



Universidade Estadual de Campinas
Instituto de Computação



Allan Mariano de Souza

Controle de Congestionamentos de Veículos utilizando Sistemas de Transporte Inteligentes

CAMPINAS
2016

Allan Mariano de Souza

**Controle de Congestionamentos de Veículos utilizando Sistemas
de Transporte Inteligentes**

Dissertação apresentada ao Instituto de
Computação da Universidade Estadual de
Campinas como parte dos requisitos para a
obtenção do título de Mestre em Ciência da
Computação.

Orientador: Prof. Dr. Leandro Aparecido Villas

Este exemplar corresponde à versão final da
Dissertação defendida por Allan Mariano de
Souza e orientada pelo Prof. Dr. Leandro
Aparecido Villas.

CAMPINAS
2016

Agência(s) de fomento e nº(s) de processo(s): CNPq, 132901/2015-3

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

So89c Souza, Allan Mariano de, 1992-
Controle de congestionamentos de veículos utilizando sistemas de transporte inteligentes / Allan Mariano de Souza. – Campinas, SP : [s.n.], 2016.

Orientador: Leandro Aparecido Villas.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Sistemas inteligentes de veículos rodoviários. 2. Sistemas de comunicação móvel. 3. Sistemas inteligentes de controle. 4. Trânsito - Congestionamento. I. Villas, Leandro Aparecido, 1983-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Traffic jam control through intelligent transportation systems

Palavras-chave em inglês:

Intelligent transportation systems

Mobile communication systems

Intelligent control systems

Traffic congestion

Área de concentração: Ciência da Computação

Titulação: Mestre em Ciência da Computação

Banca examinadora:

Leandro Aparecido Villas [Orientador]

Eduardo Coelho Cerqueira

Edmundo Roberto Mauro Madeira

Data de defesa: 03-03-2016

Programa de Pós-Graduação: Ciência da Computação



Universidade Estadual de Campinas
Instituto de Computação



Allan Mariano de Souza

Controle de Congestionamentos de Veículos utilizando Sistemas de Transporte Inteligentes

Banca Examinadora:

- Prof. Dr. Leandro Aparecido Villas
Universidade Estadual de Campinas
- Prof. Dr. Eduardo Coelho Cerqueira
Universidade Federal do Pará
- Prof. Dr. Edmundo Roberto Mauro Madeira
Universidade Estadual de Campinas

A ata da defesa com as respectivas assinaturas dos membros da banca encontra-se no processo de vida acadêmica do aluno.

Campinas, 03 de março de 2016

Resumo

O congestionamento é um problema grave nas grandes cidades, o qual acaba gerando perdas econômicas, aumento do tempo de viagem e maior emissão de gases poluentes. Uma das principais causas do congestionamento é o aumento súbito do tráfego de veículos durante os horários de pico, acidentes de trânsito e gargalos nas infraestruturas de transporte. Nos últimos anos, com os avanços nas tecnologias de comunicação sem fio e o desenvolvimento de padrões para rede veiculares, tornou-se possível a implementação de Sistemas de Transporte Inteligentes. Os sistemas de transporte inteligentes são soluções abrangentes para gerenciamento de tráfego em tempo real que se baseia, sobretudo, em dados coletados de veículos, infraestruturas de rede disponíveis às margens de ruas, rodovias e de outros sensores que monitoram o trânsito. Uma área de pesquisa promissora para sistemas de transporte inteligentes são as redes veiculares, as quais são um tipo específico de redes móveis, nas quais os nós são formados por veículos com capacidade de processamento, armazenamento e comunicação sem fio. Com esta questão em mente, nessa dissertação são propostos dois sistemas de transporte inteligentes baseados em redes veiculares para detecção e controle de congestionamentos. Em particular, para controlar congestionamentos causados por acidentes de trânsito é proposto o SPARTAN, um sistema capaz de evitar a iminência de congestionamentos em locais de acidente de trânsito, o qual detecta o acidente e alerta os veículos próximos, para que eles alterem suas rotas e evitem passar pelo local do acidente, assim, evitando o surgimento de congestionamentos no local do acidente. Além disso, para controlar congestionamentos causados pelo grande número de veículos nas vias é proposto o COLOSSUS, um sistema capaz de classificar a condição de tráfego das vias, afim de detectar congestionamentos e, por fim, gerenciar o tráfego dos veículos, roteando-os para reduzir e controlar os congestionamentos detectados. Sendo assim, os resultados mostram a eficiência do SPARTAN para controlar congestionamentos causados por acidentes, uma vez que o mesmo reduz o tempo médio de viagem, tempo médio de congestionamento, consumo médio de combustível e emissões de CO em até 62%, 88%, 36% e 35% respectivamente. Por outro lado, a eficiência do COLOSSUS pode ser observada em seus resultados, onde o mesmo reduz o tempo médio de viagem, tempo médio de congestionamento, consumo médio de combustível e emissões de CO em até 34%, 70%, 12% e 11% respectivamente para congestionamentos causados pelo grande número de veículos.

