativa, mas alguns veículos solicitam ingresso e ocorre a adição de membros individuais (*Join*) e também de um grupo (*merge*). No segundo pelotão, que contém 8 carros, ocorre uma divisão com a saída de membros. Assim, a manutenção de pelotões está relacionada aos processos de interação entre os veículos para garantir a estrutura funcional do pelotão (JIA et al., 2015).

Um pelotão inteligente pode ser considerado um sistema físico complexo que necessita de novas tecnologias para melhorar a segurança e a eficiência das ações. Um exemplo dessas tecnologias é o controle adaptativo de cruzeiro (ACC, *Adaptive Cruise Control*), que usa sensores para detectar a distância entre os veículos adjacentes e controlar a sua velocidade ou a distância entre esses veículos.

Além das tecnologias aplicadas individualmente em um veículo, o pelotão pode utilizar comunicação sem fio para formar dinamicamente uma rede móvel veicular *ad hoc*. A conectividade provida por essas redes permite o desenvolvimento de um ACC Cooperativo

(CACC, *Cooperative ACC*), que amplia as facilidades do ACC (OLIVEIRA et al., 2020), permitindo que os veículos compartilhem informações de mobilidade para o controle cooperativo de suas velocidades e direção.

Os pelotões podem auxiliar no gerenciamento da rodovia, melhorando o congestionamento quando a quantidade de veículos ultrapassa a capacidade da via. Essa capacidade pode ser ampliada pela redução da distância entre os veículos. Porém, é muito difícil gerir o encurtamento da distância com segurança ao considerarmos que os veículos são conduzidos por condutores humanos. Assim, a automação do deslocamento longitudinal e lateral, obtida por meio do ACC e do CACC, surge como uma alternativa para garantir a segurança e reduzir a distância entre os veículos (OLIVEIRA et al., 2020).

Apesar das vantagens enumeradas e das inúmeras aplicações dos pelotões é particularmente difícil gerir pelotões grandes em grandes cidades. Os pelotões longos podem impedir que outros veículos mudem de faixa em virtude da dificuldade de gerenciar a distância dos seus membros. A alta densidade do trânsito de algumas cidades podem impactar na gestão dos pelotões ao considerar que, por exemplo, um veículo pode cruzar uma rotatória, mas pode não haver espaço para acomodar os demais membros do grupo, bloqueando o cruzamento. Portanto, em grandes cidades, com grandes volumes de veículos, uma solução alternativa é utilizar múltiplos pelotões que podem interagir entre si.

2.4 SISTEMAS DE TRANSPORTES INTELIGENTES

Os Sistemas de Transporte Inteligentes (ITS, *Intelligent Transportation Systems*) têm como objetivo melhorar a segurança, mobilidade e eficiência dos transportes, além de diminuir os efeitos negativos do trânsito (CUNHA et al., 2017). Uma das principais aplicações do ITS é a melhoria da segurança nas estradas por meio de sistemas de aviso de colisão, monitoramento de velocidade e distância, e sistemas de análise de dados que identificam padrões e comportamentos de risco. Além disso, o ITS também pode auxiliar na gestão do tráfego, permitindo que as autoridades de transporte tomem decisões com base em informações precisas e atualizadas (DIMITRAKOPOULOS; UDEN; VARLAMIS, 2020)

A comunicação de dados é essencial para o bom funcionamento de um ITS. Ela permite a troca de informações em tempo real entre veículos, infraestruturas de estradas, sensores e aplicações de pagamento de serviços (pedágios) e entretenimento (multimídia), dentro do contexto de transporte.

2.4.1 Arquiteturas de ITS

Com a evolução das tecnologias de computação e comunicação, e o aumento da demanda por serviços de ITS com diferentes requisitos, surge a necessidade de padronizar a forma como seus dispositivos e componentes interagem entre si. Diversas arquiteturas foram propostas em diferentes países, incluindo Estados Unidos, Europa, Japão e Brasil.

A arquitetura do ITS dos Estados Unidos, conhecida como “National ITS Architecture”, definida pelo Departamento de Transporte dos Estados Unidos (U.S. Department

of Transportation), descreve a comunicação entre os elementos e subsistemas do sistema. Ela é composta por quatro classes principais (CUNHA et al., 2017):

- *Center*: define o centro de controle e gerenciamento de todo o sistema;
- *Field*: engloba toda a parte de infraestrutura do ambiente (e.g. RSUs) com sensores de monitoramento e câmeras;
- *Vehicles*: define os veículos com os sensores embarcados;
- *Travelers*: define os dispositivos usados pelas pessoas durante a viagem.

A arquitetura japonesa, proposta no projeto Smartway, permite a comunicação no modelo V2V e no modelo V2I. Ela utiliza o padrão de comunicação de curto alcance DSRC, juntamente com o padrão proposto pela Associação das Indústrias e Empresas de Rádio (ARIB, *Association of Radio Industries and Businesses*), similar ao protocolo WAVE.

A arquitetura europeia, além de ter características semelhantes às arquiteturas mencionadas anteriormente (uso do padrão DSRC e comunicação entre veículos e equipamentos rodoviários), inclui o uso do protocolo de comunicação para dispositivos móveis terrestres (CALM, *Communication Access for Land Mobiles*).

Em contraste com as arquiteturas mencionadas acima, que possuem tecnologias específicas para redes veiculares e para a comunicação entre as aplicações de ITS, a arquitetura brasileira se concentra principalmente em especificações de comunicação de ITS, seguindo as normas da ISO 14813-1:2007 e NBR ISO 14813-1:2011 (ADNORMAS, 2022). Atualmente, essa arquitetura está sendo aplicada em três principais serviços: operação e gerenciamento de tráfego, principalmente em rodovias e cidades; transporte público urbano; e pagamento eletrônico relacionado ao transporte (SANTOS; LEAL, 2015; ADNORMAS, 2022).

Este capítulo apresenta os principais conceitos relacionados à comunicação em grupo em redes não confiáveis, incluindo redes veiculares ad hoc.

COMUNICAÇÃO EM GRUPO EM REDES NÃO CONFIÁVEIS

3.1 INTRODUÇÃO

Um protocolo de comunicação em grupo é uma abstração básica que visa ocultar a complexidade e incertezas da comunicação entre um grupo de processos que cooperam juntos para realizar uma tarefa em um sistema distribuído (TANENBAUM; STEEN, 2008; COULOURIS et al., 2013). Essa abstração de comunicação pode oferecer dois serviços básicos: o serviço de filiação (*membership*), usado para gerenciar os membros do grupo; e o serviço de difusão (*multicast*), usado para difundir uma mensagem de um membro para os demais membros do grupo (TANENBAUM; STEEN, 2008; CHOCKLER; KEIDAR; VITENBERG, 2001).

Há uma extensa literatura sobre protocolos de comunicação em grupo, como por exemplo Macêdo, Ezhilchivan e Shrivastava (1993), Prakash, Raynal e Singhal (1996), Baldoni et al. (1998), Lima e Macêdo (1999), Chockler, Keidar e Vitenberg (2001), Willke e Maxemchuk (2005, 2007), Macêdo, Freitas e Sá (2013), Asplund, Lövhall e Villani (2017), Guidec, Launay e Mahéo (2021), entre outros. No entanto, nas próximas seções, este trabalho apresentará uma discussão sobre aspectos básicos de comunicação em grupo e, em seguida, se concentrará em protocolos de comunicação em grupo específicos utilizados em redes não confiáveis, destacando os protocolos para redes móveis e sem fio, incluindo implementações específicas para redes veiculares.

3.2 ASPECTOS BÁSICOS SOBRE COMUNICAÇÃO EM GRUPO

3.2.1 Classificação dos Grupos em Sistemas Distribuídos

Em relação à estrutura, os grupos de processos em sistemas distribuídos podem ser classificados em grupos simétricos e em grupos hierárquicos (TANENBAUM; STEEN, 2008).

Nos grupos simétricos, todos os processos do grupo são iguais e decidem coletivamente sobre a entrega de mensagens e sobre a gestão do grupo. Nessa estrutura, se um processo falhar, o restante do grupo pode continuar operacional. No entanto, a tomada de decisão é mais complexa, pois requer acordos entre os membros do grupo, o que implica em uma sobrecarga adicional de troca de mensagens.

Nos grupos hierárquicos, há um coordenador que toma as decisões, enquanto os outros processos simplesmente acatam essas decisões. A desvantagem dessa estrutura é que o coordenador é um ponto único de falha, pois se o coordenador falhar, o grupo se torna incapaz de tomar decisões.

Quanto à admissão de membros, os grupos podem ser classificados como estáticos ou dinâmicos. Em um grupo estático, o tamanho do grupo é conhecido no momento de sua criação, e o grupo termina quando todos os processos terminam sua execução. Em grupos dinâmicos, os processos podem ser adicionados ou removidos do grupo em qualquer momento da execução.

3.2.2 Serviço Membership

O serviço de *membership* gerencia a lista de processos ativos em um grupo, com o objetivo de prover informações atualizadas dos membros do grupo para a camada de aplicação, sempre que houver mudanças na composição do grupo (COULOURIS et al., 2013).

Em um grupo dinâmico, processos podem juntar-se voluntariamente à grupo (*join*), sair de forma voluntária (*leave*) ou, na ocorrência de falhas, deixarem o grupo de forma involuntariamente. Dessa forma, um serviço de *membership* abstrai da aplicação os eventos de mudanças no conjunto de membros do grupo, uma visão local, ou simplesmente visão de grupo, composta dos processos alcançáveis no grupo, em um determinado instante. Assim, as visões dos diversos processos do grupo devem ser atualizadas de uma maneira coerente, ou seja, os membros que as instalam devem concordar com sua composição (BIRMAN; JOSEPH, 1987). Resumidamente, um serviço de *membership* deve fornecer as seguintes facilidades (BIRMAN; JOSEPH, 1987):

1. considerando uma visão inicial, a cada adição ou exclusão de membros no grupo, o serviço de *membership* deve instalar uma nova visão do grupo;
2. a visão do grupo não pode ser alterada involuntariamente, somente a adição ou remoção de processos pode alterar o grupo;
3. todos os membros do grupo observam uma subsequência contínua da mesma sequência de visões do grupo, iniciando com a visão em que o membro foi adicionado ao grupo e terminando com aquela em que ele foi excluído;
4. o serviço de *membership* não deve atrasar indefinidamente uma mudança de visão associada a um evento de entrada ou remoção de membros no grupo;

3.2.3 Serviço de Multicast

O serviço de *multicast* considera a existência de grupos que contém processos previamente identificados. Assim, uma difusão *multicast* de uma mensagem m é uma transmissão

direcionada, exclusivamente, aos processos de um determinado grupo (HADZILACOS; TOUEG, 1994). No serviço de *multicast* são definidas duas primitivas básicas (HADZILACOS; TOUEG, 1994):

- **multicast**: Quando um processo p invoca *multicast* com uma mensagem m como parâmetro, dizemos que p faz *multicast* de m para o grupo;
- **delivery**: Quando um processo q retorna da execução de um *delivery(m)*, dizemos que q entrega m ;

Um serviço de *multicast* pode ser classificado quanto a sua confiabilidade (SCHIPER; SANDOZ, 1993):

- **multicast confiável (*reliable multicast*)**: um *multicast* de uma mensagem m para um grupo g é dito confiável se e somente se m for entregue por todos os processos corretos do grupo g ou por nenhum deles;
- **multicast confiável e uniforme (*uniform reliable multicast*)**: um *multicast* de uma mensagem m para um grupo g é dito confiável e uniforme se e somente se m for entregue por um processo (correto ou falho) do grupo g , então todos os processos corretos do grupo entregam m ;

O serviço de *multicast* também oferece garantias de ordenação, dentre as quais se destacam (MACÊDO, 1994; HADZILACOS; TOUEG, 1994):

- **Ordem FIFO (*FIFO-order*)**: exige que um processo, pertencente a um grupo G , entregue uma mensagem m se e somente se todas as mensagens previamente difundidas pelo mesmo remetente tenham sido entregues. Formalmente, considere-se que se um processo p faz um *multicast* de m_i antes de fazer um *multicast* de m_j , então nenhum processo correto do grupo deve entregar m_j antes de entregar m_i ;
- **Ordem Causal (*Causal-order*)**: exige que a entrega de mensagens respeite potenciais relações de causa-e-efeito. Essa ordenação é baseada na relação ‘*aconteceu-antes*’ (*happened-before*), definida por Lamport (1978), representada como $e_i \rightarrow e_j$, em que e_i e e_j são eventos potencialmente relacionados e e_i acontece antes de e_j e potencialmente causou e_j . Por exemplo, para respeitar causalidade na entrega de mensagens, se a ocorrência de dois envios de mensagens (no mesmo processo ou em processos distintos) têm relação de causalidade, i.e., $send(m_i) \rightarrow send(m_j)$, todos os membros do grupo devem entregar m_i antes de m_j . A relação *happened-before* pode ser observada diretamente em três situações (TANENBAUM; STEEN, 2023):
 1. Se e_i e e_j são eventos no mesmo processo e e_i ocorre antes de e_j , então $e_i \rightarrow m_j$;
 2. Se $send_i(m_k)$ é um evento de envio de mensagem m_k em um processo p_i , e $receive_j(m_k)$ é um evento de recebimento de m_k em um outro processo p_j , então $send_i(m_k) \rightarrow receive_j(m_k)$.

3. Se e_i , e_j e e_k são eventos distintos (em um mesmo processo ou em processos diferentes) e se $e_i \rightarrow e_j$ e $e_i \rightarrow e_k$, então $e_i \rightarrow e_k$ – isto é, *Happened-before* é uma relação transitiva (TANENBAUM; STEEN, 2023). Por exemplo, se $send_i(m_k)$ é um evento de envio de uma mensagem m_k em um processo p_i e $receive_j(m_k)$ é um evento de recebimento de m_k em um processo p_j e se $receive_j(m_k) \rightarrow send_j(m_l)$, então $send_i(m_k) \rightarrow send_j(m_l)$ e, portanto, m_k deve ser entregue antes de m_l .

- **Ordem Total (*Total-order*)**: exige que todos os processos de um grupo entreguem as mensagens na mesma ordem relativa. Formalmente, considera-se que se dois processos corretos p e q , pertencentes a um grupo g , entregam as mensagens m_i e m_j , então p entrega m_i antes de m_j se e somente se q entrega m_i antes de m_j .

As propriedades enumeradas acima não impõem nenhuma restrição em relação ao prazo para a entrega das mensagens. Para implementar a garantia de prazo na entrega em um grupo, as mensagens difundidas devem ser entregues dentro de limites de tempo definidos pela aplicação. Portanto, se uma mensagem é difundida no instante de tempo t , essa mensagem deve ser entregue até o instante do tempo $t + \Delta$, em que Δ é um prazo de entrega definido pela aplicação (HADZILACOS; TOUEG, 1994).

3.3 COMUNICAÇÃO EM GRUPO EM MÓVEIS SEM FIO

As redes não confiáveis possuem canais de comunicação sujeitos a perdas de mensagens e desconexões frequentes que estão relacionados a interferência do meio, colisões dos canais e a perda de conectividade. As veiculares são exemplos de redes não confiáveis que apresentam maior instabilidade nos canais de comunicação, quanto comparada a outras redes móveis, em virtude da topologia dinâmica e da alta mobilidade dos nós.

Nas redes veiculares e em outros tipos de redes móveis ou sem fio, é necessário adotar estratégias de monitoramento dos canais de comunicação para garantir os requisitos de entrega das mensagens do protocolo de comunicação em grupo. Portanto, os processos devem armazenar as mensagens recebidas e enviadas localmente com o objetivo de avaliar se uma mensagem difundida por um processo p_i pode ser entregue a aplicação atendendo as restrições de entrega do protocolo de comunicação em grupo como, por exemplo, atendimento de restrições de ordenação ou uniformidade na entrega das mensagens. foi recebida por todos os processos pertencentes ao grupo.

Nas próximas seções, diferentes protocolos de comunicação em grupo para redes móveis sem fio são discutidos, destacando protocolos com garantias de entrega causal em redes móveis sem fio, protocolos de comunicação em grupo para redes veiculares *ad hoc* e protocolos de comunicação em grupo para redes veiculares infraestruturada.

3.3.1 Δ -CBP: Protocolo para Redes Móveis Sem Fio

O Protocolo Δ -Causal Broadcast (Δ -CBP) foi utilizado por Guidec, Launay e Mahéo (2021) na construção de uma aplicação para comunicação em redes oportunistas. De modo geral, esse protocolo instancia e herda as facilidades de ordenação causal, proposto Prakash, Raynal e Singhal (1996), e de temporalidade, proposto por Baldoni et al. (1998).

Inicialmente, o Δ -CBP especifica que cada mensagem m , a ser difundida, deve conter o seu instante de envio (st , *send time*) e uma lista dos instantes de envio das mensagens entregues anteriormente à camada de aplicação, permitindo a processo receptor observar as relações de causalidade entre as mensagens recebidas e entregues. Essa lista é definida como barreira causal (CB , Causal Barrier), pois uma mensagem só será entregue se a relação de causalidade observada a partir dessa lista for atendida. Além da CB , o Δ -CBP utiliza dois *buffers*: um para armazenar as mensagens que necessitam respeitar os critérios de ordenação, definido como lista de pendência (*pending*); e, outro para armazenar a lista das mensagens entregues (*co-delivered*).

Quando uma mensagem é recebida no Δ -CBP, ocorre uma avaliação das dependências das mensagens relacionadas a m . Assim, os instantes de envio armazenados na barreira causal são confrontados com as mensagens presentes na lista de pendência. Se os instantes de tempo, demarcados na barreira causal, apontarem para mensagens entregues em um passo anterior, devem ser imediatamente removidos da mensagem e, conseqüentemente, a sua barreira causal vai sendo reduzida.

As mensagens que não possuem barreira causal, podem ser entregues à camada de aplicação e inseridas na respectiva lista das mensagens processadas ou entregues. No entanto, quando há algum st , presente na CB , relacionado a uma mensagem que ainda não foi entregue, essa mensagem deve ser inserida na lista de pendências – a qual é verificada periodicamente com o intuito de avaliar se as barreiras foram completamente eliminadas, em virtude do recebimento das mensagens anteriores que estavam atrasadas. Dessarte, uma mensagem recebida só será processada caso o seu tempo de vida e o tempo de vida dos seus predecessores não tenha sido expirado.

Finalmente, no Δ -CBP os tempos de vida das mensagens do *buffer* são avaliados constantemente. Assim, as mensagens expiradas são removidas. Apesar de considerar requisitos de temporalidade e ordenação causal, esse protocolo não possui estratégias para garantir que o mesmo conjunto de mensagens sejam entregues a todos os membros do grupo. Adicionalmente, ele também não possui mecanismos para recuperar mensagens perdidas nos canais de comunicação.

3.3.2 Protocolos para Redes Veicular Ad Hocs

3.3.2.1 M -RBP. O Protocolo Difusão Móvel e Confiável (M -RBP, *Mobile Reliable Broadcast Protocol*) foi inicialmente projetado para redes móveis, porém, vem sendo estudado e aplicado no contexto das redes veiculares em virtude das garantias de acordo e integridade (WILLKE; MAXEMCHUK, 2005).

A estrutura básica do M -RBP consiste em grupo simétrico que compõe uma rede anel (*token ring*), que utiliza a estratégia TDMA, do inglês *Time Division Multiple Access* (KUROSE; ROSS, 2013), e uma abordagem de passagem circular de *token* para organizar o acesso ao meio de comunicação durante a difusão das mensagens entre os membros do grupo (WILLKE; MAXEMCHUK, 2005).

Para garantir consistência nas ações dos processos membro do grupo (e.g., coordenação de ações de frenagens em grupo de veículos autônomo ou semi-autônomo), o M -RBP utiliza uma estratégia similar àquela aplicada na replicação de bases de dados distribuídas.

Assim, considera que um processo deve realizar uma abordagem de confirmação (*commit*) em duas fases antes de realizar a entrega das mensagens (WILLKE; MAXEMCHUK, 2004, 2005).