Abstract

Congestion is a major problem in large cities, which ends up causing economic losses, increases the travel time and increases greenhouse gas emissions. One of the main causes of congestion is sudden increase in vehicle traffic during peak hours, traffic accidents and bottlenecks in transport infrastructure. In recent years, with advances in wireless communication technologies and the development of standards of vehicular ad hoc networks, has become possible the implementation of Intelligent Transportation Systems. Intelligent Transportation Systems are comprehensive solutions for real-time traffic management that is based primarily on data collected from vehicles, network infrastructure and other sensors that monitor the traffic. A promising research area in intelligent transportation systems are vehicular ad hoc networks, which are a specific type of mobile networks in which the nodes are formed by vehicles with processing power, storage, and wireless communication. With this question in mind, this work proposes two intelligent transport systems based on vehicle ad hoc networks for sensing and control congestions. In particular, to control congestion caused by traffic accidents is proposed the SPARTAN, which can avoid the rise of a congestion in accident areas, SPARTAN detects the accident and warns oncoming vehicles, so they change their routes and avoid to pass in the congestion area, thus avoiding the rise of a congestion in an accident area. Furthermore, to control congestion caused by the large number of vehicles on the roads is proposed COLOSSUS, a system that are able to classify the traffic condition in all roads in order to detect congestion and, finally, manage traffic of vehicles, routing them to reduce and control the detected congestion. Thus, the results of SPARTAN showed their effectiveness for controlling congestions caused by accident, since it reduces the average travel time, average congestion time, average fuel consumption and CO emissions by 62% 88% 36% and 35% respectively. On the other hand, the COLOSSUS efficiency can be seen in its results, where it reduces the average travel time, average time of congestion, average fuel consumption and CO emissions by up to 34% 70% 12% and 11% respectively for congestion caused by large number of vehicles on the roads.

Lista de Figuras

2.1	Arquiteturas de uma VANET	17
2.2	Canais disponíveis para o IEEE 802.11p	18
2.3	Arquitetura WAVE	19
2.4	Efeito de sincronização introduzido pelo padrão 802.11p	22
4.1	Arquitetura do SPARTAN	31
4.2	Fluxograma de atividades executadas pelo SPARTAN: detecção de acidentes, alerta de acidentes e gerenciamento de tráfego.	32
4.3	Zona de preferência	34
4.4	Exemplo de como o método de dessincronização funciona	35
4.5	Mapa de Manhattam.	38
4.6	Resultado algoritmo de disseminação de dados para cenário urbano	40
4.7	Avaliação das Soluções para Controle de Congestionamentos	42
5.1	Arquitetura do COLOSSUS.	46
5.2	Processo de classificação implementado pelo COLOSSUS, mostrando as entradas utilizadas, o método implementado pelo classificador, e a saída da classificação.	49
5.3	As principais entidades implementadas pelo COLOSSUS. Um veículo no interior da RA vai passar pelo congestionamento e é roteado.	50
5.4	Fluxo de operações do COLOSSUS-RC para classificação do tráfego, detecção de congestionamentos e roteamento dos veículos	53
5.5	Cálculo de uma rota para um veículo com posição atual no vértice A e com destino no vértice H	54
5.6	Eficiência do controle de congestionamento das soluções, correlação da velocidade média e densidade total do cenário a cada instante da simulação.	57
5.7	Avaliação das Soluções para Controle de Congestionamentos	58
5.8	Análise do intervalo de classificação do COLOSSUS-RC em relação à área de roteamento.	59
5.9	Análise da área de roteamento do COLOSSUS-RC em relação ao intervalo de classificação.	60

Lista de Tabelas

3.1	Uma comparação quantitativa entre as soluções de disseminação	26
3.2	Uma comparação quantitativa entre as soluções para controle e detecção de congestionamentos	29
4.1	Parâmetros de simulação SPARTAN	38
5.1	Classificação da condição de tráfego	48
5.2	Parâmetros de simulação para detecção e controle de congestionamentos . .	55

Sumário

1	Introdução	11
1.1	Motivação	12
1.2	Objetivos	13
1.3	Contribuições	14
1.4	Estrutura da Dissertação	15
2	Fundamentação Teórica	16
2.1	Redes Veiculares	16
2.1.1	Padrão IEEE 802.11p	17
2.1.2	Características	19
2.2	Desafios	20
2.2.1	Disseminação de Dados	20
2.2.2	Sincronização Introduzida pelo Padrão IEEE 802.11p	21
2.3	Cenários	22
2.4	Resumo Conclusivo	23
3	Trabalhos Relacionados	24
3.1	Disseminação de Dados	24
3.2	Deteção e Controle de Congestionamentos	26
3.3	Resumo Conclusivo	28
4	Controle de Acidentes	30
4.1	Arquitetura SPARTAN	31
4.2	Deteção de Acidentes	32
4.3	Alerta de Acidentes	33
4.3.1	Supressão Broadcast	33
4.3.2	Mecanismo de Dessincronização	35
4.4	Gerenciamento do Tráfego	36
4.5	Avaliação de desempenho	37
4.5.1	SPARTAN <i>vs</i> Soluções de disseminação	39
4.5.2	SPARTAN <i>vs</i> Soluções para Controle de Congestionamentos	41
4.6	Resumo Conclusivo	43
5	Deteção e Controle de Congestionamentos	45
5.1	Arquitetura do COLOSSUS	46
5.2	Representação da Rede e modelo de Comunicação	47
5.3	Deteção e Classificação de Congestionamentos	47
5.4	Controle de Congestionamentos	50
5.4.1	Roteamento de Melhor Caminho – COLOSSUS-RMC	51