Antes de realizar a difusão no *M-RBP*, os nós precisam realizar o procedimento de *membership* distribuído. Inicialmente cada nó deve solicitar adesão ao grupo a partir de algum membro. A solicitação de adesão é enviada a todos os membros do grupo para ser validada e para que ocorra um acordo. Quando ocorre o acordo, o novo nó passa a pertencer a Lista de Passagem de Token (TLP, *Token Passing List*) que contém os dados dos membros e os respectivos escalonamentos dos tempos de posse do *token*. Ressaltamos que a TLP é atualizada dinamicamente e difundida a todos os membros do grupo a cada novo ingresso ou saída de um nó. Esta lista calcula o tempo de posse do *token* a partir do número de nós e do período de transmissão. Para sair do grupo, um nó pode transmitir uma mensagem de origem incluindo um *drop* para que todos os membros do grupo realizem acordo sobre a remoção do membro da lista TLP. De modo geral, os membros do grupo *M-RBP* só podem efetuar o envio de mensagens no tempo escalonado, definido na TLP. Como cada nó possui um slot (janela) de tempo dedicada para a sua transmissão, portanto não há possibilidade de envio simultâneo (WILLKE; MAXEMCHUK, 2004).

No *M-RBP*, o tempo de envio de uma mensagem é subdividido em *slots* de transmissão e retransmissão que pertencem a duas fases principais: *L1* e *L2* (WILLKE; MAXEMCHUK, 2004). Na fase *L1*, o nó remetente, que possui a posse do *token*, pode efetuar uma transmissão. Para isso, é necessário enviar o número da sequência de transmissão e a mensagem. Em seguida, o remetente deve aguardar a confirmação de todos os membros do grupo, que deve ocorrer com um envio de dois *ACKs*: um para confirmar o recebimento da sequência e outro para confirmar o recebimento da mensagem. Quando um dos membros não recebe a sequência ou a mensagem, nos *slots* pré-definidos, devem solicitar a retransmissão com um *NAK*. Após a fase de processamento do envio e a devida recuperação das mensagens, os membros do grupo precisam enviar, novamente, um voto, com um *ACK*, para confirmar o processamento da fase *L1*. Na fase *L2*, o nó remetente coleta os votos e envia um *ACK* informando que deseja realizar um *commit*, ou seja, deseja entregar a mensagem transmitida para à aplicação. No entanto, os demais membros do grupo precisam sinalizar a confirmação ou a negação. Quando a maioria dos nós confirmam a operação – acordo –, o nó remetente, juntamente com os demais nós receptores, efetuam a entrega da mensagem. Ao final do processo de difusão, os nós que não recebem a maior parte dos votos confirmando o recebimento da mensagem e nem o *commit*, não efetuam a entrega.

Por fim, o *M-RBP*, apesar de possuir estratégias para recuperar as mensagens perdidas dos canais, não garante que uma mensagem difundida será entregue a todos os participantes do grupo, em virtude do processo de votação que considera que somente os veículos que receberam a maior parte das confirmações do processos de difusões deverão fazer a entrega. Adicionalmente, o *M-RBP* possui serias desvantagens relacionadas com o escalonamento, visto que o modelo de comunicação só considera a comunicação com apenas um salto e tem um desempenho prejudicado quando o número de nós aumenta.

3.3.2.2 RNP. Protocolo de Difusão Confiável para Vizinhos (*RNP*, *Reliable Neighborcast Protocol*) estende as características do *M-RBP* e adiciona a possibilidade da comunicação em múltiplos saltos. Ele utiliza o protocolo DSRC (discutido na Seção 2.2.2) e o protocolo de mapeamento da vizinhança para proporcionar o gerenciamento eficiente de mudança de faixas entre veículos de uma rodovia, considerando as seguintes funções complexas: controle do veículo, frenagem e direção.

A partir da utilização do protocolo *RNP*, é possível mapear o conjunto veículos nas proximidades de uma vizinhança, incluindo-os nos grupos de difusão de mensagens. As vizinhanças dos veículos próximos podem se sobrepor e também conter veículos diferentes, conforme ilustrado na Figura 3.1 (MAXEMCHUK, 2009).

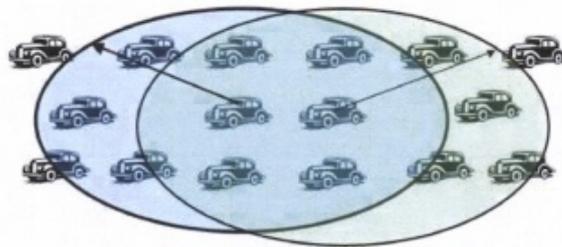


Figura 3.1: Sobreposição de grupos em uma vizinhança (MAXEMCHUK, 2009).

Para delimitar a área de alcance do mapeamento dos vizinhos, o protocolo é utilizado como parâmetro uma distância pré-definida a ser considerada na área frontal e traseira do veículo. A comunicação entre os vizinhos em pelo menos um dos seus grupos só pode ocorrer se a sobreposição entre a transmissão de grupos for maior ou igual à distância que foi pré-definida. O *RNP*, semelhante ao *M-RBP*, também utiliza uma abordagem TDMA baseada em *token* para obter atender às restrições temporais na comunicação entre os veículos, contudo também não garante o acordo na entrega das mensagens.

3.3.2.3 FSBP. O Protocolo de Difusão à Prova de Falhas (*FSBP*, *Fail Safe Broadcast Protocol*) é uma extensão do *MRP* que altera a abordagem de acordo na entrega das mensagens para oferecer garantias de entrega confiável (GU; LIN; MAXEMCHUK, 2015). Para tanto, esse protocolo utiliza 3 estados: ativo (*active*), inativo (*inactive*) e cancelado/abortado (*abort*).

Inicialmente, todos os participantes estão no estado *active* e só podem transmitir em um *slot* de tempo se estiverem nesse estado. Um participante permanece ativo até ser ordenado a abortar por um controle de processo no veículo ou até que tenha recebido uma ou mais mensagens nulas de outros transmissores. Se qualquer um desses eventos ocorrer, o participante entra no estado inativo (*inactive*) e transmite apenas mensagens nulas, sem informações da aplicação. Enquanto está inativo, ainda escuta o canal para receber mensagens, mas quando chega a sua vez, ele para de transmitir mensagens nulas

O protocolo garante que se um participante entrega uma mensagem a mesma deve ser entregue por todos os participantes. Para tanto, sempre que um participante for transmitir uma mensagem em um canal, ele deve primeiramente confirmar o recebimento de todas as mensagens recebidas anteriormente.

O *FSBP* aborta uma manobra de condução colaborativa sempre que um processo de nível superior tem um motivo para abortar a operação ou quando um par de comunicação dentro do grupo não consegue receber uma mensagem. Em aplicações veiculares, as falhas de comunicação ocorrem quando a distância entre dois veículos está além do alcance de comunicação, ou quando uma mensagem é perdida devido ao ruído do canal. Uma aplicação veicular colaborativa deve ser abortada quando os veículos saem de uma faixa de comunicação.

3.3.2.4 Vertigo. O protocolo de Vertigo, proposto por Slot (2014), utiliza o modelo de transmissão *geocast* para enviar uma mensagem a todos os veículos que estão em uma região geográfica de destino especificada, com o objetivo de prover entrega de mensagens confiável, com restrição temporal, para auxiliar na direção automatizada.

Os veículos podem receber e responder às mensagens difundidas em *geocast*, e as respostas podem ser coletadas pelo remetente. A operação *geocast* é seguida por um evento de resultado, que especifica se a mesma ocorreu com sucesso, ou seja, quando todos os veículos na área de destino receberam a mensagem e a confirmaram. Um dos benefícios desta estratégia em uma rede *ad hoc* sem fio é que não requer conhecimento prévio da presença de veículos em uma área. Para realizá-la, um veículo transmite uma mensagem contendo uma área de entrega pela rede *ad hoc* sem fio. Os veículos, ao recebem a mensagem, devem verificar se estão dentro da área de entrega com base nas informações da sua última posição e, em caso afirmativo, entregam a mensagem à aplicação e, potencialmente, encaminham a mensagem para outros veículos na área.

Para realizar um *geocast*, a aplicação precisa especificar o tempo necessário para o sistema de comunicação difundir uma mensagem e para os processos efetuarem o acordo e a entrega da mesma. Assim, uma difusão é bem-sucedida se o evento com o resultado da mesma for conhecido por todos os veículos que podem estar na área de destino, dentro do prazo especificado para realizar a entrega da mensagem.

Finalmente, apesar de garantir entrega confiável, o protocolo de Vertigo não garante os requisitos de ordenação causal e total.

3.3.3 Protocolos para Redes Veiculares Infraestruturadas

O Protocolo de Difusão Indireta Escalonável Baseado em Posição para Redes Veiculares (*PDIVanets*) foi proposto por Evropeytsev et al. (2019) com o objetivo de utilizar os veículos para transportar as mensagens dos Sistemas de Transporte Inteligente (ITS).

O *PDIVanets* possui o requisito de entrega causal para ordenar as mensagens antes de entregá-la à aplicação. Esse requisito é implementado nas RSUs, que são responsáveis por receber as mensagens provenientes dos veículos e atribuir identificadores e marcadores de direção e propagação. Dessa forma, as RSUs utilizam os veículos para transportar as mensagens. Eles armazenam as mensagens em um *buffer* e as encaminham para outras

RSUs.

De modo geral, para cada mensagem recebida de uma OBU, a RSU deve verificar se a mesma já foi entregue por outro veículo e, em caso afirmativo, a mensagem atual será descartada. Se a mensagem não foi recebida anteriormente, a condição de entrega causal é validada e, se for satisfeita, a mensagem será entregue. Quando as condições de entrega não são satisfeitas, a entrega é adiada por um intervalo de tempo fixo.

Por fim, o *PDIVanets* utiliza a infraestrutura da via para difundir as mensagens nas redes veiculares. Contudo, a confiabilidade fica comprometida, uma vez que o protocolo não possui uma rotina consistente de recuperação das mensagens perdidas nos canais, e não garante ordem idêntica.

Este capítulo apresenta o protocolo de comunicação em grupo para aplicações distribuídas de coordenação veicular baseadas em redes veiculares ad hoc proposto neste trabalho.

A PROPOSTA DE PROTOCOLO DE COMUNICAÇÃO EM GRUPO PARA REDES VEICULARES AD HOC

4.1 INTRODUÇÃO

Em aplicações distribuídas de ITS, a coordenação veicular demanda que os veículos se organizem em um grupo e concordem com as ações relacionadas às decisões de direção, buscando atribuir maior segurança, bem-estar e eficiência no trânsito. Para o enfrentamento dos desafios relacionados ao suporte de comunicação para coordenação veicular em redes veiculares *ad hoc*, este trabalho propõe um protocolo confiável de comunicação em grupo para VANET, denominado *VCBP* (*Vehicular Causal Block Protocol*).

A estrutura básica do *VCBP* é baseada na abordagem de Blocos Causais de Macêdo (1994), possibilitando, dentre outras coisas, que os veículos tomem decisões descentralizadas a respeito da entrega das mensagens. O *VCBP* é projetado para prover garantias de tempo real e confiabilidade na troca de mensagens entre os membros de um grupo veicular. Para isso, o *VCBP* realiza a entrega das mensagens aos membros do grupo atendendo a restrições temporais e garantindo que os membros do grupo obtenham informações consistentes a respeito das decisões tomadas de forma coletiva pelos veículos. Além disso, o *VCBP* garante que os membros do grupo entreguem o mesmo conjunto de mensagens, em ordem idêntica e respeitando potenciais relações de causalidade existentes entre as mensagens trocadas entre os veículos.

Para facilitar a adoção nas aplicações de ITS baseadas em VANETs, o projeto do *VCBP* considera uma camada subjacente de comunicação que se assemelha àquela prevista no IEEE 802.11p, um padrão para comunicação *ad hoc* e sem fio em ambientes veiculares.

Como nas redes veiculares os carros se organizam em grupo e não possuem um relógio global, uma alternativa prática para evitar erros na marcação temporal dos eventos é utilizar o GPS para efetuar a sincronização dos relógios. Contudo, a latência de consulta ao GPS é alta. Por isso, muitos eventos dependentes entre si podem receber uma mesma

marcação temporal, prejudicando a ordenação dos eventos no sistema. Portanto, o VCBP adota a estratégia de relógios lógicos (LAMPOR, 1978), com mecanismos de ordenação causal.

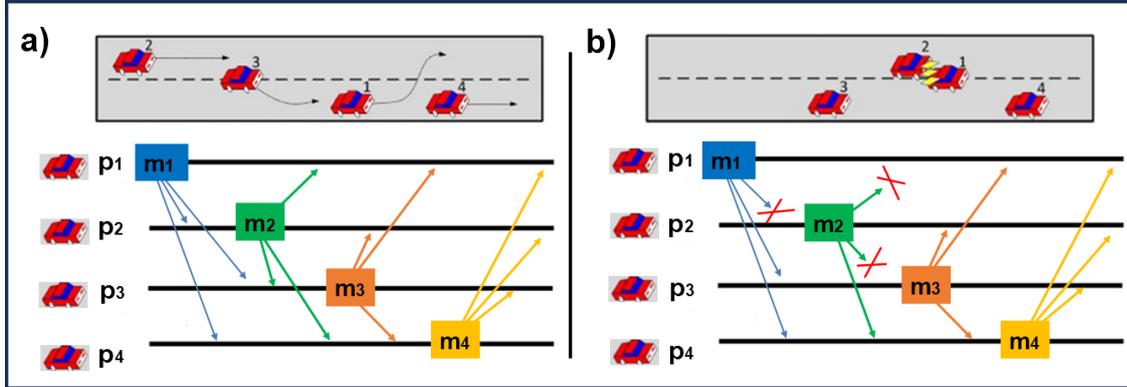


Figura 4.1: Exemplos de carros de um pelotão realizando ultrapassagem com colisão em virtude da perda de mensagens e da falta de visibilidade. Adaptado de (HARTENSTEIN; LABERTEAUX, 2009)

Para exemplificar a necessidade da ordenação idêntica e de potenciais relações de causalidade na coordenação veicular, a Figura 4.1 ilustra dois cenários. Ambos apresentam um pelotão com 4 veículos (intitulados como p_1, p_2, p_3 e p_4) que trafegam em duas faixas. Durante o período de deslocamento, os carros dos pelotões tentam fazer ultrapassagem para desviar de obstáculos da via ou de veículos não pertencentes ao grupo. Todos os veículos possuem sensores de mapeamento da vizinhança. No entanto, como esses sensores podem ser lentos e propensos a falhas (SLOT; CAHILL, 2011; SLOT, 2014), os carros necessitam utilizar a estratégia de difusão de mensagens, demarcadas com relógios físicos e lógicos, com o objetivo de informar aos demais veículos as ações que serão realizadas.

Inicialmente, no cenário da Figura 4.1(a), o carro p_1 envia uma mensagem (m_1) informando que deseja se deslocar para a faixa da direita, no entanto, os veículos p_3 e p_2 estão trafegando nesta faixa em velocidades altas. O veículo p_2 informa, a partir do envio de um mensagem (m_2), que se manterá na faixa da esquerda porque está trafegando a uma velocidade alta. O veículo p_3 sinaliza, a partir do envio de uma nova mensagem (m_3), que irá fazer a transição para a faixa da direita para permitir a ultrapassagem de p_1 . O veículo p_4 também enviou uma mensagem (m_4) informando que continuaria na faixa da direita. Considerando que todas as mensagens foram difundidas sem atrasos e sem perdas, o veículo p_1 terá o devido conhecimento dos veículos da via, a partir das mensagens difundidas e não efetuará a mudança de faixa até que o veículo p_2 passe por ele. Entretanto, quando há perdas de mensagens e atrasos na comunicação da rede veicular, os veículos podem receber as mensagens desordenadas e não ter o devido conhecimento relacionado ao tráfego dos demais veículos, conforme descrito no cenário da Figura 4.1(b) que teve perda de mensagens nos canais de disseminação de p_1 para p_2 e de p_2 para p_3 .

Na Figura 4.1(b), p_1 não recebeu a mensagem do processo p_2 e nem a mensagem do processo p_3 que deveria conter a história causal demarcando o evento de p_2 . Adicional-

mente, p_1 não conseguiu obter o devido conhecimento, a partir dos sensores, dos demais veículos. Dessa forma, p_1 recebeu as mensagens dos outros veículos e concluiu que era seguro fazer a ultrapassagem porque p_3 respondeu a sua mensagem informando que iria sair da faixa da esquerda e que era seguro para p_1 mudar de faixa – ou seja, a mensagem de p_3 não considerou o processo p_2 . No entanto, assim que p_3 saiu da faixa, p_2 tornou-se visível porque continuou acelerando e se aproximando, pois não recebeu a mensagem de p_1 e, conseqüentemente, colidiu. Se o processo p_3 tivesse recebido a mensagem de p_2 , iria gerar uma nova mensagem relacionada a ação de p_2 . Portanto, se o pelotão do cenário da Figura 4.1(b) tivesse sido implementado com o requisito de difusão confiável com ordenação idêntica, o processo p_1 só iria fazer a ultrapassagem após receber todas as mensagens relacionadas aos demais processos, sinalizando segurança na ação pretendida. Adicionalmente, se o pelotão também tivesse implementado o requisito de ordenação com potencial relação de causa e efeito, o veículo p_4 não iria permitir a ultrapassagem de p_1 porque geraria a sua mensagem contendo informações referentes aos eventos gerados por p_2 . Assim, torna-se evidente a importância dos requisitos de ordenação idêntica e de garantia das potenciais ações relacionadas entre os veículos, membros de grupo de comunicação, que executam aplicações de coordenação veicular.

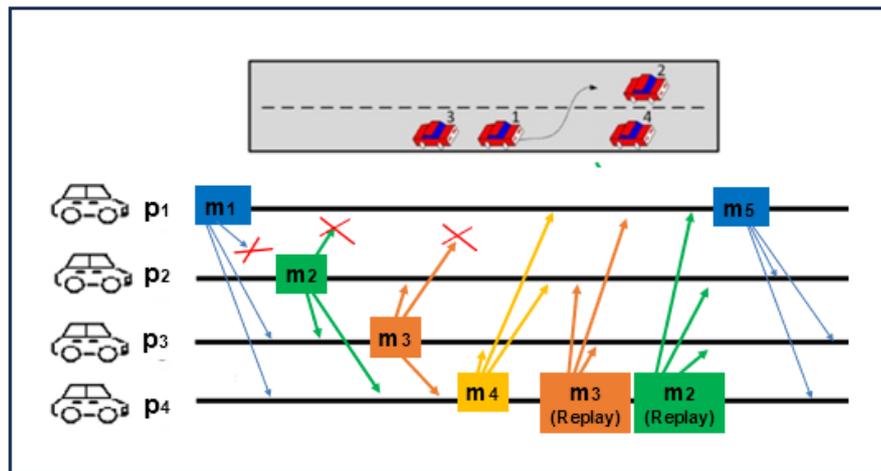


Figura 4.2: Exemplo de carros de um pelotão realizando ultrapassagem segura a partir da utilização do VCBP. Adaptado de (HARTENSTEIN; LABERTEAUX, 2009)

Para solucionar os problemas relacionados à falta de confiabilidade dos canais e ao atraso de mensagens no cenário de coordenação veicular, a Figura 4.2 exemplifica a difusão de mensagens a partir da utilização do VCBP. Nesta figura, quando o veículo p_1 enviou sua primeira mensagem informando que desejava mudar para a faixa da esquerda, os veículos p_2, p_3 , e p_4 responderam com outras difusões. Entretanto, p_1 não recebeu algumas mensagens devido as perdas que ocorreram nos canais. Como o VCBP garante entrega idêntica e causal, nenhum veículo conseguiu entregar as mensagens, pois esse passo só pode ser executado quando a difusão de todos os veículos ocorrer em sequência e com sucesso ou quando há uma expiração do tempo de vida das mensagens. No entanto, o protocolo permite que os veículos efetuem retransmissões para recuperar mensagens

perdidas. Em geral, todos os veículos foram equipados para armazenar localmente as mensagens e as difusões efetuadas pelos membros do grupo. Adicionalmente, todas as mensagens foram adaptadas para transportar metadados contendo as histórias causais referentes ao recebimento de cada veículo. A partir destas estratégias, o veículo p_4 inferiu que p_2 não tinha recebido a mensagem m_1 e que p_1 não tinha recebido as mensagens m_2 e m_3 . Após a detecção das mensagens perdidas, p_4 efetuou as devidas retransmissões que chegaram a todos os veículos. Ao receber as mensagens desordenadas, p_1 e os demais veículos as organizaram em seus *buffers* locais, ordenando-as de acordo com os critérios de ordenação idêntica e de potenciais relações de causa e efeito. Em seguida, entregaram-as à aplicação, pois todas estavam dentro do prazo de validade (*deadline*) - mensagens com prazos de validade expirados são descartadas porque podem ser incompatíveis com o cenário atual da coordenação veicular. Com o devido processamento das mensagens disseminadas, p_1 inferiu que era inseguro realizar a ultrapassagem e esperou um tempo t para p_2 realizar a ultrapassagem. Em seguida, p_1 enviou outra mensagem (m_5) informando sua segunda tentativa de ultrapassagem. Enfim, a partir deste exemplo, é possível observar a importância do *VCBP* em aplicações de coordenação veicular.