5.4.2	Roteamento Cooperativo – COLOSSUS-RC	52
5.5	Avaliação de Desempenho	54
5.6	Eficiência das soluções de controle de congestionamentos	55
5.7	COLOSSUS <i>vs</i> DSP, RkSP e With Routing	56
5.8	Análise Exploratória do COLOSSUS	59
5.8.1	Análise do Intervalo de Classificação	59
5.8.2	Análise da Área de Roteamento	60
5.9	Resumo Conclusivo	61
6	Considerações Finais	63
6.1	Conclusões	63
6.2	Trabalhos Futuros	64
	Referências Bibliográficas	65

Capítulo 1

Introdução

A evolução dos dispositivos de comunicação sem fio proporciona e acelera o crescimento das redes móveis. Uma das principais características desse tipo de rede é a capacidade de comunicação entre os dispositivos, ou nós da rede, sem a necessidade de uma infraestrutura. Além de possuir um baixo custo de implantação e manutenção em comparação com as redes cabeadas, as redes móveis possibilitam a livre movimentação dos nós. Outro ponto interessante é a possibilidade de serem empregadas em locais de difícil acesso e locais distantes geograficamente de centros urbanos.

Uma das formas de classificação das redes móveis sem fio é com base nas características de mobilidade dos nós, podendo ser classificadas em *Redes Infraestruturadas* e *Mobile Ad hoc Networks – MANETs* [12, 20]. Nas *Redes Infraestruturadas* a comunicação entre os nós ocorre por meio das infraestruturas fixas de comunicação responsáveis pelo funcionamento da rede. Dessa forma, mesmo os nós localizados próximos uns dos outros não se comunicam diretamente. Nas *MANETs* a comunicação é realizada somente entre os nós da rede, não necessitando da disponibilidade de uma infraestrutura de comunicação de apoio. Toda vez que, dois ou mais nós, muitos distantes (fora de alcance do raio de comunicação) necessitam realizar a troca de informação, eles utilizam os nós intermediários para estabelecer o canal de comunicação.

Diferentemente das redes infraestruturadas, as MANETs utilizam protocolos de roteamento na rede, afim de estabelecer a melhor rota de comunicação entre os nós. Porém, devido a mobilidade dos nós, os protocolos de roteamento precisam tratar alguns desafios, de modo que, o roteamento nem sempre pode ser feito pelo mesmo caminho [12].

Um tipo especial de MANETs são as redes veiculares (*Vehicular Ad Hoc Networks – VANETs*), as quais são constituídas por veículos (nós), os quais possuem a capacidade de processamento, armazenamento e comunicação sem fio [39, 22]. Além disso, algumas das principais características das VANETs são: mobilidade previsível, alta velocidade, topologias altamente dinâmicas e fonte de energia altamente disponível devido à bateria dos veículos [32].

VANETs e MANETs apresentam algumas características semelhantes como, baixa largura de banda, transmissão de curto alcance e *broadcast* omnidirecional. No entanto, redes veiculares apresentam algumas características únicas como: topologias altamente dinâmicas, desconexões frequentes da rede, modelo de propagação e endereçamento geográfico [19].

Um dos grandes benefícios das VANETs é permitir a implementação de Sistemas de Transporte Inteligentes (*Intelligent Transportation Systems* – ITS) [35], para melhorar o gerenciamento do tráfego. ITS são soluções abrangentes para a gestão de tráfego em tempo real, o qual se baseia na coleta de dados de veículos, infraestruturas (*Road Side Units* – RSU) e outros sensores, os quais são componentes que podem interagir e cooperar entre si através da VANET [21].

Para melhorar o gerenciamento do tráfego, um ITS pode permitir a classificação das condições de tráfego nas vias, além de permitir detecção de congestionamentos e de acidentes e, roteamento dos veículos para melhorar a condição de tráfego nas vias para controlar e reduzir congestionamentos.

Além de permitir a implementação de ITS, as VANETs também permitem o desenvolvimento de varias outras aplicações, como: aplicações para alertas de situações de emergência, aplicações para entretenimento e acesso a Internet. Adicionalmente, as VANETs também proporcionam um maior conforto aos motoristas e passageiros.

1.1 Motivação

Os grandes centros urbanos têm sofrido com o grande crescimento do número de veículos. Neste contexto, os congestionamentos tornam-se uma preocupação para as autoridades, os quais geram diversas questões negativas para a sociedade e meio ambiente, incluindo acidentes, desenvolvimento econômico e poluição. Por exemplo, os congestionamentos geram perdas econômicas, reduzem a produtividade global, geram bloqueio de ruas impedindo a passagem de carros de polícia e ambulâncias em situações de emergência e aumentam as emissões CO devido ao alto consumo de combustível em períodos ociosos de acelerações frequentes e frenagens [4, 14].

Um estudo realizado em 2013 por Marcos Cintra, Ph.D. em Economia pela universidade Harvard, revela que os prejuízos de custo ligados aos congestionamentos na cidade de São Paulo estão associados ao tempo perdido, combustível consumido e aumento nas emissões de CO. O tempo perdido nos congestionamentos resulta em 85% dos prejuízos totais, enquanto o consumo de combustível e emissões de CO resultam em respectivamente 13% e 2% dos prejuízos causados pelos congestionamentos. O estudo também revela que as despesas geradas pelos congestionamentos são crescentes ao longo do tempo e passaram de R\$ 7 bilhões em 2002 para R\$ 10 bilhões em 2012. Já os custos do tempo perdido saltaram de R\$ 10,3 bilhões para R\$ 30,2 bilhões no mesmo período, resultando em um total de perdas em 2012 na ordem de R\$ 40 bilhões [8]. Os gastos com o congestionamento foram estimados em um estudo do Instituto de Pesquisa Econômica Aplicada (IPEA), em parceria com a Associação Nacional de Transporte Público (ANTP), considerando-se o consumo de combustível e a emissão de poluentes relacionados à velocidade de circulação dos veículos.

A procura de uma solução para evitar congestionamentos é muito clara. A aplicação mais popular para este fim é Waze¹, que ganha cada vez mais atenção nos últimos anos. No entanto, Waze está longe de atingir os seus objetivos, uma vez que depende exclusivamente

¹<https://www.waze.com>

de informações postadas por pessoas que usam seus *smartphones*. Além disso, as rotas sugeridas para os usuários são baseadas exclusivamente em algoritmos que não levam em conta o impacto que a mudança de rota tem nas condições de tráfego futuras. Outras soluções baseadas em redes veiculares podem ser encontradas na literatura [26, 10, 3, 35, 7, 11, 40, 29]. Estas soluções se concentram em resolver apenas parte do problema, como, Bauza et al. [3], que incidem sobre a detecção de congestionamento, enquanto [26, 7, 10, 29] focam apenas no roteamento dos veículos.

1.2 Objetivos

Este trabalho tem como objetivo principal o desenvolvimento de Sistemas de Transporte Inteligentes baseados em redes veiculares para controle de congestionamentos de veículos em centros urbanos. Os sistemas propostos visam detectar a iminência de congestionamentos, classificar as condições de tráfego nas vias e balancear o tráfego de uma forma inteligente para distribuir a densidade de veículos, a fim de evitar congestionamentos e maximizar o fluxo de veículos.

Os objetivos específicos deste trabalho são:

- *Objetivo 1 - Controle de congestionamentos causados por acidentes de trânsito:* Criar um sistema para identificar a ocorrência do acidente e evitar o surgimento de um congestionamento no local do acidente. Para isso, o sistema deve implementar os seguintes módulos:
 - *Detecção de acidentes:* implementar um mecanismo de detecção de acidentes, onde o mesmo possa detectar a ocorrência de um acidente em tempo real e coletar informações sobre o acidente, para que os veículos próximos possam ser alertados.
 - *Disseminação de alertas:* implementar um protocolo para disseminar as mensagens de alerta de acidentes em tempo real, onde o mesmo deve maximizar a entrega das informações para todos os veículos em uma área de interesse em diversas condições de tráfego com baixo *overhead* e pequenos atrasos.
 - *Gerenciamento do tráfego de veículos:* implementar um mecanismo de tomada de decisão e alteração de rota em tempo real, onde quando um veículo recebe a mensagem de alerta de acidente, o mesmo verifica se vai passar pelo local do acidente e caso necessário calcular uma nova rota para evitar o surgimento de um congestionamento.
- *Objetivo 2 - Controle de congestionamentos causados pelo aumento do tráfego:* Criar um sistema para detecção e controle de congestionamentos, onde o mesmo possa detectar quando um congestionamento ocorre e, logo em seguida gerenciar o tráfego dos veículos afim de controlar e reduzir os congestionamentos detectados. Para isso, o sistema deve implementar os seguintes módulos:

- *Classificação do tráfego*: implementar um classificador, para que o mesmo possa classificar as vias de acordo com a condição do tráfego, onde após a classificação seja possível detectar um congestionamento.
- *Deteção de congestionamentos*: implementar um mecanismo para detectar quais vias estão congestionadas, de acordo com a classificação da condição de tráfego, onde após a deteção das vias congestionadas seja possível verificar quais veículos irão passar por elas.
- *Roteamento dos veículos*: implementar algoritmo para rotear os veículos no cenário que irão passar por alguma via congestionada, onde o roteamento deve obedecer a condição de tráfego em cada via. Para isso, duas abordagens serão utilizadas:
 - * **Roteamento de melhor caminho**: implementar um algoritmo de roteamento, onde o mesmo selecione as melhores rotas para cada veículo, respeitando as condições de tráfego. Além disso, o algoritmo deve minimizar o surgimento de congestionamentos em outras áreas.
 - * **Roteamento cooperativo**: implementar um algoritmo de roteamento cooperativo, onde o mesmo selecione as melhores rotas que contribuam para uma melhor condição de tráfego no cenário todo.