O restante deste capítulo apresenta o *VCBP* com maiores detalhes. Na Seção 4.2, o *VCBP* é posicionado em relação a outras soluções existentes no estado da arte. Em seguida, a Seção 4.3 aborda o modelo de sistema distribuído considerado na concepção do algoritmo proposto. A Seção 4.4 discute resumidamente o modelo de blocos causais, proposto por (MACÊDO, 1994), utilizado na implementação do *VCBP*. A Seção 4.5 apresenta uma visão geral das propriedades e do funcionamento do *VCBP*. Por fim, a Seção 4.7 descreve detalhadamente o algoritmo proposto.

4.2 TRABALHOS RELACIONADOS

Embora haja uma ampla variedade de protocolos de transmissão para VANETs e redes *ad hoc* móveis em geral, a maioria das abordagens fornece apenas garantias de entrega de mensagens de melhor esforço (RUIZ; BOUVRY, 2015; WILLKE; TIENTRAKOOL; MAXEMCHUK, 2009).

O *M-RBP* (WILLKE; MAXEMCHUK, 2005), o *RNP* (MAXEMCHUK, 2009) e o *FSBP* (GU; LIN; MAXEMCHUK, 2015) são protocolos de comunicação em grupo para VANETs. Esses protocolos foram especialmente projetados para suportar a coordenação veicular, garantindo restrição temporal na entrega de mensagens. Contudo, diferente do *VCBP*, esses protocolos não garantem causalidade e confiabilidade na entrega das mensagens – requisitos importantes para garantir ações consistentes em aplicações de coordenação veicular. Isto porque, em cenários nos quais há perda de mensagens nos canais de comunicação, a abordagem adotada por esses protocolos admite que os veículos entreguem um conjunto distinto de mensagens em ordem adversa da ordem causal. Outro diferencial importante é que o *M-RBP*, o *RNP* e o *FSBP* consideram o uso de TDMA para ordenar as transmissões, enquanto o *VCBP* confia na abordagem de Blocos Causais de Macêdo (1994) para permitir que os veículos tomem decisões de forma descentralizadas – isto é, sem a necessidade que um veículo assumo o papel de coordenador do grupo.

O *RCBP* (LIMA; MACÊDO, 1999) e o Δ -*CBP* (GUIDEC; LAUNAY; MAHÉO,

2021) são protocolos de comunicação que garantem ordem causal na entrega de mensagens em redes não confiáveis.

O protocolo Δ -CBP foi projetado para redes oportunistas – um subtipo de rede móvel e *ad hoc* com conexões muito transitórias entre seus nós (PELUSI; PASSARELLA; CONTI, 2006). Para garantir ordenação causal na entrega das mensagens, este protocolo usa a abordagem de barreira causal, proposta em (PRAKASH; RAYNAL; SINGHAL, 1996), em ambientes de redes móveis celular. Além disso, atende a restrições temporais na entrega das mensagens em redes *broadcast*, incorporando na solução de barreira causal um tempo de vida para as mensagens, o que é a mesma proposta apresentada por Baldoni et al. (1998) para garantia de entrega eficiente com ordem causal para aplicações multimídias em ambiente de redes móveis celular. Contudo, Δ -CBP não dá garantias de que o mesmo conjunto de mensagens sejam entregues aos membros do grupo, o que representa um problema para aplicações de coordenação veicular.

O RCBP é um protocolo de comunicação em grupo para redes de comunicação sujeitas a perda de mensagens. É um protocolo desenvolvido usando o modelo de Blocos Causais (MACÊDO, 1994) que garante ordem total e causal na entrega das mensagens. O VCBP é uma extensão do RCBP. Todavia, diferente do VCBP, o RCBP não considera restrições temporais na entrega das mensagens e nem foi projetado ou devidamente avaliado para atuar em VANETs. O VCBP também usa o modelo de Blocos Causais para garantir ordem total e causal na entrega de mensagens. Além disso, assim como o RCBP, também considera um mecanismo de recuperação de mensagens perdidas. Contudo, diferente do RCBP, sua política de retransmissão considera a vizinhança entre os veículos, evitando retransmissões desnecessárias.

A Tabela 4.1 sumariza a relação entre os protocolos avaliados neste trabalho e os respectivos requisitos que são garantidos.

Tabela 4.1: Avaliação dos Requisitos.

Protocolos	VANETs	Requisitos			
		Entrega Confiável	Ordem Idêntica	Ordem Causal	Restrição Temporal
M-RBP	X	X			X
RNP	X	X			X
FSBP	X	X	X		X
Δ -CBP				X	X
RCBP			X	X	
VCBP	X	X	X	X	X

Por fim, considerando as diferentes abordagens da literatura, o VCBP oferece um conjunto maior de garantias de entrega de mensagens, observando os requisitos da coordenação veicular em VANETs.

4.3 MODELO DE SISTEMA

O sistema é composto por um conjunto $\pi = \{p_1, p_2, \dots, p_n\}$ de processos, cada um executando em um veículo (nó móvel), dotado de facilidade de comunicação sem fio e de

sistema operacional multi-tarefa e de tempo real.

A rede subjacente de comunicação é *ad hoc* e os processos do sistema interagem a partir de troca de mensagens via canais de comunicação sem fio bidirecionais e não confiáveis, estando sujeitos a perda de mensagens, entrega de mensagens fora de ordem e atrasos de comunicação imprevisíveis. Mensagens corrompidas são descartadas pelos canais de comunicação subjacentes.

A interação entre pares de veículos é realizada a partir de pares de canais de comunicação: um dos canais é dedicado à troca de mensagens da aplicação; enquanto outro canal é dedicado à troca periódica de mensagens de controle (*beacons*) utilizadas para detecção de defeitos, gestão de visão de grupo veicular e obtenção de informações relacionadas à mobilidade dos veículos.

Cada veículo é equipado com um GPS (*Global Positioning System*) (HAIDER et al., 2019), que lhes permite obter sua localização atual e manter seu relógio físico sincronizado com os relógios físicos dos demais veículos do sistema. Os desvios de relógio local (*clock drift*) e entre relógios de veículos distintos (*clock skew*) são desprezíveis. As informações locais do GPS e do relógio físico estão acessíveis ao processo no veículo. Outros sensores embarcados no veículo (e.g., acelerômetro e IMU¹) compensam a alta latência e possíveis erros nas informações de localização do GPS.

Um processo executa em passos, representados por qualquer mudança local de estado ou por ações de recepção e de envio de uma mensagem e pelas respectivas mudanças locais de estado advindas dessas ações. Existem limites conhecidos para os tempos de processamento. Processos podem falhar, parando prematuramente de funcionar – i.e., modelo de falha por parada (*crash faulty model*²). Processos são ditos corretos, se não falham por parada. Falhas bizantinas não são consideradas.

Para cooperar na coordenação das ações dos veículos, processos do sistema se organizam em um grupo g . Por simplicidade, a solução apresentada considera um grupo único, mas pode ser estendida para trabalhar com múltiplos grupos – ver (MACÊDO; EZHILCHLVAN; SHRIVASTAVA, 1993). Além disso, um serviço de afiliação confiável (*membership*) é responsável por gerenciar o ingresso e a saída de membros em um grupo g , garantindo que cada veículo possua uma visão consistente sobre a composição do grupo. Os detalhes desse serviço de *membership* estão fora do escopo deste trabalho. Uma variedade de implementações de serviços de *membership* para protocolos de comunicação em grupo baseados na abordagem de Blocos Causais podem ser encontradas na literatura, por exemplo em (MACÊDO; EZHILCHLVAN; SHRIVASTAVA, 1993; MACÊDO; FREITAS, 2009).

Por fim, questões relacionadas à forma de condução dos veículos estão fora do escopo desse trabalho. Por isso, não há enfoque específico sobre condução autônoma, semi-autônoma, ou não automatizada, ou sobre possíveis aspectos de implementação relacionados.

¹A Unidade de Medição Inercial (*Inertial measurement unit*, IMU) é um dispositivo eletrônico que mede e relata a força específica de um corpo, a taxa angular e, às vezes, a orientação do corpo, usando uma combinação de acelerômetros, giroscópios e, às vezes, magnetômetros (AHMAD et al., 2013).

²ver Cristian (1991) para uma discussão sobre modelos e tipos de falhas em sistemas distribuídos.

4.4 VISÃO GERAL DO MODELO DE BLOCOS CAUSAIS

No modelo de Blocos Causais (MACÊDO, 1994) adotado no *VCBP*, cada processo $p_i \in g$ mantém um relógio lógico (LAMPART, 1978), denominado de contador de bloco (*block counter*) e denotado por bc_i – uma variável inteira incrementada monotonicamente e que é inicializada com zero quando o grupo g é criado.

Cada mensagem m enviada por p_i carrega um número de bloco ($m.bn$), correspondendo ao valor atual de bc_i . Cada processo p_i incrementa seu bc_i antes do envio ($send_i(m)$) ou entrega ($deliver_i(m)$) de uma mensagem m :

(R1.1) No evento de envio de m ($send_i(m)$) em p_i :

$$bc_i \leftarrow bc_i + 1;$$

$$m.bn \leftarrow bc_i;$$

(R1.2) No evento de entrega de m ($deliver_i(m)$) em p_i :

$$bc_i \leftarrow \max \{bc_i, m.bn\};$$

em que “ $x \leftarrow y$ ” representa uma atribuição do valor de y para x .

As regras **R1.1** e **R1.2** observam a potencial relação de caso-e-efeito (causalidade) entre eventos de envio e entrega. Dois eventos a e b possuem relação de causalidade, se o evento b puder ter sido causado ou influenciado por a . Portanto, o evento b “acontece antes” (“*happen-before*”) do evento a e essa relação é representada por $a \rightarrow b$. Seguindo as regras **R1.1** e **R1.2**, dado duas mensagens distintas m e m' , se $send_i(m) \rightarrow send_i(m')$, então $m.bn < m'.bn$. Além disso, essas regras garantem que quaisquer mensagens distintas com o mesmo número de bloco são concorrentes, i.e. não possuem relação de causalidade e foram difundidas (originalmente) por processos diferentes (MACÊDO, 1994). Em algumas versões de protocolos de comunicação em grupo baseados no modelo de Blocos Causais, uma mensagem transmitida por um processo do grupo pode ser repassada (retransmitida ou roteada) por outro processo do grupo para outros membros do grupo. Ainda assim o número de bloco da mensagem é definido pelo primeiro transmissor – i.e., quem originou a mensagem.

Cada processo p_i armazena localmente as mensagens concorrentes (i.e. com os mesmo número de bloco) em um vetor, denominado bloco causal (*causal block*). O conjunto de blocos causais ordenados pelo seus números de blocos permite que um processo p_i construa localmente uma matriz de blocos (*Block Matrix*), denotada por bm_i . Portanto, toda vez que p_i origina ou recebe uma mensagem m pela primeira vez, o mesmo insere m em bm_i no bloco causal correspondente, isto é $bm_i[m.bn][m.src] \leftarrow m$. A Figura 4.3 ilustra uma matriz de blocos para um grupo com seis processos – as mensagens enviadas ou recebidas são representadas em cada bloco causal com o símbolo “+”. Se a matriz de blocos ilustrada na Figura 4.3 pertencer a p_2 (i.e. bm_2), então ela indica que a última mensagem transmitida por p_2 tem número de bloco igual a 5 e as últimas mensagens recebidas de p_1 , p_3 , p_4 , p_5 e p_6 tem número de bloco 4, 3, 3, 4 e 2, respectivamente. Cabe salientar que essa representação matricial é apenas conceitual, a matriz de blocos causais pode ser implementada usando uma estrutura de dados mais eficiente em termos de custo de armazenamento, por exemplo, *hash* ou lista de mensagens.

	p_1	p_2	p_3	p_4	p_5	p_6
1	+			+		
2		+			+	+
3	+		+	+		
4	+				+	
5		+				

Figura 4.3: Exemplo de matriz de blocos com 6 processos (MACÊDO, 1994)

Nos protocolos de comunicação em grupo baseados em Blocos Causais, um processo p_i usa sua matriz de blocos local (bm_i) para decidir quando deve entregar as mensagens recebidas para a aplicação, observando os requisitos de entrega de mensagem (e.g. ordem causal, total etc.). Para garantir ordem causal, basta que as mensagens sejam entregues em ordem crescente do número de bloco. A entrega em ordem total pode ser realizada se os processos do grupo usam uma mesma regra de ordenação na entrega de mensagens concorrentes (i.e., pertencentes a um mesmo bloco causal). A depender do modelo de sistema distribuído, além do número de bloco e do identificador do remetente da mensagem, informações de controle adicionais devem ser inseridas nas mensagens (ou serem mantidas nos processos do grupo) para garantir a correção do protocolo de comunicação em grupo (MACÊDO; EZHILCHLVAN; SHRIVASTAVA, 1993; MACÊDO, 1994; LIMA; MACÊDO, 1999; MACÊDO; FREITAS, 2009; MACÊDO; FREITAS; SÁ, 2013).

Independente do modelo de sistema distribuído, antes de realizar a entrega das mensagens de um bloco causal, uma condição básica que qualquer protocolo baseado em Blocos Causais deve observar é a *Completeness of Causal Block* (MACÊDO; EZHILCHLVAN; SHRIVASTAVA, 1993). Um bloco causal de número de bloco b na matriz de blocos de p_i (i.e. $bm_i[b]$) será *completo*, quando p_i não puder enviar ou receber uma mensagem m com $m.bn = b$ (MACÊDO; EZHILCHLVAN; SHRIVASTAVA, 1993). Essa condição de completude pode variar de acordo com as características dos canais de comunicação. No caso particular do *VCBP*, construído sobre canais não confiáveis, sujeitos a perdas de mensagens e entrega de mensagens fora de ordem, um bloco causal de número de bloco b estará completo em p_i se todos os processos do grupo colaboraram com mensagem em $bm_i[b]$, isto é: $\forall p_k (\exists m \in bm_i[b][p_k] : m.bn = b \wedge m.src = p_k)$.

4.5 VISÃO GERAL DO VCBP

Para atender aos requisitos de confiabilidade das aplicações envolvendo coordenação veicular, o *VCBP* implementa um protocolo de difusão confiável (*reliable multicast*) que provê as seguintes garantias na entrega das mensagens trocadas entre os veículos membros de um grupo veicular:

(G1) **Integridade** – um veículo entrega uma mensagem m apenas uma vez e se m tiver

sendo enviada por algum outro veículo.

- (G2) **Ordem Causal** – a entrega garante consistência nas ações dos veículos acionadas a partir da troca de mensagens com potencial relação de causalidade, isto é:
- (G2.1) se um veículo v_i envia m e em seguida envia m' , então m deverá ser entregue antes de m' ;
- (G2.2) se um veículo v_i entrega uma mensagem m antes de enviar uma mensagem m' , então os demais veículos devem entregar m antes de m' ;
- (G3) **Ordem Total** – se dois veículos v_i e v_j ambos entregam duas mensagens m e m' , então v_i entrega m antes de m' se e somente se v_j entrega m antes de m' .
- (G4) **Acordo** – se um veículo entrega uma mensagem m , então todos os demais veículos entregam m .
- (G5) **Restrição Temporal** – se uma mensagem m é entregue no instante t , então m não foi difundida antes de $t' = t - \Delta$, em que Δ é um prazo de entrega definido pela aplicação;

4.5.1 Garantindo Integridade, Ordem Causal, Ordem Total e Restrição Temporal na Entrega de Mensagens

As garantias de **Integridade**, de **Ordem Causal** e de **Ordem Total** na entrega das mensagens são facilidades herdadas do *RCBP* de Lima e Macêdo (1999). Para tanto, o contador de blocos bc_i de cada processo $p_i \in g$ evolui usando as regras **R1.1** e **R1.2** apresentadas na Seção 4.4. Além disso, para assegurar ordem idêntica, cada processo correto de g entrega as mensagens de sua respectiva matriz de blocos causais em ordem crescente do número de bloco e as mensagens de um mesmo bloco são entregues em ordem crescente dos identificadores dos membros de g . Para assegurar que as mensagens de um bloco são entregues uma única vez, cada processo p_i mantém uma variável ld_i que armazena o número do último bloco entregue.

Para assegurar **Restrição Temporal** na entrega das mensagens, cada mensagem m difundida carrega o prazo ($m.d$) de validade de seu respectivo bloco causal. Na difusão ou no recebimento de uma mensagem é escalonada a expiração do bloco causal – apenas uma vez para cada bloco. No prazo de expiração de um bloco causal, caso não haja condição de entrega das mensagens do mesmo, elas são anuladas. Mensagens nulas não são entregues para a aplicação.

4.5.2 Minimizando Perdas de Mensagens nos Canais de Comunicação

Para minimizar perdas de mensagens, e.g. provocadas pelas condições dos canais de comunicação ou pela mobilidade dos veículos, o VCBP combina o mecanismo de envio periódico de *beacons* com uma estratégia de detecção e retransmissão de mensagens. *Beacons* são mensagens de controle usadas nas VANETs para que um veículo notifique aos veículos vizinhos sua presença e sua situação em termos de posição, velocidade, aceleração

etc. Uma vez que os veículos interagem usando enlaces de comunicação sem fio baseados em padrões de comunicação veicular, e.g. IEEE 802.11p (JIANG; DELGROSSI, 2008), as difusões de *beacons* usam canais de controle e difusões de mensagens de aplicação usam canais de dados, o que reduz potenciais colisões nas transmissões entre veículos. O envio periódico de *beacons* garante que todo processo correto $p_i \in g$ contribuirá em todo bloco causal, dispensando o uso de outros mecanismos adotados em abordagens de blocos causais para completude de blocos, como o mecanismo de *time-silence* (MACÊDO; EZHILCHLVAN; SHRIVASTAVA, 1993). Com isso, entradas sem mensagens na matriz de blocos sinaliza possível perda de mensagens.

Para detectar e recuperar mensagens, cada processo $p_j \in g$ armazena dois vetores:

- vetor mbv_j – contendo uma entrada para cada processo $p_k \in g$, na qual armazena o valor do maior número de bloco ($m.bn$) recebido (ou enviado) em uma mensagem m originada em p_k (i.e., $m.src = p_k$); e
- vetor $vlsbn_j$ – contendo uma entrada para cada processo $p_k \in g$, na qual armazena o valor do último número de bloco ($m.bn$) recebido em sequência (e sem descontinuidade) a partir de mensagens originada em p_k .