1.3 Contribuições

Os resultados dessa dissertação de mestrado foram publicados nos seguintes eventos:

- 1. **SOUZA, A. M.** ; MAIA, G. ; VILLAS, L. A. . ADD: A Data Dissemination Solution for Highly Dynamic Highway Environments. In: The 13th IEEE International Symposium on Network Computing and Applications, NCA-2014, Cambridge.
- 2. **SOUZA, A. M.** ; BOUKERCHE, A. ; MAIA, G. ; MENEGUETTE, R. ; LOUREIRO, A. A. F. ; VILLAS, L. A. . Decreasing Greenhouse Emissions Through an Intelligent Traffic Information System Based on Inter-Vehicle Communication. In: The 12th ACM International Symposium on Mobility Management and Wireless Access, MOBIWAC-2014, Montreal.
- 3. **SOUZA, A. M.** ; Villas, L. A. A new Solution based on Inter-Vehicle Communication to Reduce Traffic jam in Highway Environment. Revista IEEE América Latina, v. 13, 2015.
- 4. **SOUZA, A. M.** ; Sadao, R. Y. ; MAIA, G. ; LOUREIRO, A. A. F ; Villas, L. A. Minimizing Traffic Jams in Urban Centers Using Vehicular Ad Hoc Networks. In: 7th IFIP International Conference on New Technologies, Mobility and Security. 2015, Paris.
- 5. **SOUZA, A. M.** ; Sadao, R. Y. ; FONSECA, N. ; Villas, L. A. GARUDA: A New Geographical Accident Aware Solution to Reduce Urban Congestion. In: In:

15th IEEE International Conference on Computer and Information Rechnology . CIT-2015, Liverpool.

- 6. **SOUZA, A. M.** ; Sadao, R. Y. ; BOTEGA L. ; Villas, L. A. SCORPION: A Solution using Cooperative Rerouting to Prevent Congestion and Improve traffic Condition. In: In: 15th IEEE International Conference on Computer and Information Rechnology. CIT-2015, Liverpool.

Além disso, esse trabalho contribuiu para o seguinte trabalho:

- 1. BRENNAND, C. ; **SOUZA, A. M.** ; MAIA, G. ; BOUKERCHE, A. ; LOUREIRO, A. A. F. ; Villas, L. A. An Intelligent Transportation System for Detection and Control of Congested Roads in Urban Centers. In: The Twentieth IEEE Symposium on Computers and Communications. ISCC-2015, Cyprus.

Por fim, os seguintes trabalhos foram submetidos para periódicos:

- 1. **SOUZA, A. M.** ; SADAIO R. Y. ; BOUKERCHE, A. ; MAIA, G. ; CERQUEIRA E. ; LOUREIRO, A. A. F. ; Villas, L. A. ICARUS: Improvement of Traffic Condition through an Alerting and Re-routing System. Revista ELSEVIER Computer Networks (COMNET).
- 2. **SOUZA, A. M.** ; SADAIO R. Y. ; CERQUEIRA E. ; LOUREIRO, A. A. F. ; Villas, L. A. SPARTAN: System to Prevent Congestion caused by Accidents with Real-Time Alert and Re-routing. Revista ELSEVIER Performance Evaluation.

1.4 Estrutura da Dissertação

Esta dissertação é dividida em seis capítulos, onde no Capítulo 2, é apresentada a fundamentação teórica utilizada para o desenvolvimento do trabalho. No Capítulo 3, são apresentados os trabalhos relacionados. No Capítulo 4, é apresentado um sistema para controle de congestionamentos causados por acidentes de veículos. No Capítulo 5, é apresentado um sistema para detecção, classificação e controle de congestionamentos. Por fim, no Capítulo 6, são apresentadas as conclusões e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo são apresentados os conceitos básicos de redes veiculares, a sua arquitetura, o padrão 802.11p, algumas características das redes veiculares e desafios. Por fim é apresentada uma breve consideração final.

2.1 Redes Veiculares

As redes *Ad hoc* têm ganho destaque entre as redes sem fio, devido aos avanços nas tecnologias de curto alcance. Uma rede *ad hoc* é constituída por nós equipados com interfaces de comunicação sem fio, as quais permitem que os nós se comuniquem sem a necessidade de qualquer tipo de infraestrutura de apoio [24]. As principais redes *ad hoc* são: MANETs, *Wireless Mesh Networks* (WMNs), *Wireless Sensor Networks* (WSNs) e *Vehicular Ad hoc Networks* (VANETs).

As redes veiculares constituem um tipo especial de redes móveis sem fio, em que os nós móveis são constituídos por veículos que possuem a capacidade de processamento, armazenamento e comunicação sem fio. Os veículos podem se comunicar entre si e com equipamentos de infraestruturas fixos, posicionados às margens das ruas e/ou estradas, como as *Road Side Units* (RSUs). Nas redes veiculares os nós podem ser origem ou destino de informações e também podem atuar como roteadores para encaminhamento de informações [19]. A arquitetura das redes veiculares define a forma como os nós se organizam e se comunicam. Sendo assim, a comunicação pode ser realizada de três maneiras [14]. A Figura 2.1 mostra as arquiteturas de uma rede veicular.

- **Veículo-Veículo (V2V):** permite a comunicação direta entre veículos sem depender do auxílio de uma infraestrutura fixa [14]. Portanto, os próprios veículos encarregam de realizar a disseminação dos dados na rede, encaminhando-os por meio de outros veículos por múltiplos saltos. Apesar dessa estrutura de redes ser simples, ela possui a desvantagem da conectividade depender diretamente da densidade e do padrão de mobilidade dos veículos;
- **Veículo-Infraestrutura (V2I):** permite a comunicação dos veículos com uma infraestrutura de comunicação distribuída e estática às margens de ruas ou rodovias, conhecidas como RSUs [19]. As RSUs centralizam todo o tráfego da rede, servindo

como nós intermediários de comunicação. A vantagem da arquitetura V2I é a ampliação da conectividade e a possibilidade de comunicação com outras redes, como a Internet. Entretanto, todo esse benefício só é possível mediante a instalação de inúmeras RSUs nas margens de ruas e rodovias, o que aumenta muito o custo da implantação;

- **Arquitetura híbrida:** combina arquiteturas V2V e V2I. Neste caso, utiliza-se uma infraestrutura para aumentar a conectividade da rede, isto é, um veículo pode comunicar com a infraestrutura rodoviária em um único salto ou por múltiplos saltos com outros veículos de acordo com a localização do nó na rede.

Para realizar as comunicações nas VANETs foi necessário a evolução da tecnologia sem fio, em particular do padrão 802.11. Dessa forma, o *Institute of Electrical and Electronics Engineers* (IEEE) desenvolveu um padrão para realizar as comunicações nas VANETS, o padrão IEEE 802.11p [14].

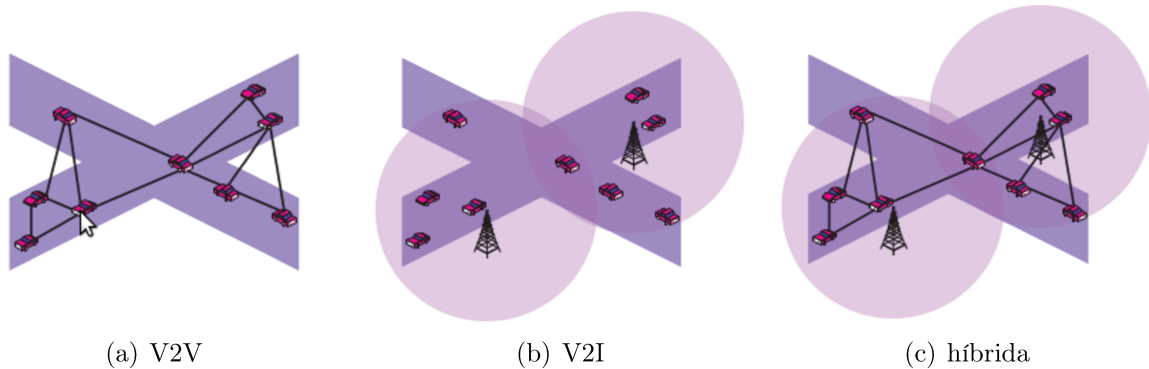


Figura 2.1: Arquiteturas de uma VANET

2.1.1 Padrão IEEE 802.11p

IEEE 802.11p também conhecido como *Wireless Access in a Vehicular Environment* (WAVE), é o protocolo utilizado para comunicações entre veículos em redes veiculares. Cabe ressaltar que o padrão WAVE (802.11p) é uma extensão do padrão IEEE 802.11a, ajustado para operações com baixo *overhead* no espectro DSRC [14].

Dedicated Short-Range Communication (DSRC) [19], é uma faixa de frequência usada para comunicação de curto alcance de forma *ad hoc*. O DSRC ainda não é um padrão de fato, embora há tentativas de padronizá-lo. Em 1999 nos Estados Unidos, a *Federal Communications Commission* (FCC) alocou 75 MHz do espectro de frequências, na faixa de 5,9 GHz. Já na Europa, é utilizado o intervalo de frequência de 5725 a 5875 MHz e no Japão está sendo utilizado na faixa de 5,8 GHz [14].