Para exemplificar o uso de $vlsbn_j$ no *VCBP*, suponha que $p_j \in g$ recebeu mensagens de algum $p_k \in g$, cada uma contendo, em seus respectivos campos bn , os números de bloco 1, 2, 3, 5, 6, 8, 9, então o valor de $vlsbn_j[p_k]$ é 3. Se em algum momento p_j receber a mensagem m originada de p_k com $m.bn = 4$, o valor de $vlsbn_j[p_k]$ é atualizado para 6, visto que terá recebido sem descontinuidade as mensagens com número de bloco 1 a 6. Por outro lado, se p_j tivesse recebido de p_k uma mensagem m' com $m'.bn = 7$, sem ter recebido $m.bn = 4$, então o valor de $vlsbn_j[p_k]$ continuaria 3. Quando ambas as mensagens m e m' originadas em p_k forem recebidas por p_j , o valor de $vlsbn_j[p_k]$ será atualizado para 9.

Quando algum processo $p_i \in g$ difunde uma mensagem m , ele insere as informações de seu vetor $vlsbn_i$ em m . As informações de $vlsbn$ recebidos nas mensagens de processos do grupo, em conjunto com $vlsbn_j$, permitem a p_j construir a matriz $mlsbn_j$, em que $mlsbn_j[p_k]$ contém as informações de $vlsbn_k$ que p_j recebeu da mensagem mais recente originada em p_k .

Com o seu mbv_j e conhecendo as informações de $vlsbn_i$ de p_i (i.e., $mlsbn_j[p_i]$), o processo p_j suspeita que p_i não recebeu alguma mensagem m originada em p_k (i.e. $m.src = p_k$), se $mlsbn_j[p_i][p_k] < m.bn < \min(mbv_j)$, sendo $p_i \neq p_k \neq p_j$. Essa abordagem de detecção de mensagens perdidas é similar a introduzida no *RCBP* por Lima e Macêdo (1999). Contudo, no caso do *RCBP*, os processos inserem nas mensagens informações dos últimos blocos entregues por cada processo ao invés de usar informações dos últimos blocos recebidos em sequência por cada processo, como é feito no *VCBP*.

Um processo p_j só retransmite m para p_i se o respectivo bloco causal de m não tiver expirado e p_i estiver dentro de seu raio de retransmissão (R), o qual é inferior ao raio de alcance de comunicação. Usar um raio de retransmissão inferior ao alcance da comunicação sem fio é uma estratégia para selecionar os processos retransmissores dentre os vizinhos de p_i , reduzindo a chances de colisões de transmissões. Além disso, para

evitar que múltiplos vizinhos próximos de p_i retransmitam mensagens simultaneamente, cada vizinho aguarda um intervalo de tempo equivalente a metade do período de *beacon* somado a uma pequena janela de tempo aleatória (intervalo de *backoff*). Essa abordagem especializa o mecanismo de retransmissão do *RCBP*, introduzindo na política de retransmissão de mensagens as restrições relacionadas ao prazo de validade para o bloco causal e ao raio de retransmissão entre veículos. Adicionalmente, o intervalo de *backoff* considera os identificadores dos veículos no sorteio da janela de retransmissão no intervalo de *backoff*.

4.5.3 Garantindo Acordo na Entrega das Mensagens

As informações de *vlsbn* trocadas entre os processos do grupo são suficientes para detectar mensagens perdidas. Contudo, para garantir **Acordo** na entrega das mensagens é necessário que os processos disseminem as informações de suas respectivas matrizes *mlsbn*. Para tanto, cada processo $p_j \in g$ insere as informações de sua matriz *mlsbn_j* em cada mensagem m enviada. Isso permite, a divulgação tanto das informações de seu *vlsbn_j*, como também das informações dos *vlsbn_i* recebidos de outros processos $p_i \in g$.

As informações de sua matriz *mlsbn_j* em conjunto com as informações de cada matriz *mlsbn_i*, recebidas nas mensagens, permitem a p_j construir a matriz *glsbn_j*, em que *glsbn_j*[p_k] contém as informações de *mlsbn_k* que p_j recebeu da mensagem mais recente originada em p_k . O uso de *glsbn_j*, *mlsbn_j* e *vlsbn_j* é redundante, uma vez que *glsbn_j*[p_j] contém *mlsbn_j* e *glsbn_j*[p_j][p_j] contém *vlsbn_j*. Contudo, nesta discussão, manteremos as três variáveis para facilitar o entendimento e descrição do algoritmo.

Com a sua *mlsbn_j* e conhecendo as *mlsbn_i* de p_i (i.e. *glsbn_j*[p_i]), o processo p_j é capaz de detectar condições de estabilidade e de super-estabilidade para a entrega das mensagens de cada bloco causal:

- **Estabilidade** – um bloco causal bn é **estável** em p_j , se p_j sabe que tal bloco pertence a uma sequência de blocos completos em todos os membros de g , i.e., $(\forall p_i \in g)(\forall p_k \in g)(bn \leq mlsbn_j[p_i][p_k])$, o que é equivalente a $(\forall p_i \in g)(\forall p_k \in g)(bn \leq glsbn_j[p_j][p_i][p_k])$; e
- **Super-Estabilidade** – um bloco causal bn é **super-estável** em p_j , se p_j sabe que tal bloco é **estável** em todo o processo p_k do grupo – i.e. $(\forall p_l \in g)(\forall p_i \in g)(\forall p_k \in g)(bn \leq glsbn_j[p_l][p_i][p_k])$.

Os membros de g realizam a entrega de mensagens apenas de blocos **super-estáveis**, garantindo a propriedade de **Acordo** necessária para a entrega confiável (*reliable delivery*) de mensagens.

4.6 DETALHAMENTO DA TROCA DE MENSAGENS DO VCBP

Conforme detalhado na Seção 4.5.2, o VCBP utiliza metadados – definidos como *vlsbn* e *mlsbn* –, que são transmitidos nas mensagens e armazenados no *buffer* local com o objetivo de permitir que cada membro do grupo veicular tenha conhecimento sobre as

transmissões realizadas pelos veículos e realizar retransmissões para minimizar as perdas dos canais, além de contribuir com o requisito de entrega confiável.

Para possibilitar uma melhor compreensão sobre a difusão de mensagens do VCBP, as Figuras 4.4, 4.5, 4.6 e 4.7 ilustram o processo de difusão de mensagens em um grupo veicular com 3 veículos: p_1 , p_2 e p_3 .

Inicialmente, a Figura 4.4 detalha a difusão da primeira mensagem de p_1 para o grupo veicular. Antes de enviá-la, p_1 armazena o $m.bn$ em $vlsbn$, que contém a demarcação (em cada posição) dos números de blocos causais disseminados pelos veículos, na sua $mlsbn$ local – ou seja, $mlsbn$ de p_1 . Como p_1 não havia recebido nenhuma mensagem, ele inseriu no seu $vlsbn$ os valores $\{1,0,0\}$ para informar que o seu número de bloco causal era igual a 1 e que não conhecia, até o momento, nenhuma mensagem de bloco causal disseminado por p_2 e p_3 e, portanto, os seus respectivos valores estavam zerados. Após a atualização do $vlsbn$ e a sua devida inserção na $mlsbn$ do veículo local – armazenada dentro da $glsbn$ –, o veículo emissor deve inserir, na mensagem, os $vlsbn$ relativos ao conhecimento das disseminações de todos os veículos, contendo os últimos blocos causais, recebidos em sequência e sem descontinuidade – estes blocos ficam armazenados nas $mlsbn$ local de cada veículo, conforme ilustrado na parte direita da Figura 4.4 que contém os respectivos $vlsbn$ recebidos. Na parte esquerda da ilustração, o veículo p_1 não havia recebido nenhuma mensagem anterior, por isso não inseriu nenhuma informação na $mlsbn$ da mensagem a ser disseminada, referente aos processos p_2 e p_3 . Por fim, a mensagem gerada foi difundida nos canais de comunicação. Os demais veículos do grupo, ao recebê-la, extraíram os $vlsbn$ contidos na mesma e os inseriu na $mlsbn$ do veículo remetente, dentro da $glsbn$. Ressaltamos que o passo de inserção dos metadados na mensagem e a sua devida inserção na bm deve ser realizado antes de cada envio a ser efetuado pelos veículos.

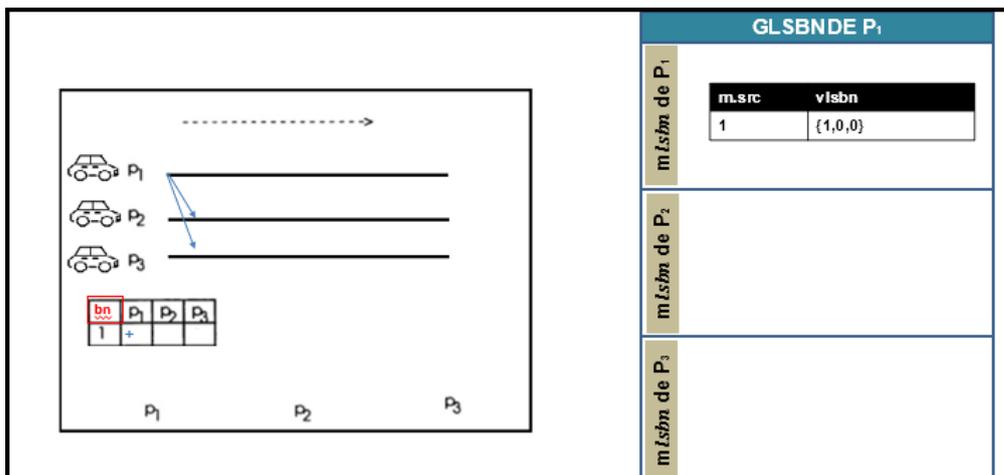


Figura 4.4: Disseminação da Primeira Mensagem do Grupo Utilizando o VCBP

Na Figura 4.5, o veículo p_3 se prepara para difundir a sua primeira mensagem, que também contém o $bc_i = 1$ correspondente à sua primeira difusão. Antes de realizar a

difusão, p_3 armazena o seu conhecimento local sobre as difusões no $vlsbn$ definido como $\{1,0,1\}$ na sua $mlsbn$ – ou seja, $mlsbn$ de p_3 . Este $vlsbn$ contém as seguintes informações: o primeiro valor do vetor corresponde a primeira mensagem difundida por p_1 que contém $bn = 1$; o segundo valor corresponde a ausência de difusões do processo p_2 ; e, o terceiro valor corresponde a primeira difusão que será realizada pelo próprio processo p_3 (com $bn = 1$). Além do $vlsbn$ recém-gerado, o veículo p_3 já havia recebido o $vlsbn$ do veículo p_1 , que foi inserido na sua $mlsbn$ local e incluído na mensagem recém-gerada com o objetivo de permitir o conhecimento sobre as disseminações de p_1 (definida como $mlsbn$ de p_1 pertencente a $glsbn$). Após as inserções dos metadados, a mensagem foi difundida nos canais de comunicação do grupo, como ocorreu na Figura 4.4. Assim, é possível concluir que a $glsbn$, que armazena o conhecimento sobre as difusões de cada processo a partir das suas respectivas $vlsbns$, deve armazenar os $vlsbns$ gerados pelo veículo local e também os $vlsbns$ recebidos dos demais veículos do grupo.

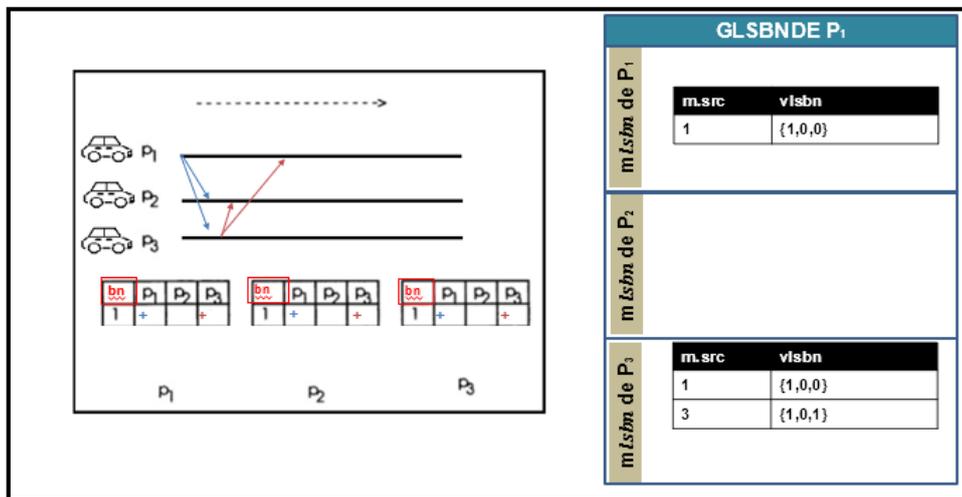


Figura 4.5: Disseminação da Segunda Mensagem do Grupo Utilizando o VCBP

Nas Figuras 4.6 e 4.7, a difusão das mensagens de p_1 e p_2 , com os seus respectivos metadados, foram efetuadas nos canais de comunicação. Contudo, na Figura 4.6 a segunda mensagem de p_1 foi perdida no canal de p_3 . No momento da difusão, p_1 tinha o conhecimento das difusões recebidas pelo veículo p_3 a partir dos dados, armazenados localmente, na $mlsbn$ de p_3 que continha os seguintes valores: $\{1,0,1\}$, difundido por p_1 ; e, $\{1,0,1\}$ difundido por p_3 . Portanto, p_1 adicionou os metadados relativos ao conhecimento das disseminações recebidas localmente (ou seja, da $mlsbn$ de p_1), contendo $bn = 2$, além dos bns disseminados em sequência e sem descontinuidade dos demais veículos que correspondeu ao seguinte vetor $vlsbn$: $\{2,0,1\}$. A mensagem recebida também transportou o vetor $vlsbn$, correspondente ao conhecimento da $mlsbn$ de p_3 que continha o conhecimento dos bns , gerados em sequência e sem descontinuidade, das primeiras disseminações geradas por p_1 e por p_3 – ou seja $bn = 1$, e de nenhuma disseminação efetuada do processo p_2 : $vlsbn\{1,0,1\}$.

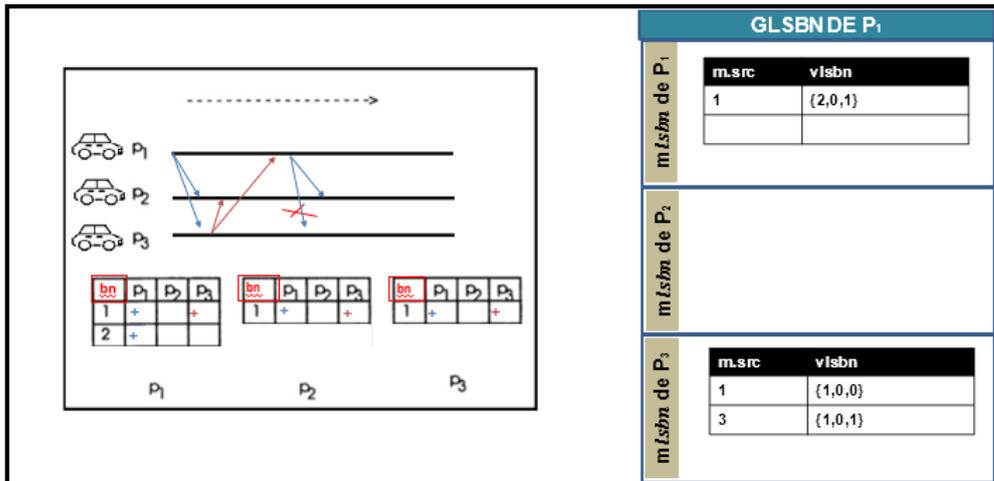


Figura 4.6: Disseminação da Terceira Mensagem do Grupo Utilizando o VCBP

A Figura 4.7 ilustra a disseminação de uma mensagem do veículo p_2 . Entretanto, ao receber a mensagem nos canais, os veículos receptores extraem os *vlsbns* que vieram na *mlsbn* e os inserem na sua *glsbn* local, para efetuarem comparações de recebimento entre as matrizes de cada veículo. Portanto, ao analisar a *bm* e a *glsbn* de p_1 , é possível perceber que a *mlsbn* de p_2 possui um *vlsbn* que contém a demarcação do $bn=2$ para o veículo p_1 e este mesmo conhecimento está registrado na *bm* de p_1 , porém, não está contido na *mlsbn* de p_3 , o que sinaliza uma possível perda de mensagem. Desta forma, é possível concluir que a utilização destes metadados, disseminados nas mensagens, podem contribuir para a detecção de perdas de mensagens – em cada veículo do grupo – e para a devida recuperação destas, visto que todos os veículos realizam o armazenamento nas suas respectivas *bms* locais.

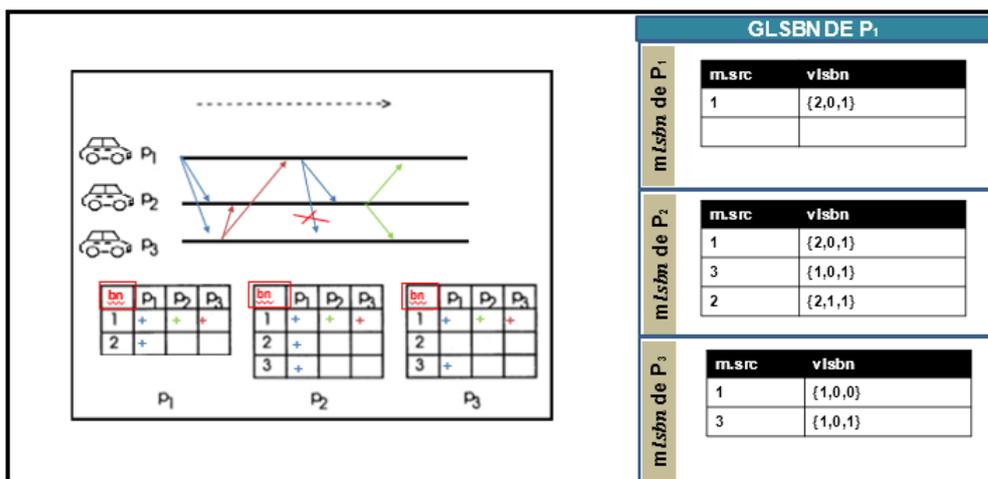


Figura 4.7: Disseminação da Quarta Mensagem do Grupo Utilizando o VCBP

4.7 DETALHAMENTO DO ALGORITMO PROPOSTO NO VCBP

Nesta seção, são apresentadas as principais tarefas, operações e funções auxiliares implementadas no *VCBP*.

4.7.1 Principais Tarefas Implementadas no VCBP

O VCBP é composto por cinco tarefas principais (ver Algoritmo 1): *Init*, *Beaconing*, *BlockExpiration*, *Delivery* e *Retransmission*.

Algoritmo 1: VCBP (Parte 01) – Principais Tarefas

```

1 task Init at  $p_i \in g$ 
2    $bc_i \leftarrow 0; ld_i \leftarrow 0; bm_i \leftarrow \{\};$ 
3    $mbv_i \leftarrow \{\}; glsbn_i \leftarrow \{\};$ 
4   start task Beaconing;
5 task Beaconing at  $p_j \in g$ 
6   loop forever
7      $sleep(beaconInterval());$ 
8      $beacon \leftarrow newBeacon();$ 
9      $rmcast(beacon, beaconChannel());$ 
10 task BlockExpiration(bn) at  $p_j \in g$ 
11   if  $\neg sstable(bn)$  then
12     for each  $p_i \in g$  do
13        $null.bn \leftarrow bn;$ 
14        $null.src \leftarrow p_i;$ 
15        $bm_j[bn][p_i] \leftarrow null;$ 
16   start task Delivery;
17 task Delivery at  $p_j \in g$ 
18   while ( $sstable(ld_j + 1)$ ) do
19      $ld_j \leftarrow ld_j + 1;$ 
20     cancel task BlockExpiration( $ld_j$ );
21     for each  $m \in bm_j[ld_j]$  order by  $m.src$  do
22        $bc_j \leftarrow max(bc_j, m.bn);$ 
23       if ( $m \neq null$ ) then
24         deliver  $m$  at  $p_j$ 
25 task Retransmission(m) at  $p_j \in g$ 
26   if  $\neg expired(m.bn) \wedge (\exists p_i \in g | miss(p_i, m) \wedge distance(p_i, p_j) \leq R)$  then
27     send  $m$  to  $g - \{p_j\}$  using default;

```

Na criação do grupo g , cada processo $p_i \in g$ usa a tarefa *Init* (Algoritmo 1, linhas 1–4) para inicializar suas variáveis locais (i.e. bc_i , ld_i , bm_i , mbv_i e $glsbn_i$) e disparar a tarefa *Beaconing*, responsável pelo envio periódico de *beacons*. A cada intervalo de emissão de *beacons*, i.e. $beaconInterval()$, a tarefa *Beaconing* executando em $p_j \in g$ cria um novo *beacon* e usa a operação de difusão confiável do VCBP (*rmcast*) para distribuí-lo pelo canal de controle (*beaconChannel*) para os demais processos do grupo (Algoritmo 1, linhas 6–9). Os *beacons* disseminados periodicamente garantem a completude dos blocos causais e a distribuição das informações de controle necessárias para garantia das propriedades do VCBP.

A tarefa *BlockExpiration* (Algoritmo 1, linhas 10-16) é responsável por verificar o prazo de entrega de um bloco causal e é escalonada toda vez que um bloco causal é criado na matriz de blocos. No término do prazo de entrega de um bloco causal, caso ele não tenha atingindo as condições para entrega de suas mensagens, a tarefa *BlockExpiration* anula as mensagens do mesmo (ver Algoritmo 1, linhas 11-15). Após anular as mensagens de um determinado bloco causal, a tarefa *BlockExpiration* ativa a tarefa *Delivery* para realizar a entrega de outros blocos causais que possivelmente tenham atingindo as condições de entrega de mensagens (ver Algoritmo 1, linha 16).

A tarefa *Delivery* (Algoritmo 1, linhas 17-24) é ativada, não apenas quando o prazo de entrega de um bloco causal expira, mas também quando um processo do grupo recebe e insere uma mensagem em sua matriz de blocos causais. *Delivery* entrega mensagens em ordem crescente do número de bloco causal. Ela usa a variável ld_i para identificar o número do último bloco causal cujas mensagens foram entregues por p_i . Quando ativada, ela iterativamente verifica se o próximo bloco causal ($ld_i + 1$) alcançou o requisito de **super-estabilidade**, podendo ter suas mensagens entregues (Algoritmo 1, linha 18). Caso o bloco tenha alcançado super-estabilidade, a tarefa *Delivery* incrementa ld , cancela qualquer tarefa de expiração de bloco que esteja agendada para o respectivo bloco causal e realiza a entrega das mensagens não nulas do bloco causal, em ordem crescente dos identificadores dos processos do grupo (Algoritmo 1, linhas 19-24). A tarefa *Delivery* em p_j repete iterativamente esses passos até que não hajam mais blocos causais **super-estáveis** na matriz de blocos bm_j .

A tarefa *Retransmission* (Algoritmo 1, linhas 25-27) é responsável pela retransmissão de mensagens que não foram recebidas por algum processo do grupo. Para isso, toda vez que um processo p_j recebe e insere em sua matriz bm_j uma mensagem m , ele agenda a ativação da tarefa *Retransmission*, caso identifique que há mensagem em sua matriz de blocos que não foi recebida por algum processo do grupo. Quando a tarefa *Retransmission* é executada para alguma mensagem m , ela retransmite m para os membros do grupo dentro de seu alcance, caso verifique que o bloco causal que contém m ainda esteja dentro do prazo de validade e que exista algum processo p_i em seu raio de retransmissão R que ainda não recebeu m (Algoritmo 1, linhas 26-27).

4.7.2 Principais Operações do VCBP

No *VCBP*, *rmcast* e *receive* são as duas principais operações (ver Algoritmo 2).

A operação *rmcast* responsável por realizar a difusão confiável de uma mensagem m , usando um determinado canal de comunicação *channel* (ver Algoritmo 2, linhas 28-39).

A cada mensagem m a ser difundida por p_i , a operação *rmcast* insere na mesma a identificação do seu remetente ($m.src$), de seu número de bloco ($m.bn$) e o respectivo prazo de entrega de seu respectivo bloco causal ($m.d$) – ver Algoritmo 2, linhas 29-32). Em seguida, p_i insere a mensagem m no bloco causal correspondente em sua matriz de blocos bm_i (Algoritmo 2, linha 33). Toda vez que uma mensagem é inserida em sua matriz de blocos, p_i deve escalonar a tarefa de expiração de blocos causais (*BlockExpiration*), caso ela ainda não tenha sido escalonada (Algoritmo 2, linhas 34-35). Além disso, antes de difundir a mensagem m para os demais membros do grupo, p_i deve atualizar as matrizes

Algoritmo 2: VCBP (Parte 02) – Operações *rmcast* e *receive*

```

28 proc rmcast(m, channel) at  $p_i \in g$ 
29   m.src  $\leftarrow p_i$ ;
30   bci  $\leftarrow bc_i + 1$ ;
31   m.bn  $\leftarrow bc_i$ ;
32   m.d  $\leftarrow deadline(m.bn)$ ;
33   bmi[m.bn][m.src]  $\leftarrow m$ ;
34   if  $\neg scheduled(BlockExpiration(m.bn))$  then
35      $\lfloor$  schedule task BlockExpiration(m.bn) at m.d;
36   mbvi[m.src]  $\leftarrow m.bn$ ;
37   glsbni[m.src][m.src][m.src]  $\leftarrow m.bn$ ;
38   m.mlsbn  $\leftarrow glsbn_i[m.src]$ ;
39   send m to  $\forall p_j \in g - \{p_i\}$  using channel;
40 upon receive(m) at  $p_j \in g$  do
41   if (m.src  $\in g \wedge \neg expired(m.bn)$ ) then
42     if  $\neg received(m)$  then
43       bmj[m.bn][m.src]  $\leftarrow m$ ;
44       if  $\neg scheduled(BlockExpiration(m.bn))$  then
45          $\lfloor$  schedule task BlockExpiration(m.bn) at m.d;
46       if mbvj[m.src]  $< m.bn$  then
47          $\lfloor$  mbvj[m.src]  $\leftarrow m.bn$ ;
48          $\lfloor$  glsbnj[m.src]  $\leftarrow m.mlsbn$ ;
49       b'  $\leftarrow lsb_n(m.src)$ ;
50       glsbnj[p_j][p_j][m.src]  $\leftarrow \max(b', glsbn_j[p_j][p_j][m.src])$ ;
51       for all ( $m' \in bm_j | missed(m')$ ) do
52          $\lfloor$  schedule task Retransmission(m') at GPSj.time() + rto();
53       for all ( $m' \in bm_j | \neg missed(m')$ ) do
54          $\lfloor$  cancel task Retransmission(m');
55       start task Delivery;
56     else
57        $\lfloor$  rt  $\leftarrow GPS_j.time() + rto()$ ;
58        $\lfloor$  reschedule task Retransmission(m) at rt;

```

locais usadas na detecção de mensagens perdidas e na verificação de super-estabilidade de bloco, i.e. *mbv_i* e *glsbn_i*, e inserir em *m.mlsbn* as informações de *mlsbn_i* (i.e. *glsbn_i[p_i]* ou *glsbn_i[m.src]*) – ver Algoritmo 2, linhas 36–38. Por fim, *p_i* difunde a mensagem *m* para os demais membros do grupo, usando o canal especificado (Algoritmo 2, linha 39).

No *VCBP*, cada mensagem *m* recebida é repassa a um processo *p_j* usando a operação *receive* (ver Algoritmo 2, linhas 40–58). Quando uma mensagem *m* é recebida em *p_j*, ela será descartada caso não pertença a um bloco causal dentro do prazo de entrega ou não tenha sido originada por um membro do grupo (Algoritmo 2, linha 41). Caso contrário, se *m* tiver sido recebida em duplicidade por *p_j*, então um novo *timeout* de retransmissão é calculado e a tarefa de retransmissão da mensagem *m* (*Retransmission*(*m*)) é reagendada em *p_j* (Algoritmo 2, linhas 42, 57 e 58). Quando um mensagem *m* enviada por um processo do grupo, dentro do prazo de entrega de seu bloco causal, é recebida por um processo *p_j* pela primeira vez, ele insere a mensagem *m* em sua matriz de blocos *bm_j* e escalona a tarefa de expiração de bloco causal (*BlockExpiration*(*m.bn*)), caso esta ainda não tenha

sido escalonada (ver Algoritmo 2, linhas 43–45). Em seguida, p_j usa as informações da mensagem m para, caso necessário, atualizar as matrizes locais usadas na detecção de mensagens perdidas e na verificação de super-estabilidade de bloco, i.e. mbv_i e $glsbn_i$ (Algoritmo 2, linhas 46–48). Além disso, p_j atualiza seu $vlsbn_j$ (i.e. $glsbn_j[p_j][p_j]$) com as informações recebidas do processo que originou a mensagem m (Algoritmo 2, linhas 49–50). Depois disso, se p_j detecta que existe alguma mensagem m' em sua matriz de blocos que não foi recebida por algum processo do grupo, ele escalona a tarefa de retransmissão de mensagens (*Retransmission*(m')), ver Algoritmo 2, linhas 51–52. Além disso, p_j cancela todas as retransmissões de mensagem m' que possuam retransmissões escalonadas, mas que já foram recuperadas pelos demais processos do grupo (Algoritmo 2, linhas 53–54). Por fim, uma vez que uma nova mensagem foi inserida em bm_j e suas matrizes de controle mbv_j e $glsbn_j$ foram atualizadas, p_j dispara a tarefa de entrega de mensagens (*Delivery*) para verificar se existem blocos causais com mensagens em condições de serem entregues à aplicação (Algoritmo 2, linha 55).

4.7.3 Funções e Procedimentos Auxiliares Importantes para o VCBP

O *VCBP* é composto por um conjunto de funções e procedimentos auxiliares que são essenciais em seu funcionamento (Algoritmo 3).

As tarefas *Delivery* e *BlockExpiration* utilizam a função *sstable*(bn) (Algoritmo 3, linhas 59–60) para verificar a condição de **super-estabilidade** de um bloco causal de número bn e garantir o **Acordo** na entrega das mensagens (ver discussão da Seção 4.5.3).

A função *expired*(bn) é utilizada, no recebimento (i.e. operação *receive*(m)) e na retransmissão (i.e. tarefa *Retransmission*(m)) de uma mensagem m , para verificar se o bloco causal relacionado (i.e. $bn = m.bn$) está dentro de seu prazo de validade (Algoritmo 3, linhas 61–62).

O *timeout* para retransmissão de uma mensagem é calculado usando a função *rto*. Quando uma perda de mensagem é detectada, a função *rto* calcula o *timeout* de retransmissão como sendo metade do período de difusão de *beacon* (i.e. *beaconInterval*()/2) acrescido de um valor aleatório (*backoff*), para evitar colisões entre retransmissões de mensagens realizadas por processos distintos (Algoritmo 3, linhas 63–66). O identificador do processo é usado como semente na escolha de um valor aleatório no intervalo de *backoff*, definido entre bo_{min} e bo_{max} .

A função *missed*(m) é utilizada por um processo $p_j \in g$ para verificar se existe algum processo do grupo que não tenha recebido a mensagem m – i.e. *missed*(m) será verdade se existe algum $p_i \in g$ para o qual a função *miss*(p_i, m) retorna verdadeiro (Algoritmo 3, linhas 67–68). Para verificar se um processo p_i não recebeu a mensagem m , p_j usa a função *miss*(p_i, m) para fazer uma análise similar ao que é proposto em (LIMA; MACÊDO, 1999). Para tanto, a função *miss* usa o vetor mbv_j para verificar qual foi o menor dos maiores valores de número de bloco que foi recebido (ou enviado) em mensagens de processos do grupo (i.e. $\min(mbv_j)$) e o vetor $vlsbn_i$ conhecido por p_j (i.e. $glsbn_j[p_j][p_i]$) para saber qual o último número de bloco que p_i recebeu em sequência (e sem descontinuidade) a partir de mensagens originadas em p_k . Desse modo, se $m.bn < \min(mbv_j)$ significa que todos os processos do grupo (o que inclui p_i) receberam mensagens com

Algoritmo 3: VCBP (Parte 03) – Procedimentos e Funções Auxiliares

```

59 func sstable(bn) at  $p_j \in g$ 
60 | return  $((\forall p_k \in g)(\exists bm_j[bn][p_k] \wedge ((\forall p_l \in g)(\forall p_m \in g)(bn \leq glsbn_j[p_k][p_l][p_m]))));$ 
61 func expired(bn) at  $p_j \in g$ 
62 | return  $GPS_j.time() > deadline(bn);$ 
63 func rto() at  $p_j \in g$ 
64 |  $bointerval \leftarrow [bo_{min}, bo_{max}];$ 
65 |  $boseed \leftarrow p_j;$ 
66 | return  $beaconInterval()/2 + random(bointerval, boseed);$ 
67 func missed(m) at  $p_j \in g$ 
68 | return  $(\exists p_i \in g | miss(p_i, m));$ 
69 func miss(pi, m) at  $p_j \in g$ 
70 | if  $(p_i \in g \wedge m \in bm_j)$  then
71 | | return  $(\exists p_k \in g | p_i \neq p_k \neq p_j \wedge m.src = p_k \wedge glsbn_j[p_j][p_i][p_k] < m.bn < \min(mbv_j));$ 
72 | return false;
73 func lsbn(pk) at  $p_j \in g$ 
74 |  $bn \leftarrow mbv_j[p_k];$ 
75 | return  $lsbn(p_k, bn);$ 
76 func lsbn(pk, bn) at  $p_j \in g$ 
77 | if  $bn \leq 0$  then
78 | | return 0;
79 | if  $bn \leq ld_j$  then
80 | | return bn;
81 |  $m \leftarrow bm_j[bn][p_k];$ 
82 | if rsequence(m) then
83 | | return bn;
84 | return  $lsbn(p_k, bn - 1);$ 
85 func rsequenced(m) at  $p_j \in g$ 
86 | return  $(\forall bn((ld_j < bn \leq m.bn) \wedge (\exists m' \in bm_j | m'.bn = bn \wedge m'.src = m.src)));$ 

```

número de bloco maior que $m.bn$, mas, por outro lado, se $m.bn > vlsbn_i[p_k]$, então $m.bn$ ainda não foi recebida por p_i , indicando que pode ter ocorrido uma falha no canal de comunicação (Algoritmo 3, linhas 69–71).

Um processo p_j utiliza a função $lsbn(p_k)$ para verificar qual o maior valor de número de bloco que recebeu em sequência e sem descontinuidade a partir de mensagens originadas em p_k (Algoritmo 3, linhas 73–75). Para isso, a partir do maior valor de número de bloco bn que recebeu em mensagens originadas em p_k (i.e. $bn = mbv_j[p_k]$), usa a função $lsbn(p_k, bn)$ para analisar recursivamente sua matriz de blocos causais bm_j , em busca do maior número de bloco, menor que bn , que tenha recebido em sequência e sem descontinuidade a partir de mensagens originadas em p_k (Algoritmo 3, linhas 76–84).

Neste capítulo apresenta-se uma avaliação de desempenho do VCBP e discutem-se os resultados obtidos quando comparado com outros protocolos de comunicação em grupo existentes na literatura.

AVALIAÇÃO DE DESEMPENHO

A avaliação de desempenho do *VCBP* foi realizada em ambiente simulado, observando sua latência e taxa de entrega de mensagens, quando comparado com outras abordagens de comunicação em grupo similares existentes na literatura. O *VCBP* e demais protocolos foram completamente implementados na linguagem C++, usando o VEINS (*Vehicles In network Simulation*), um framework de código aberto dedicado a simulação de redes veiculares (SOMMER et al., 2019). O Diagrama de Classe utilizado na implementação do *VCBP* está disponível no Apêndice B.

Para viabilizar a simulação tanto da comunicação sem fio entre os veículos quanto dos cenários de tráfego em ambiente urbano, o VEINS integra o simulador de rede OM-NeT++, *Object Modular Network Testbed in C++* (VARGA; HORNIG, 2010), e o simulador de tráfego veicular rodoviário SUMO, *Simulation of Urban MObility* (BEHRISCH et al., 2011). Os cenários de simulação foram desenvolvidos utilizando VEINS 4.6, OM-NeT++ 5.1 e SUMO 0.30.0. O Apêndice A ilustra detalhes do acoplamento entre os simuladores utilizados.

Maiores detalhes sobre a avaliação de desempenho do *VCBP* são apresentados nas seções a seguir. Para tanto, a Seção 5.1 descreve o ambiente de simulação, a Seção 5.2 apresenta a parametrização dos protocolos usados na comparação, a Seção 5.3 lista as métricas de desempenho usadas na avaliação e, por fim, a Seção 5.4 descreve os experimentos realizados e discute os respectivos resultados obtidos.

5.1 DESCRIÇÃO DO AMBIENTE DE SIMULAÇÃO

Nos cenários simulados, as aplicações que executam nos veículos enviam mensagens diretamente na camada de enlace, usando o padrão de comunicação veicular *IEEE 802.11p* e seguindo a arquitetura WAVE 2.2.1 –(JIANG; DELGROSSI, 2008; ALVES et al., 2009).

Os enlaces sem fio utilizados pelos veículos adotam potência de transmissão que permite comunicação em um raio de alcance de até 1000m e possuem taxa de transferência

de $6Mbps$, como é típico em redes veiculares baseadas no padrão *IEEE 802.11p* (JIANG; CHEN; DELGROSSI, 2008; WANG et al., 2013). Os experimentos consideraram que os enlaces de comunicação sem fio podem apresentar interferências que implicam em perdas de mensagens variando entre 0% e 10%. As perdas de mensagens nos canais de comunicação são induzidas randomicamente usando uma distribuição de probabilidade binomial. A Tabela 5.1 apresenta um resumo dos parâmetros de rede usados na simulação.

Tabela 5.1: Parâmetros de configuração de rede

Protocolo de comunicação da camada MAC	IEEE 802.11p
Arquitetura de comunicação	WAVE
Taxa de transferência dos enlaces de comunicação sem fio (camada MAC)	$6Mbps$
Raio de alcance da comunicação sem fio (camada MAC)	$1000m$
Percentual de perda de mensagens nos enlaces de comunicação	0 e 10 %

Os veículos utilizam um serviço de *Beaconing* para, a cada segundo, trocar informações de mobilidade (e.g. posição, velocidade, aceleração etc.). Cada *beacon* carrega não apenas informações de mobilidade, mas também informações de controle do protocolo de comunicação. Por conta disto, os tamanhos dos *beacons* podem variar entre 300 e 512 bytes. A Tabela 5.2 apresenta um resumo dos parâmetros do serviço de *Beaconing*.

Tabela 5.2: Parâmetros de configuração do serviço de *Beaconing*

Tipo da tarefa de envio de <i>beacons</i>	Periódica
Período de envio de <i>beacons</i> (<i>PEB</i>)	$1000ms$
Tamanho dos <i>beacons</i>	300 a 512 bytes

Em relação aos aspectos de tráfego veicular das simulações, os veículos são caminhões com $8m$ de comprimento, se deslocando com velocidade média de $50km/h$ ($\approx 14m/s$) em pista reta sem obstáculos (i.e. prédios, pontes, equipamentos da via, acidentes etc.). Os caminhões se organizam em um pelotão (*platoon*) e colaboram via troca de mensagens para permitir controle cooperativo de suas velocidades. Nessa direção cooperativa, adota-se o modelo de Krauss (1998), uma abordagem do tipo carro-seguidor (*car-following*), na qual os veículos se organizam em fila, garantindo uma distância mínima entre si (*minimum gap*, denotado por *mingap*) e controlando as suas respectivas velocidades considerando as variações de velocidade do veículo a sua frente. Esse modelo emula um controlador de velocidade, e é bastante rigoroso no atendimento dos requisitos de direção e, em cenários livres de obstáculos ou curvas, consegue manter a velocidade e a distância entre os veículos dentro do desejado, o que permite avaliar os impactos específicos dos protocolos de comunicação em grupo, desconsiderando deficiências específicas de um ou de outro algoritmo de controle de velocidade.