O WAVE foi proposto para garantir confiabilidade às operações de aplicações de segurança. Para tanto, WAVE faz uso de múltiplos canais de comunicação. Conforme mostra a Figura 2.2, o FCC nos Estados Unidos reservou sete canais de comunicação não sobrepostos na faixa de frequência de 5.9 GHz apenas para a comunicação de veículos. Um

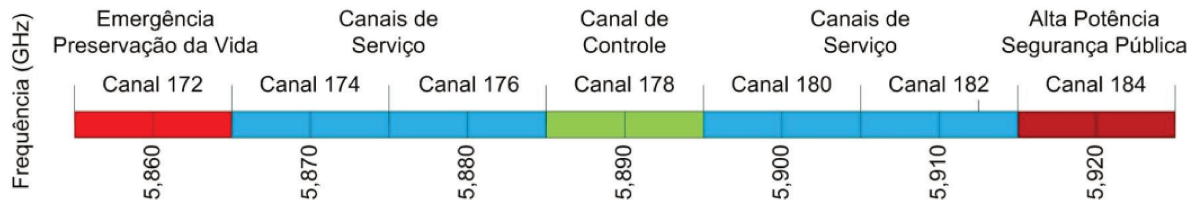


Figura 2.2: Canais disponíveis para o IEEE 802.11p

desses canais é designado como o Canal de Controle (CCH - *Controle Channel*) e os seis demais como Canais de Serviço (SCH - *Service Channel*).

O padrão IEEE 802.11p define melhorias, em relação ao padrão 802.11, necessárias para as aplicações de sistemas de transporte inteligentes. As melhorias incluem a capacidade de troca de conteúdos entre veículos que estão trafegando em alta velocidade, e entre veículo em movimento e a infraestrutura. Há dois principais parâmetros na camada física do padrão 802.11p: (i) Taxas de transferência de dados que podem ser ajustadas em: 3, 4.5, 6, 9, 12, 18, 24 e 27 Mbps (*Megabits per second*) e (ii) as modulação BPSK (*Binary Phase Shift Keying*), QPSK (*Quadrature Phase Shift Keying*), 16 e 64 QAM (*Quadrature Amplitude Modulation*). Em relação à camada de enlace, emprega o protocolo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) como método de acesso ao meio [13].

A arquitetura WAVE é definida pelos padrões da família IEEE 1609 em conjunto com IEEE 802.11, isto é, a arquitetura WAVE é definida por meio de seis documentos: IEEE P1609.1, IEEE P1609.2, IEEE P1609.3, IEEE P1609.4, IEEE 802.11 e IEEE 802.11p [13]. Esses documentos descrevem todas as camadas da pilha de protocolos WAVE e o objetivo principal da arquitetura WAVE é possibilitar as comunicações V2V e V2I. A Figura 2.3 apresenta a pilha de protocolos WAVE.

- **IEEE P1609.1:** Descreve os serviços de dados e gerenciamento dentro da arquitetura WAVE. Define os serviços da aplicação de Gerente de Recursos dessa arquitetura. Especifica formatos de mensagens de comando e as respostas apropriadas para essas mensagens, os formatos de armazenamento de dados que devem ser usados pelas aplicações para comunicações entre os componentes da arquitetura e os formatos de mensagens de status e de pedido;
- **IEEE P1609.2:** Define os formatos de mensagens seguros e seu processamento. Define os momentos que devem ser utilizadas essas mensagens seguras e como deve ser feito o processamento delas de acordo com o seu propósito;
- **IEEE P1609.3:** Define os serviços das camadas de rede e de transporte, incluindo roteamento e endereçamento, para dar suporte seguro a troca de dados da arquitetura WAVE. Para tal define um plano de dados e um de gerenciamento, explicados a seguir;
 - **Plano de dados:** Especifica os protocolos de comunicação que são responsáveis pelo envio de dados gerados pelas aplicações WAVE e também pelo tráfego que as entidades do plano de gerenciamento geram na rede.