Nos experimentos, dependendo do cenário simulado, um pelotão pode ser composto por 2, 4 ou 8 veículos. A escolha por considerar um pelotão com tamanho máximo de 8 veículos foi realizada para evitar instabilidades no controle de velocidade (ZHOU; ZHU,

2021; BIEKER-WALZ et al., 2017). Configurou-se uma distância mínima entre veículos de $2,5m$ (i.e. $mingap = 2,5$), de modo a orientar o controle simulado de velocidade a garantir uma distância de $2,5m$, em cenários sem manobras de mudança de faixa, obstáculos ou ultrapassagens. A Tabela 5.3 resume os parâmetros de mobilidade usados na simulação.

Tabela 5.3: Parâmetros de configuração de mobilidade

Tipo de via	via reta e sem obstáculos
Tamanho do pelotão	2, 4 e 8 veículos
Tipo do veículo	caminhão
Tamanho do veículo	$8m$
Distância entre veículos	$2,5m$
Velocidade do veículo	$14m/s$ ($50km/h$)
Controle de direção	abordagem de Krauss (1998)

5.2 CONFIGURAÇÃO DOS PROTOCOLOS DE COMUNICAÇÃO EM GRUPO

Nos experimentos realizados, o desempenho do *VCBP* é comparado com o desempenho dos protocolos *RCBP*, *M-RBP* e Δ -*CBP*, todos discutidos no Capítulo 3.

O *M-RBP* usa uma rodada TDMA de tamanho igual ao período de emissão de *beacons* (i.e. $PEB = 1000ms$). Assim, para um pelotão com n veículos, cada veículo transmite durante pelo menos $1000/n$ milissegundos – isto é, 500 , 250 e $125ms$ para pelotões com 2, 4 e 8 veículos, respectivamente.

As informações de controle do *RCBP* são inseridas a cada *beacon* difundido pelo serviço de *Beaconing*. O mecanismo de retransmissão do *RCBP* em cada veículo sorteia um *timeout* de retransmissão dentro de uma janela de tempo entre 0 e $1000ms$ – mais especificamente, a janela de retransmissão é limitada pelo período de envio de informações de controle (i.e. *time-silence*), conforme sugerido originalmente por (LIMA; MACÊDO, 1999).

As informações de controle do *VCBP* também são inseridas nos *beacons*. O mecanismo de retransmissão do *VCBP* foi configurado com uma janela de *backoff* entre 0 e $0.19ms$. Assim, uma vez detectada uma perda de mensagem, uma retransmissão no *VCBP* pode acontecer entre 500 e $500.19ms$ – isto é, metade do período de *beacon* somado com um valor aleatório dentro da janela de *backoff*¹. O raio de retransmissão R do *VCBP* é calculado em função do tamanho dos veículos e do espaçamento entre eles. Considerou-se um raio de retransmissão para que um veículo possa alcançar veículos vizinhos localizados imediatamente a frente e atrás, conseqüentemente $R = 18,5m$ (i.e. $R = 2 * 8m + 2,5m$). O *VCBP* foi configurado com um prazo de entrega de mensagens de $5000ms$, o que equivale a 5 vezes o período de emissão de *beacons*.

Por fim, o Δ -*CBP* foi configurado para que cada mensagem tenha um tempo de vida de $5000ms$, equivalente a cinco vezes o período de emissão de *beacons*.

¹ver discussão no Capítulo 4.

A Tabela 5.4 apresenta um resumo dos parâmetros de configuração utilizados pelos protocolos de comunicação em grupo considerados na avaliação de desempenho.

Tabela 5.4: Parâmetros de configuração dos protocolos de comunicação em grupo

Janela de <i>backoff</i> usada pelo <i>VCBP</i>	[0, 0.19ms]
Janela de retransmissão no <i>VCBP</i>	[500, 500.19ms];
Janela de retransmissão no <i>RCBP</i>	[0, 1000ms];
Tamanho da rodada TDMA no <i>M-RBP</i>	1000ms;
Prazo para entrega de mensagens no <i>VCBP</i>	5000ms
Raio de retransmissão no <i>VCBP</i>	18, 5m
Tempo de vida das mensagens no Δ - <i>CBP</i>	1000ms
Prazo de entrega das mensagens no <i>VCBP</i>	5000ms

5.3 MÉTRICAS DE DESEMPENHO

Os protocolos de comunicação em grupo foram avaliados considerando as seguintes métricas de desempenho:

- **Latência de Recepção de Mensagens** – corresponde a latência média entre o envio e o recebimento das mensagens, a qual considera todo o intervalo de tempo entre o envio da mensagens e sua recepção no destino. No caso dos protocolos que possuem mecanismos de detecção e recuperação de mensagens perdidas, a latência de recepção inclui o tempo usado para recuperar possíveis mensagens perdidas.
- **Latência de Entrega de Mensagens** – corresponde a latência média entre a difusão e a entrega das mensagens, inclui a latência de recepção e o intervalo de tempo no qual as mensagens recebidas ficam retidas até serem entregues para a aplicação – quando as condições de entrega são alcançadas.
- **Taxa de Mensagens Recebidas** – corresponde ao percentual das mensagens difundidas que foram recebidas em seu destino. Por exemplo, se uma difusão deve alcançar 4 veículos, mas apenas 2 receberam a mensagem, há 50% de recebimento. Portanto, a taxa de mensagens recebidas será uma média desses percentuais de recebimento. No caso dos protocolos que possuem mecanismos de detecção e recuperação de mensagens perdidas, as mensagens recuperadas são contabilizadas como recebidas.
- **Taxa de Mensagens Entregues** – corresponde ao percentual das mensagens recebidas que foram entregues para a aplicação veicular.
- **Taxa de Mensagens Bloqueadas** – corresponde ao percentual das mensagens recebidas, mas que não foram entregues para a aplicação dos veículos.

5.4 EXPERIMENTOS

Nas simulações, foram definidos dois grupos de experimentos:

- **Cenário 1 – sem interferências na comunicação veicular:** nesse primeiro grupo de experimentos não há indução de perda de mensagens nos canais de comunicação. Nesse grupo todas mensagens difundidas são recebidas no destino. Assim, os mecanismos de detecção e recuperação de mensagens do *M-RBP*, *RCBP* e *VCBP* nunca são utilizados. Portanto, esse cenário é usado apenas para verificar exclusivamente o desempenho dos protocolos em termos da latência para entrega das mensagens.
- **Cenário 2 – com interferências na comunicação veicular:** nesse grupo de experimentos é induzido 10% perdas de mensagens, com o objetivo de simular interferências e desconexões. Nesse cenário, é possível observar os impactos tanto das interferências quanto dos mecanismos de detecção e recuperação de mensagens no desempenho dos protocolos.

Os resultados obtidos para cada um desses grupos de experimentos são apresentados e discutidos nas Seções 5.4.1 e 5.4.2.

5.4.1 Cenário 1: Comunicação Veicular sem Interferências

A Figura 5.1 apresenta o desempenho dos protocolos em termos da latência de recebimento e da taxa de recebimento de mensagens – a latência é apresentada em faixas no eixo x , enquanto que o eixo y apresenta os valores da taxa de recebimento de mensagens. Quando não há interferência na comunicação veicular, todos os protocolos recebem e entregam 100% das mensagens difundidas em até 100ms (ver Figura 5.1). Portanto, os valores de latências de entrega maiores que 100ms refletem quanto tempo as mensagens ficaram bloqueadas, esperando no *buffer* de recebimento até serem entregues à camada de aplicação.

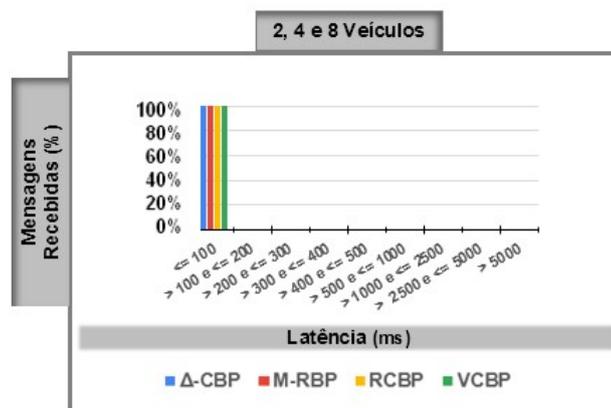


Figura 5.1: Desempenho dos protocolos em termos da latência de recebimento e da taxa de recebimento de mensagens nos experimentos sem interferência na comunicação veicular

A Figura 5.2 apresenta o desempenho dos protocolos em termos da latência de entrega e da taxa de entrega de mensagens – a latência de entrega é apresentada em faixas no eixo x , enquanto que o eixo y apresenta os valores da taxa de entrega de mensagens.

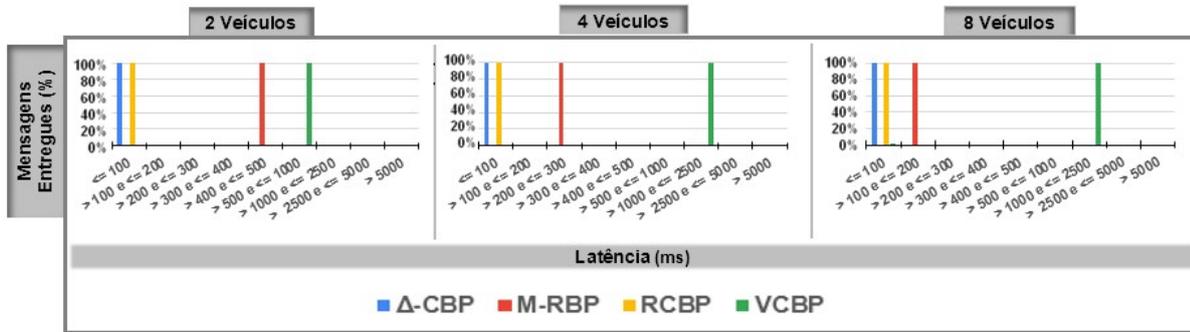


Figura 5.2: Desempenho dos protocolos em termos da latência de entrega e da taxa de entrega de mensagens nos experimentos sem interferência na comunicação veicular

Nos experimentos, o *M-RBP* apresentou latências de entrega entre: 400 e 500ms para pelotões com 2 veículos; 200 e 300ms para pelotões com 4 veículos; e entre 100 e 200ms para pelotões com 8 veículos. Conforme discutido no Capítulo 3, a cada rodada *TDMA*, um veículo é habilitado para difundir uma mensagem de aplicação para o grupo. Para não comprometer o tempo de resposta, na medida em que o número de veículos cresce, o tamanho da rodada *TDMA* diminui. Nos experimentos, os tamanhos das rodadas *TDMA* para esse protocolo foram 500, 250 e 125ms para pelotões com 2, 4 e 8 veículos, respectivamente. Portanto, se não há perda de mensagens, a latência média para a entrega de mensagens também diminui, explicando os resultados obtidos em termos da latência de entrega de mensagens para esse protocolo neste cenário. Por outro lado, esse comportamento é consistente até certo ponto, pois se o tamanho do grupo aumenta muito, o tamanho da rodada *TDMA* torna-se muito pequena e insuficiente para que os veículos realizem todas as etapas de acordo previstas no protocolo.

O *RCBP* e o *Δ-CBP* apresentaram latência de entrega de mensagens menores que 100ms para todos os tamanhos de pelotão considerados. Quando não há perda de mensagem, o envio periódico de mensagem de controle do *RCBP* faz com que a completude de blocos causais aconteça mais rapidamente, o que implica em latências de entrega de mensagens menores. No *Δ-CBP*, cada mensagem transporta metadados com o histórico da difusão causal das mensagens anteriores. Assim, se não há perda de mensagem, as entregas das mensagens acontecem com menores latências.

O *VCBP* obteve latências de entrega entre 500 e 1000ms para pelotões com 2 veículos e entre 1000 e 2500 para pelotões com 4 e 8 veículos. A latência de entrega desse protocolo é maior do que a latência dos demais protocolos avaliados. Isso por que o *VCBP* requer várias rodadas de difusão de *beacons* com os metadados necessários para que os membros do grupo alcancem o acordo para entrega confiável das mensagens. Por isso, pelotões maiores apresentam maiores latências de entrega de mensagens.

5.4.2 Cenário 2: Comunicação Veicular com Interferência

As figuras 5.3, 5.4 e 5.5 apresentam o desempenho dos protocolos em cenários com interferência na comunicação veicular, considerando pelotões compostos por 2, 4 e 8 veículos, respectivamente. Nessas figuras, os gráficos à esquerda apresentam as taxas de mensagens recebidas (eixo y) e as latências de recepção de mensagens (eixo x), enquanto que os gráficos à direita apresentam a taxa de mensagens entregues (eixo y) e a latência de entrega de mensagens (eixo x).

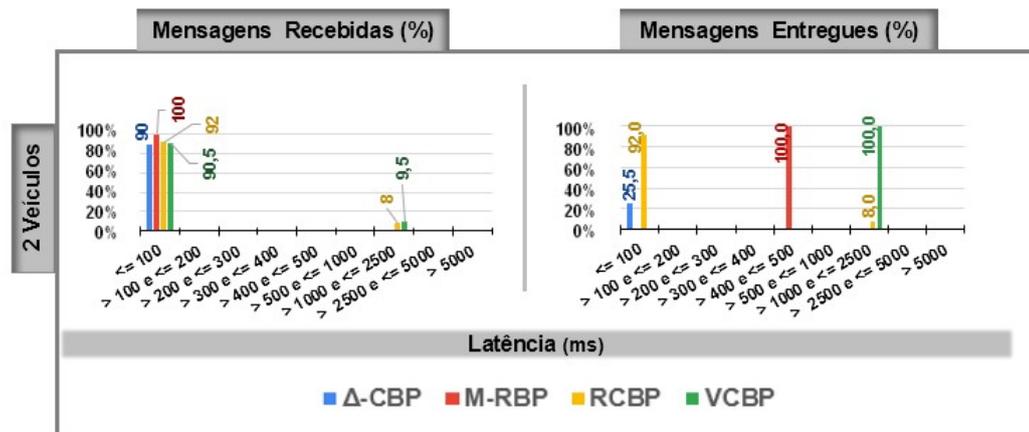


Figura 5.3: Desempenho dos protocolos em termos da taxa de mensagens recebidas, da taxa de mensagens entregues, da latência de recebimento de mensagens e da latência de entrega de mensagens nos experimentos com interferência na comunicação veicular e pelotões compostos por 2 veículos

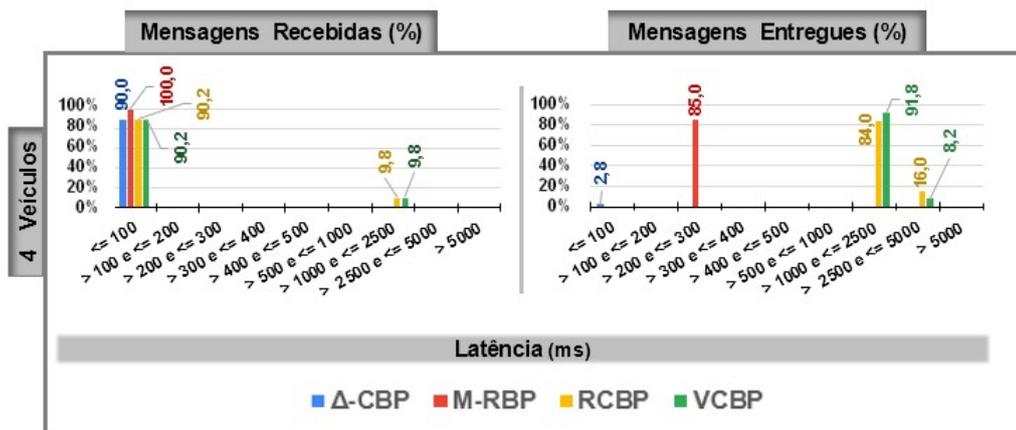


Figura 5.4: Desempenho dos protocolos em termos da taxa de mensagens recebidas, da taxa de mensagens entregues, da latência de recebimento de mensagens e da latência de entrega de mensagens nos experimentos com interferência na comunicação veicular e pelotões compostos por 4 veículos

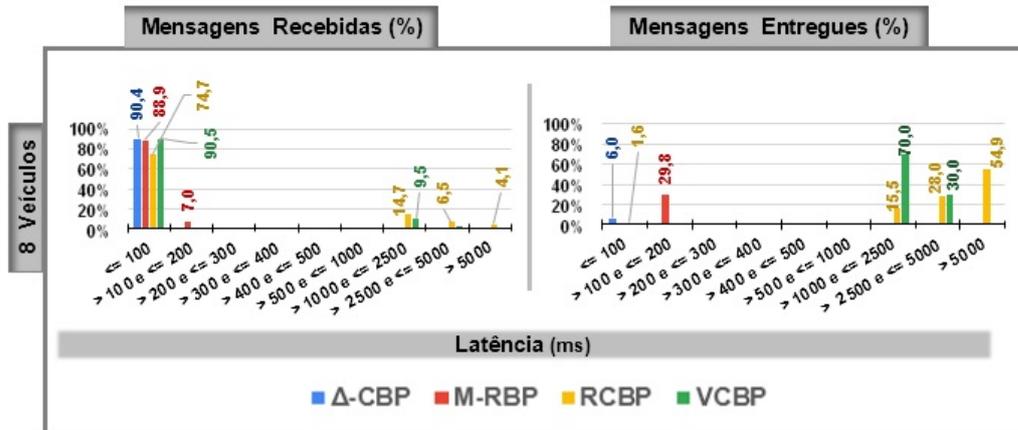


Figura 5.5: Desempenho dos protocolos em termos da taxa de mensagens recebidas, da taxa de mensagens entregues, da latência de recebimento de mensagens e da latência de entrega de mensagens nos experimentos com interferência na comunicação veicular e pelotões compostos por 8 veículos

Conforme pode ser observado nos gráficos das Figuras 5.3, 5.4 e 5.5, nos experimentos com interferência na comunicação veicular, o *RCBP* e o *VCBP* receberam e entregaram 100% mensagens em todas as configurações de pelotão (i.e. 2, 4 e 8 veículos). Esse desempenho é obtido por conta dos mecanismos de detecção e retransmissão de mensagens, que permitiu a recuperação de mensagens perdidas. Para pelotões com 2 e 4 veículos, o *RCBP* e o *VCBP* possuem desempenhos bastante similares em termos da latência de mensagens recebidas (Figuras 5.3 e 5.4). Contudo, para um pelotão com 8 veículos, o *RCBP* apresenta maiores latências de recepção de mensagens, contendo também uma maior variabilidade nas latências de recepção (ver Figura 5.5). Nos experimentos com pelotões compostos por 2 veículos, o *RCBP* apresentou um melhor desempenho que o *VCBP* em termos da latência de entrega mensagens: obteve latências de entrega inferior a 100ms em 92% das entregas e, em 8% das entregas, obteve latências de entrega entre 1000 e 2500ms; enquanto que *VCBP* obteve latências entre 1000 e 2500ms em 100% das entregas. Na medida em que o tamanho do pelotão cresce o desempenho do *VCBP* se torna melhor que o do *RCBP* em termos de latência de entrega: para um pelotão com 4 veículos, o *RCBP* obteve latências entre 1000 e 2500ms em 84% das entregas e latências entre 2500 e 5000ms em 16% das entregas; enquanto que o *VCBP* obteve latências entre 1000 e 2500ms em 91,8% das entregas e latências entre 2500 e 5000ms em 8,2% das entregas. No caso dos experimentos com pelotões compostos por 8 veículos, o *RCBP* apresentou latências superiores a 5000ms em 54,9% das entregas, possuindo ainda uma variabilidade muito maior em suas latências de entrega. No *VCBP*, 100% das mensagens são entregues com latência igual ou inferior a 5000ms. Esse melhor desempenho do *VCBP* para pelotões maiores é obtido por conta da retransmissão apenas para veículos vizinhos dentro de um raio de alcance limitado, combinado com um prazo para expiração de blocos causais, pois quando prazo expira, as retransmissões cessam. No *RCBP* as mensagens são retransmitidas por qualquer membro do pelotão e não há prazo para que

as retransmissões cessem.

O *M-RBP* apresenta baixas latências de recepção e de entrega de mensagens, mas possui um baixo desempenho em termos de taxas de recebimento e de entrega de mensagens quando comparado com o *RCBP* e com o *VCBP* – apesar de também possuir uma abordagem para confirmação de recebimento e retransmissão de mensagens perdidas. Nos cenários com interferência na comunicação veicular e pelotões com apenas 2 veículos, o *M-RBP* consegue realizar o recebimento e entrega de 100% das mensagens, com latência de recepção inferior a $100ms$ e com latência de entrega entre 400 e $500ms$. Na medida em que o tamanho do pelotão aumenta, as taxas de recebimento e de entrega diminuem: em um pelotão com 4 veículos, 100% de mensagens são recebidas com latência inferior a $100ms$ e apenas 85% de mensagens entregues com latência entre 200 e $300ms$; e, em um pelotão com 8 veículos, 88,9% de mensagens recebidas com latência inferior a $100ms$, 7% de mensagens recebidas com latência entre 100 e $200ms$, mas apenas 29% das mensagens foram entregues, com latência entre 100 e $200ms$. Esse desempenho do *M-RBP* se deve ao fato de que o aumento do tamanho do pelotão implica em uma redução do intervalo de tempo da rodada *TDMA*, o que compromete as etapas de acordo de recebimento e de acordo de entrega de mensagens.

Por fim, todas as mensagens do Δ -*CBP* são recebidas e entregues com latência inferior a $100ms$, contudo ele apresenta o mais baixo desempenho em termos das taxas de recepção e de entrega de mensagens: 90% de mensagens recebidas e 25,5% de mensagens entregues em um pelotão com 2 veículos; 90% de mensagens recebidas e 2,8% de mensagens entregues em um pelotão com 4 veículos; e 90,4% de mensagens recebidas e 6,0% de mensagens entregues para um pelotão com 8 veículos. Conforme explicado no Capítulo 3, o Δ -*CBP* não possui mecanismos eficientes de detecção e retransmissão de mensagens perdidas. Por outro lado, quando um veículo difunde uma mensagem para o grupo, os demais veículos que a receberam fazem uma retransmissão da mesma para os seus vizinhos. Essa inundação permite que algumas mensagens perdidas sejam recuperadas – apenas não têm chance de serem recuperadas aquelas mensagens que foram difundidas, mas que não foram recebidas por nenhum outro membro do grupo. Como a aplicação de pelotão executa envio período de *beacons*, o mecanismo de *backoff* nativo do *IEEE 802.11p* evita certas colisões entre as difusões de *beacons*.

Neste capítulo são apresentadas as considerações finais e propostas de trabalhos futuros.

CONSIDERAÇÕES FINAIS

Neste trabalho, apresentamos o *VCBP*, um protocolo de comunicação em grupo para suportar operações consistentes em aplicações distribuídas de coordenação veicular baseadas em redes veiculares *ad hoc*. O protocolo proposto oferece confiabilidade, com ordem causal e restrições temporais na entrega das mensagens, requisitos compatíveis com aqueles demandados pelas aplicações distribuídas confiáveis de coordenação veicular. A abordagem proposta fornece um conjunto mais extenso de garantias de entrega de mensagens do que outras abordagens de comunicação em grupo projetadas para aplicações de coordenação veicular distribuída em VANETs.

Mesmo não havendo um protocolo com o mesmo conjunto de facilidades de entrega, o desempenho da abordagem proposta foi comparada com protocolos de comunicação em grupo com facilidades parciais similares existentes na literatura. Um conjunto de experimentos foram realizados em ambiente simulado, considerando grupos de veículos de diferentes tamanhos em diferentes condições de interferência na comunicação veicular.

Os resultados obtidos mostram que o protocolo de comunicação em grupo proposto neste trabalho, apesar de necessitar de mais rodadas de difusão para garantir as condições para a entrega das mensagens, possui uma eficiência maior em termos da taxa de recebimento e de entrega de mensagens nos cenários de experimentação considerados. Além disso, ele oferece suporte a um conjunto mais amplo de requisitos para aplicativos de coordenação veicular em VANET.

Apesar dos resultados, existem vários possíveis trabalhos futuros a serem realizados. A escalabilidade da abordagem proposta ainda precisa ser avaliada, considerando grupos com um número de veículos mais significativo. Atualmente, o tamanho dos grupos tolerados pelo *VCBP* é limitado a 18 veículos, por conta dos metadados que são difundidos nas mensagens e do tamanho máximo previsto para cada mensagem (*frame*) no padrão de comunicação veicular IEEE 802.11p. Além disso, mecanismos de associação (*membership*) ao grupo precisam ser adequadamente avaliados em diferentes cenários e mudanças variadas na composição do grupo. A avaliação de desempenho também precisa considerar um maior número de cenários de coordenação veicular, tais como rotatórias, interseções,

mudanças de faixa, etc. Um outro trabalho futuro é avaliar propostas para realizar adaptação do raio de retransmissão das condições do contexto de comunicação veicular, observando aspectos com densidade de veículos, níveis de interferência na comunicação, variações na mobilidade dos veículos etc. Segurança nas interações entre veículos, falhas bizantinas ou falhas simétricas – três problemas críticos em alguns cenários VANET – precisam ser devidamente avaliados. Também é trabalho futuro verificar se a coordenação veicular pode ser melhorada quando abordagens de co-design são usadas para desenvolver o controle veicular cooperativo combinado com os protocolos de comunicação em grupo. Por fim, é necessário avaliar a adequação do protocolo proposto em outros tipos de redes veiculares, tais como redes veiculares aéreas, aquáticas, subaquáticas etc.

REFERÊNCIAS BIBLIOGRÁFICAS

- ADNORMAS. A Qualidade dos Sistemas Inteligentes de Transporte (ITS). Ad-Normas, 2022. Disponível em: <https://www.revistaadnormas.com.br/2022/04/26/a-qualidade-dos-sistemas-inteligentes-de-transporte-its>.
- AHMAD, N. et al. Reviews on various inertial measurement unit (imu) sensor applications. *International Journal of Signal Processing Systems*, EJournal Publishing, v. 1, n. 2, p. 256–262, 2013.
- ALVES, R. S. et al. Redes Veiculares: principios, aplicações e desafios. *Minicursos do Simpósio Brasileiro de Redes de Computadores, SBRC*, p. 17–24, 2009.
- ANWER, M. S.; GUY, C. A Survey of VANET Technologies. *Journal of Emerging Trends in Computing and Information Sciences*, Citeseer, v. 5, n. 9, p. 661–671, 2014.
- ASPLUND, M.; LÖVHALL, J.; VILLANI, E. Specification, Implementation and Verification of Dynamic Group Membership for Vehicle Coordination. In: IEEE. *Dependable Computing (PRDC), 2017 IEEE 22nd Pacific Rim International Symposium on*. Christchurch, New Zealand, 2017. p. 321–328.
- BALDONI, R. et al. Efficient Δ -causal boadcasting. *COMPUT SYST SCI ENG*, v. 13, n. 5, p. 263–269, 1998.
- BASTOS, J. T. et al. *Desempenho Brasileiro na Década da Ação pela Segurança do Trânsito: Análise, Persperctiva e Indicadores de 2011-2020*. Brasília, Brasil, 2020. 120 p. Disponível em: <https://www.onsv.org.br/estudos-pesquisas/estudo-desempenho-brasileiro-na-decada-de-acao-pela-seguranca-no-transito>.
- BEHRISCH, M. et al. Sumo—simulation of urban mobility: an overview. In: *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. Barcelona: ThinkMind, 2011. p. 23–28. Disponível em: <https://elib.dlr.de/71460/>.
- BIEKER-WALZ, L. et al. Evaluation of car-following-models at controlled intersections. In: EUROSIS. Lissabon, Portugal, 2017.
- BIRMAN, K. P.; JOSEPH, T. A. Reliable communication in the presence of failures. *ACM Transactions on Computer Systems (TOCS)*, ACM New York, NY, USA, v. 5, n. 1, p. 47–76, 1987.
- BOUKERCHE, A. et al. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer communications*, Elsevier, v. 31, n. 12, p. 2838–2849, 2008.

BUCHENSCHWEIT, A. et al. A VANET-based emergency vehicle warning system. *2009 IEEE Vehicular Networking Conference, VNC 2009*, p. 1–8, 2009.

CAMPOLO, C. et al. Modeling broadcasting in IEEE 802.11p/WAVE vehicular networks. *IEEE Communications letters*, IEEE, v. 15, n. 2, p. 199–201, 2010.

CHEN, Q.; JIANG, D.; DELGROSSI, L. IEEE 1609.4 DSRC multi-channel operations and its implications on vehicle safety communications. p. 1–8, 2009.

CHOCKLER, G. V.; KEIDAR, I.; VITENBERG, R. Group communication specifications: a comprehensive study. *ACM Computing Surveys (CSUR)*, ACM, v. 33, n. 4, p. 427–469, 2001.

COMMITTEE, I. C. S. L. S. et al. Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11[^]*, 2007.

COULOURIS, G. et al. *Sistemas Distribuídos-: Conceitos e Projeto*. Porto Alegre, Brasil: Bookman Editora, 2013.

CRISTIAN, F. Understanding fault-tolerant distributed systems. *Communications of the ACM*, ACM, New York, NY, USA, v. 34, n. 2, p. 56–78, February 1991. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/102792.102801>. Acesso em: October, 4, 2021.

CUNHA, F. et al. Sistemas de transporte inteligentes: Conceitos, aplicações desafios. *Livro de Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'17)*, 2017.

DIMITRAKOPOULOS, G. J.; UDEN, L.; VARLAMIS, I. *The future of intelligent transport systems*. Amsterdã, Holanda: Elsevier, 2020.

EVROPEYTSEV, G. et al. A scalable indirect position-based causal diffusion protocol for vehicular networks. *IEEE Access*, IEEE, v. 7, p. 14767–14778, 2019.

GOLUBEVA, A.; MAGARIL, E. Improved economic stimulation mechanism to reduce vehicle CO₂ EMISSIONS. *WIT Transactions on The Built Environment*, WIT Press, v. 130, 2013. ISSN 1743-3509.

GU, Y.; LIN, S.-p.; MAXEMCHUK, N. F. A fail safe broadcast protocol for collaborative intelligent vehicles. *Proceedings of the WoWMoM 2015: A World of Wireless Mobile and Multimedia Networks*, p. 1–6, 2015.

GUIDEC, F.; LAUNAY, P.; MAHÉO, Y. Causal and Δ -causal broadcast in opportunistic networks. *Future Generation Computer Systems*, Elsevier, v. 118, p. 142–156, 2021.

GUO, M.; AMMAR, M. H.; ZEGURA, E. W. V3: A vehicle-to-vehicle live video streaming architecture. *Pervasive and Mobile Computing*, Elsevier, v. 1, n. 4, p. 404–424, 2005.

HADZILACOS, V.; TOUEG, S. A modular approach to fault-tolerant broadcasts and related problems. Cornell University, Ithaca, NY, 1994. Disponível em: <https://ecommons.cornell.edu/bitstream/handle/1813/6207/94-1425.pdf>.

HAIDER, S. et al. DABFS: A robust routing protocol for warning messages dissemination in vanets. *Computer Communications*, Elsevier, v. 147, p. 21–34, 2019.

HARTENSTEIN, H.; LABERTEAUX, K. *VANET: vehicular applications and inter-networking technologies*. John Wiley & Sons, 2009. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470740637.fmatter>.

HULT, R. et al. Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation. *IEEE Signal Processing Magazine*, IEEE, v. 33, n. 6, p. 74–84, 2016.

JIA, D. et al. A survey on platoon-based vehicular cyber-physical systems. *IEEE communications surveys & tutorials*, IEEE, v. 18, n. 1, p. 263–284, 2015.

JIANG, D.; CHEN, Q.; DELGROSSI, L. Optimal data rate selection for vehicle safety communications. In: *Proceedings of the Fifth ACM International Workshop on Vehicular Inter-NETworking*. New York, NY, USA: Association for Computing Machinery, 2008. (VANET '08), p. 30–38. ISBN 9781605581910. Disponível em: <https://doi.org/10.1145/1410043.1410050>.

JIANG, D.; DELGROSSI, L. IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments. p. 2036–2040, 2008.

KRAUSS, S. Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics. *Forschungsbericht - Deutsche Forschungsanstalt fuer Luft - und Raumfahrt e. V.*, 1998. ISSN 14348454.

KUROSE, J. F.; ROSS, K. W. *Redes de Computadores e a Internet: uma abordagem top-down*. São Paulo: Pearson, 2013.

LAMPORT, L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, ACM, v. 21, n. 7, p. 558–565, 1978.

LEE, K. C. et al. First experience with cartorrent in a real vehicular ad hoc network testbed. In: IEEE. *2007 Mobile Networking for Vehicular Environments*. USA, 2007. p. 109–114.

LIMA, G.; MACÊDO, R. A. A robust causal order protocol. *11th Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'99)*, v. 11, 1999.

MA, X.; CHEN, X. Performance analysis of IEEE 802.11 broadcast scheme in ad hoc wireless lans. *IEEE Transactions on Vehicular Technology*, v. 57, n. 6, p. 3757–3768, 2008.

MACÊDO, R. A.; EZHILCHLVAN, P.; SHRIVASTAVA, S. K. Newtop: A total order multicast protocol using causal blocks. *BROADCAST project deliverable report*, v. 1, 1993.

MACÊDO, R. J. d. A. *Fault-tolerant group communication protocols for asynchronous systems*. Tese (Doutorado) — Newcastle University, 1994.

MACÊDO, R. J. d. A.; FREITAS, A. E. S.; SÁ, A. S. de. Enhancing group communication with self-manageable behavior. *Journal of Parallel and Distributed Computing*, v. 73, n. 4, p. 420–433, 2013. ISSN 0743-7315. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0743731512002869>.

MACÊDO, R. J. de A.; FREITAS, A. E. S. A generic group communication approach for hybrid distributed systems. In: SENIVONGSE, T.; OLIVEIRA, R. (Ed.). *IFIP International Conference on Distributed Applications and Interoperable Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 102–115. ISBN 978-3-642-02164-0.

MAXEMCHUK, N. F. Safety issues in collaborative vehicle control. In: IEEE. *Sarnoff Symposium, 2009. SARNOFF'09. IEEE*. Princeton, NJ, USA, 2009. p. 1–5.

NADEEM, T. et al. Trafficview: traffic data dissemination using car-to-car communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM, v. 8, n. 3, p. 6–19, 2004.

NANDAN, A. et al. Co-operative downloading in vehicular ad-hoc wireless networks. In: IEEE. *Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on*. IEEE, 2005. p. 32–41.

OLIVEIRA, R. R. et al. *Co-design de abordagem baseada em consenso e protocolo de comunicação confiável para o controle de pelotões de veículos*. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2020. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/216313/PEAS0348-T.pdf?sequence=-1&isAllowed=y>.

ONU. Brasil tem cerca de 30 mil mortes por acidentes de trânsito ao ano. ONU, 2021. Disponível em: <https://news.un.org/pt/story/2021/05/1750922>.

PELUSI, L.; PASSARELLA, A.; CONTI, M. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine*, v. 44, n. 11, p. 134–141, 2006.

PEREIRA, R. H. M. et al. Tendências e desigualdades da mobilidade urbana no Brasil I : o uso do transporte coletivo e individual. *Texto para Discussão IPEA*, Instituto de Pesquisa Econômica Aplicada (Ipea), v. 2673, 2021. ISSN 1415-4765. Disponível em: <http://repositorio.ipea.gov.br/handle/11058/10713>.

PRAKASH, R.; RAYNAL, M.; SINGHAL, M. An efficient causal ordering algorithm for mobile computing environments. *IEEE*, p. 744–751, 1996. Disponível em: [⟨https://ieeexplore.ieee.org/document/508027⟩](https://ieeexplore.ieee.org/document/508027).

QURESHI, K. N.; 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.

ROBINSON, T.; CHAN, E.; COELINGH, E. Operating platoons on public motorways: An introduction to the sartre platooning programme. Researchgate, 2010.

RUBINSTEIN, M. G. et al. A survey on wireless ad hoc networks. In: PUJOLLE, G. (Ed.). *Mobile and Wireless Communication Networks*. Boston, MA: Springer US, 2006. p. 1–33. ISBN 978-0-387-34736-3.

RUIZ, P.; BOUVRY, P. Survey on broadcast algorithms for mobile ad hoc networks. *ACM Computing Surveys*, Association for Computing Machinery, New York, NY, USA, v. 48, n. 1, jul 2015. ISSN 0360-0300. Disponível em: [⟨https://doi.org/10.1145/2786005⟩](https://doi.org/10.1145/2786005).

SANTOS, A. S. d.; LEAL, A. G. *Sistemas inteligentes de transporte: um panorama das tendências e caminhos de pesquisa*. São Paulo: IPT, 2015. Disponível em: [⟨http://www.ipt.br/noticia/871-transporte_inteligente.htm⟩](http://www.ipt.br/noticia/871-transporte_inteligente.htm).

SCHIPER, A.; SANDOZ, A. Uniform reliable multicast in a virtually synchronous environment. In: *IEEE. [1993] Proceedings of the 13th International Conference on Distributed Computing Systems*. Pittsburgh, USA: IEEE, 1993. p. 561–568. Disponível em: [⟨https://doi.org/10.1109/ICDCS.1993.287667⟩](https://doi.org/10.1109/ICDCS.1993.287667).

SLOT, M.; CAHILL, V. A reliable membership service for vehicular safety applications. In: *IEEE. Intelligent Vehicles Symposium (IV), 2011 IEEE*. [S.l.], 2011. p. 1163–1169.

SLOT, M. C. F. *Group Communication for Cooperative Automated Vehicles*. Tese (Doutorado) — Trinity College Dublin, 2014.

SOMMER, C. et al. Veins: The open source vehicular network simulation framework. In: _____. *Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem*. Cham: Springer International Publishing, 2019. p. 215–252. ISBN 978-3-030-12842-5. Disponível em: [⟨https://doi.org/10.1007/978-3-030-12842-5_6⟩](https://doi.org/10.1007/978-3-030-12842-5_6).

SOMMER, C.; GERMAN, R.; DRESSLER, F. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on mobile computing*, IEEE, v. 10, n. 1, p. 3–15, 2010.

TANENBAUM, A. S.; STEEN, M. V. *Sistemas distribuídos: princípios e paradigmas*. 2. ed. São Paulo: Pearson Prentice Hall, 2008.

TANENBAUM, A. S.; STEEN, M. V. *Distributed Systems*. 4. ed. Boston: Pearson Education, 2023. ISBN 978-90-815406-4-3.

TANENBAUM, A. S.; WETHERALL, D. *Computer Networks*. 5. ed. Boston: Prentice Hall, 2011. ISBN 978-0-13-212695-3.

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*. Brussels, Belgium: ICST, 2010.

WANG, Y. et al. Throughput and delay limits of 802.11p and its influence on highway capacity. *Procedia - Social and Behavioral Sciences*, v. 96, p. 2096–2104, 2013. ISSN 1877-0428. Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th COTA International Conference of Transportation Professionals (CICTP2013). Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877042813023628>.

WILLKE, T. L.; MAXEMCHUK, N. F. Reliable collaborative decision making in mobile ad hoc networks. In: SPRINGER. *Management of Multimedia Networks and Services: 7th IFIP/IEEE International Conference, MMNS 2004, San Diego, CA, USA, October 3-6, 2004. Proceedings 7*. [S.l.], 2004. p. 88–101.

WILLKE, T. L.; MAXEMCHUK, N. F. Coordinated interaction using reliable broadcast in mobile wireless networks. In: SPRINGER. *International Conference on Research in Networking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 1168–1179. ISBN 978-3-540-32017-3.

WILLKE, T. L.; MAXEMCHUK, N. F. Coordinated interaction using reliable broadcast in mobile wireless networks. *Computer Networks*, Elsevier, v. 51, n. 4, p. 1052–1059, 2007.

WILLKE, T. L.; TIENRAKOOL, P.; MAXEMCHUK, N. F. A survey of inter-vehicle communication protocols and their applications. *IEEE Communications Surveys & Tutorials*, IEEE, v. 11, n. 2, p. 3–20, 2009.

ZHOU, J.; ZHU, F. Analytical analysis of the effect of maximum platoon size of connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, v. 122, p. 102882, 2021. ISSN 0968-090X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0968090X20307828>.

ACOPLAMENTO DOS SIMULADORES

A Figura A.1 ilustra o acoplamento provido pelo VEINS que permite integrar o OMNeT++ para simulação do ambiente de comunicação e o SUMO para a simulação do tráfego dos veículos nas vias. Desta forma, os movimentos de veículos no SUMO são refletidos em movimentos dos nós em uma simulação de rede no OMNET++. Os nós podem interagir com a simulação do SUMO, por exemplo, para verificar a influência da comunicação entre veículos no tráfego rodoviário. O VEINS é responsável por gerenciar a disseminação de dados, permitindo a comunicação entre as aplicações de ITS e maior segurança e conforto no trânsito. A troca de informações entre os dois simuladores é interfaciada pelo TraCI (Traffic Control Interface).

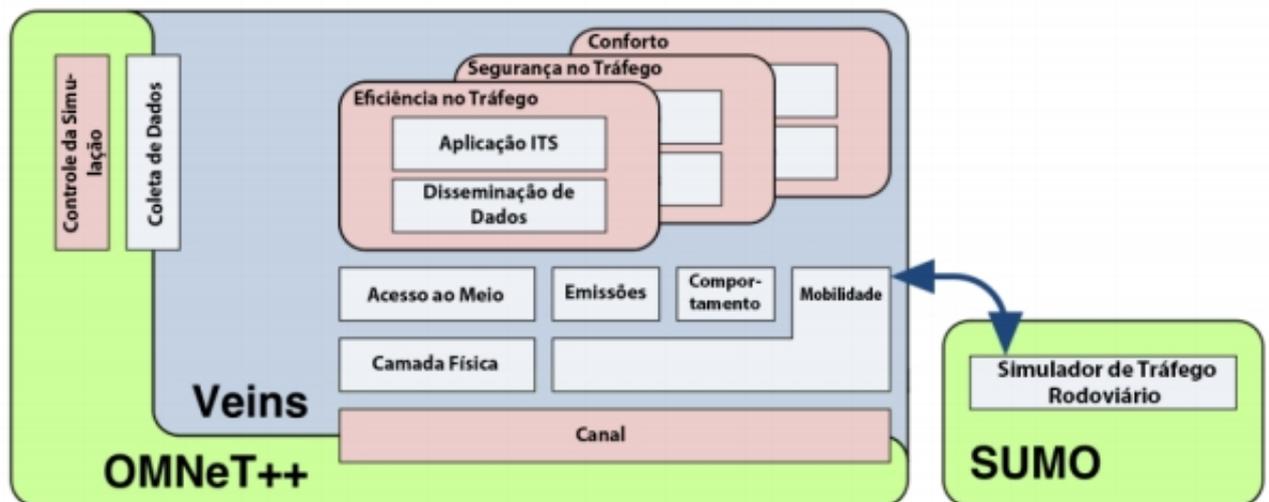


Figura A.1: Visão geral da estrutura de simulação acoplada do VEINS Adaptado de (SOMMER; GERMAN; DRESSLER, 2010)

DIAGRAMAS DE CLASSE

B.1 RCBP

A Figura B.1 ilustra o Diagrama de Classes construído para o *RCBP*.

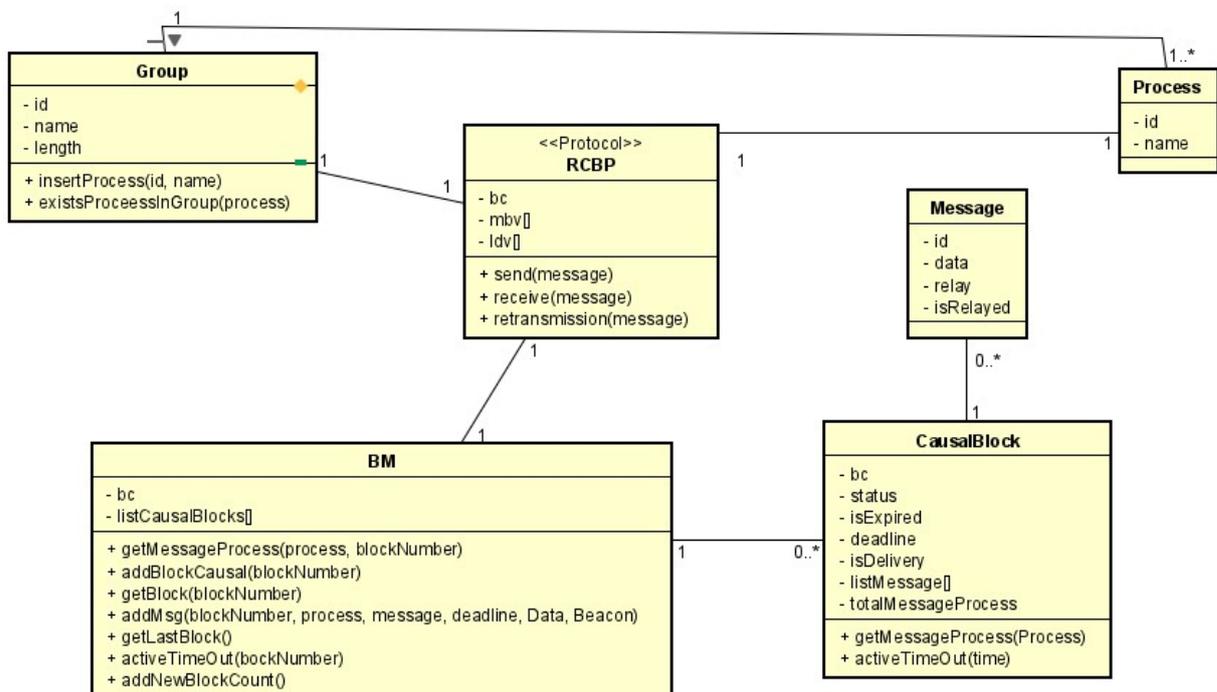


Figura B.1: Diagrama de Classe do RCBP

B.2 VCBP

A Figura B.2 ilustra o Diagrama de Classes do *VCBP*. O *VCBP* herda as características do *RCBP* que possui uma estrutura de armazenamento de blocos causais (BM, Matriz de Blocos) e um processo. Cada bloco está relacionada a um conjunto de mensagens. O *RCBP* possui o conhecimento do grupo de processos. Para melhorar a visualização, estas particularidades do *RCBP* foram omitidas na Figura B.2 e detalhadas no Diagrama da Figura B.1 da Seção B.1

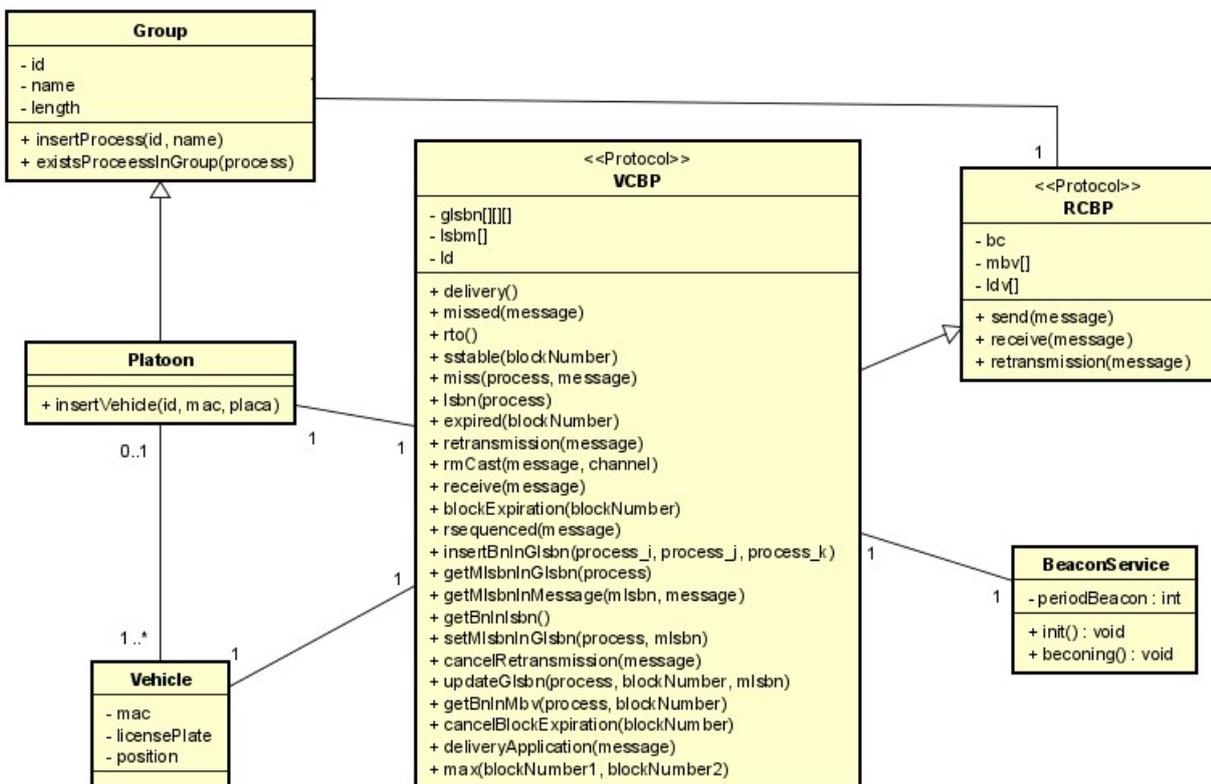


Figura B.2: Diagrama de Classe do VCBP

CÓDIGOS-FONTE DAS PRINCIPAIS TAREFAS E OPERAÇÕES DO *VCBP* IMPLEMENTADAS NO VEINS

C.1 FUNÇÃO INIT

```
1 void VCBP::init()
2 { /*
3     *0 numero de carros deve ser inserido na variavel totalProcesso
4     *e no arquivo erlangen.rou.xml, parametro number.
5     */
6     this->totalProcesso = getSimulationManager()->countCar; //Foram
7     efetuados testes com 2, 4, e 8 veiculos
8     timeSimulation =getSimulationManager()->periodSimulation; //100
9     insertFailsByInterference = true; //foram efetuados testes com
10    inducao de Falhas e sem inducao;
11    this->countMsgWithfailure = 10; //porcentagem
12    this->sendMsgApp = false;
13    this->sendBeacon = true;
14    this->contSendBeacon= 0;
15    //0 platoon contem managerMembership(this->totalProcesso, true) ;
16    this->platoon = new Platoon(this->idCar,this->totalProcesso);
17
18    //hash obtido a partir do MAC do Veiculo.
19    managerHashVehicleInPlatoon(getSimulationManager()->ListCar, this
20    ->platoon);
21    this->group =dynamic_cast<Group*>(this->platoon);
22
23    this->bc = 0;
24    this->ld=0;
25    this->createMbv(this->idCar ,this->totalProcesso);
26    this->createGlsbn(this->idCar ,this->totalProcesso);
```

```

26     this->BM = new BM(this->totalProcesso);
27
28     Beaconing();
29
30
31 }

```

C.2 FUNÇÃO BEACONING

```

1
2 void VCBP::beaconing()
3 {
4     /* MessageBeaconCausalBlock herda as propriedades da classe
5         BasicSafetyMessage que eh uma mensagem curta do protocolo
6         Wave */
7     MessageBeaconCausalBlock * msgBeacon = new
8         MessageBeaconCausalBlock();
9     msgBeacon->setKind("BEACON");
10
11     /* O Veins utiliza este metodo para acrescentar no beacon os
12         valores de posicao, velocidade, entre outros */
13     populateWSM(msgBeacon);
14
15     rmcast(msgBeacon, CCH);
16
17     /*Cada mensagem agendada considerou o valor do rto() no delay
18         double delay = getSimulationManager()->periodMSGSend;
19
20     /*O gerenciador de agendamentos ira verificar a mensagem agendada
21         quando o timeout expirar. Como o tipo he de uma mensagem de
22         agendamento de Beacon, deve-se chamar o metodo beaconing
23         novamente ate o periodo de simulacao maximo expirar.
24         Definimos este periodo no arquivo de simulacao
25         da aplicacao do VEINS */
26     MessageBeaconCausalBlock * scheduleMsgBeacon = new
27         MessageBeaconCausalBlock();
28     scheduleMsgBeacon->setKind("SCHEDULEBEACON");
29     scheduleAt(delay, scheduleMsgBeacon);
30
31     /*no Veins o agendador invoca o evento handleSelfMsg quando o
32         timeut expira.
33 }

```

C.3 FUNÇÃO HANDLESELFMSG

```

1 //Funcao acionada pelo agendador de eventos/tarefas quando o tempo
2     expira
3 void VCBP::handleSelfMsg(cMessage* msg) {
4     double delay = 0;

```

```

4     double delay2=0;
5     if (msg->getKind()=="SCHEDULEBEACON" && this->sendBeaco)
6     {
7         //Envia as mensagens agendadas a partir do beaconing()
8
9         beaconing();
10        this->contSendBeacon++;
11    }
12    else if(msg->getKind()=="SCHEDULEMSG")
13    {
14        retransmission(msg);
15    }
16    else if( msg->getKind()=="SCHEDULE_TIMEOUT_BLOCK")
17    {
18        BlockExpiration(msg->getBn());
19    }
20 }

```

C.4 FUNÇÃO RMCAST

```

1 void VCBP:: rmcast(cMessage* msg, Channel ch)
2 {
3     int idNewBC = 0;
4     int bn;
5     String src;
6     double d;
7
8
9     if (msg !=NULL)
10    { scr = idCar;
11      d = deadline;
12      idNewBC =this->BM->incrementaBlockCount();
13      idNewBC = this->BM->getBlockCount();
14      msg->setBn(idNewBC);
15      msg->setId(src);
16      msg->setDeadLine(d);
17
18
19      BM->addMsg(msg, idNewBC, src);
20      this->BM->updateMBVi(this->src, idNewBC);
21      if !(blockExpirationSchedule(idNewBC))
22      { activeTimeOutBlock(idNewBC, msg->getDeadLine);}
23      this->BM->insertBnInGlsbn(src, src, src, bn);
24      msg->setMlsbn(getMlsbnInGlsbn(src));
25
26      /* O metodo sendDown do Veins ja verifica automaticamente o tipo
          de mensagem para decidir qual canal utilizar. Se for
          beacon (que pode ser de seguranca ou de controle), o canal
          sera CCH. Configuramos o VEINS para utilizar somente o canal
          CCH, mas por precaucão, criamos uma rotina para activar o
          canal CCH, caso esteja inativo. Ressaltamos que no WAVE ha o
          periodo de sincronizacao */

```

```

27     activateChannel(ch);
28
29     // realiza a disseminacao na rede
30     veicular
31     sendDown(msgBeacon);
32
33
34
35 }
36
37 }

```

C.5 FUNÇÃO RECEIVE

```

1 void VCBP::receive(cMessage* msg) {
2
3     MessageBeaconCausalBlock* msgBeacon = dynamic_cast<
4         MessageBeaconCausalBlock*>(msg);
5     int mlsbn [][];
6     int bnAux;
7     double rt;
8
9     if ( platoon->existsProceessInGroup(msgBeacon->src) && !expired(
10         msgBeacon->bn))
11     {
12         if (!existsMsgInBM(msgBeacon))
13         {
14             mlsbn = getMlsbnInMessage(pMsgBeacon);
15             addMsgInBM(msgBeacon->getBn(),
16                 msgBeacon->getSrc(),
17                 mlsbn,
18                 msgBeacon->getId(),
19                 msgBeacon->getDeadline(),
20                 msgBeacon->getData(), msgBeacon);
21             if (!scheduled(msgBeacon->getBn()))
22             {
23                 msgTimeOut = new MessageBlock(nome);
24                 msgTimeOut->setKind("SCHEDULE_TIMEOUT_BLOCK");
25                 msgTimeOut->setIdBC(msgBeacon->getBn());
26                 msgTimeOut->setName("SCHEDULE_TIMEOUT_BLOCK");
27                 msgTimeOut->setTimeSender(msgBeacon->getDeadline());
28                 scheduleAt(GPS->getTime() +msgBeacon->getDeadline(),
29                     msgTimeOut);
30             }
31             if (getBnInMbv(msgBeacon->getSrc() < msgBeacon->getBn()))
32             {
33                 getBnInMbv(msgBeacon->getSrc(), msgBeacon->getBn());
34                 setMlsbnInGlsbn(msgBeacon->getSrc(), mlsbn);
35             }

```

```

36         bnAux = getBnInlsbn(msgBeacon->getSrc());
37
38         //incluir a atualizacao do glsbn
39         updateGlsbn(msgBeacon->getSrc(), bnAux, mlsbn));
40
41         for (int countBC = 0; countBC < this->bc; countBC++)
42         {
43             for (int contMsgProcess = 0; contMsgProcess < group->
44                 Length; contMsgProcess++)
45             { Message* msg = BM->getMessageProcess(contMsgProcess,
46                 countBC);
47                 if (missed(msg))
48                 { msg->relay = true;
49                     rt = GPS->getTime() + rto();
50                     msg->setKind("SCHEDULEMSG");
51                     scheduleAt(rt, msg);
52                 }
53             }
54         }
55         delivery();
56
57     }
58     else
59     {
60         rt = GPS->getTime() + rto();
61         scheduleAt(rt, msgBeacon);
62     }
63 }
64 }

```

C.6 FUNÇÃO RETRANSMISSION

```

1 void VCBP::retransmission(cMessage * msg)
2 {
3
4     if (!expired(msg->bn))
5     {
6         for(int contProcess=0; contProcess < group->Length; contProcess
7             ++))
8         {
9             if (miss(contProcess, msg) && distance(idCar, contProcess) <=
10                 raioRetransmission)
11             {
12                 MessageBeaconCausalBlock * pacoteMsgBeacon =
13                     dynamic_cast<MessageBeaconCausalBlock*>(msg);
14                 this->populateWSM(pacoteMsgBeacon);
15
16                 //Disseminacao da Rede veicular
17                 sendDown(msgBeacon);
18             }
19         }
20     }
21 }

```

```
16         msgBeacon->relay =false;
17         msgBeacon->isRelayed= true;
18     }
19 }
20 }
21 }
```

C.7 FUNÇÃO DELIVERY

```
1
2 void VCBP:: delivery()
3 {   CausalBlock* causalBlock;
4     while (sstable(ld+1))
5     {   //ld armazena o bn do ultimo bloco entregue
6         ld = ld + 1;
7         cancelBlockExpiration(ld);
8         causalBlock = BM->getBlock(ld);
9         for (int countMsgProcess = 0; countMsgProcess < group->length
10             ; countMsgProcess++)
11         {   MessageBeaconCausalBlock* msg = causalBlock->
12             getMessageProcess(countMsgProcess);
13             bc = max(bc, causalBlock->getMessageProcess(
14                 countMsgProcess));
15             if (msg != NULL)
16             {
17                 deliveryApplication(msg);
18             }
19         }
20     }
21 }
```

Salvador, 07/12/2023

DECLARAÇÃO

Declaro para os devidos fins que o texto final apresentado para a conclusão do meu curso de Mestrado em Mecatrônica da Universidade Federal da Bahia é de minha autoria. Declaro também que quaisquer informações utilizadas neste texto, mas que sejam provenientes de outros trabalhos, têm fonte claramente expressa e, quando for o caso, o seu uso foi devidamente autorizado pelo(s) respectivo(s) autor(es).

 Documento assinado digitalmente
MARCELA ALVES PEREIRA
Data: 07/12/2023 08:31:25-0300
Verifique em <https://validar.itf.gov.br>

Nome: Marcela Alves Pereira
CPF: 012.323.485-97