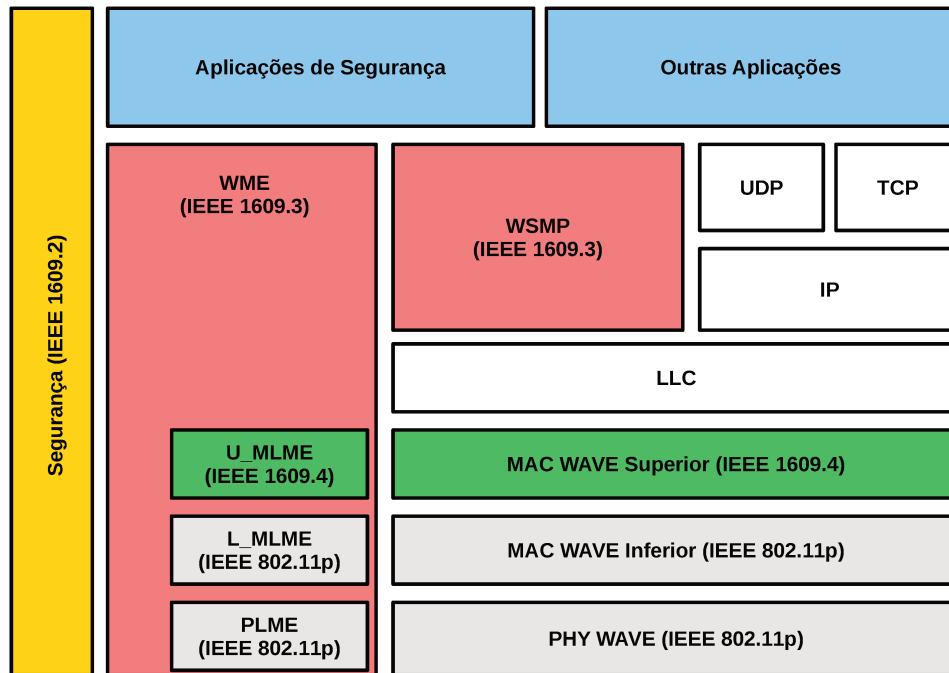


Figura 2.3: Arquitetura WAVE

- **Plano de gerenciamento:** Configura e faz a manutenção do sistema. Implementa serviços de monitoramento da utilização de canais, registro de aplicações e manutenção da base de dados de gerenciamento.
- **IEEE 802.11p:** define as diferenças específicas do controle de acesso ao meio em ambientes de comunicação WAVE com relação ao IEEE 802.11 tradicional.

Os documentos da família IEEE 1609 definem um conjunto de padrões para a comunicação em VANETs. O principal objetivo dessa família é prover um conjunto padronizado de interfaces para que diferentes fabricantes de automóveis possam prover comunicações entre veículos ou veículo e infraestrutura [14]. Pode-se observar que a arquitetura WAVE (veja a Figura 2.3) possui duas pilhas de protocolos, uma para o padrão da Internet (IP - *Internet Protocol*) e outra com base no protocolo WSMP (*WAVE Short Message Protocol*), a qual foi projetada visando a comunicação em ambientes veiculares.

2.1.2 Características

As VANETs possuem suas próprias características quando comparadas com outros tipos de MANETs, onde essas características próprias incluem:

- **Mobilidade previsível:** as VANETs se diferem dos outros tipos de redes *ad hoc* móveis nas quais os nós se movem de forma aleatória, devido ao fato da mobilidade dos veículos ser limitada pela topologia do cenário (ruas e estradas), além disso os veículos têm a obrigação de obedecer a sinais de trânsito e semáforos [14, 24, 19], resultando assim em uma mobilidade previsível;

- **Sem restrições de energia:** a energia em VANETs não é um desafio como nas MANETs, porque os veículos têm a capacidade de fornecer energia contínua para as unidades de processamento, armazenamento e comunicação através das baterias de longa duração [19];
- **Densidade variável:** a densidade da rede nas VANETs varia dependendo do tráfego, o qual pode ser muito alto durante as ocorrências de congestionamentos ou horários de pico e, muito baixa em tráfegos de bairros suburbanos [19];
- **Topologias altamente dinâmicas:** a alta mobilidade dos nós impactam diretamente em rápidas mudanças de topologia. Além disso, o tempo em que os veículos permanecem conectados depende da direção em que os veículos estão se movendo e, também, do alcance da interface de comunicação equipada nos veículos.
- **Alto poder computacional:** devido ao fato dos nós em VANETs serem veículos, os mesmos podem ser equipados com um número expressivo de sensores e recursos computacionais, como: processadores, memórias com grandes capacidades, GPS. Esses recursos aumentam a capacidade computacional do nó, o qual ajuda a obter uma comunicação confiável e coletar informações em relação à posição corrente, velocidade e direção.

2.2 Desafios

Nesta seção são apresentados alguns desafios referentes ao desenvolvimento desse trabalho, onde os desafios englobam a disseminação de dados em redes veiculares, e o problema de sincronização introduzido pelo padrão IEEE 802.11p durante a disseminação de dados.

2.2.1 Disseminação de Dados

A alta mobilidade dos nós que compõem a rede e topologias altamente dinâmicas são características que geram grandes desafios para disseminar dados em redes veiculares. Esse alto dinamismo representa um empecilho nas comunicações e na disseminação de dados, visto que o tempo de contato entre os veículos na maioria das vezes é muito pequeno [18]. A disseminação de dados é uma função vital em VANETs, pois uma determinada informação pode interessar a um grupo de veículos de uma determinada região. Além disso, em cada cenário (rural, rodoviário e urbano) a rede veicular comporta de maneira distinta. Por exemplo, em cenário urbano a concentração de veículos é maior (rede densa) em relação ao cenário rural (rede esparsa). Com isso, uma solução de disseminação de dados proposta para cenários rurais pode causar alta taxa de *overhead* quando aplicada em cenários urbanos. Por outro lado, uma solução proposta para cenário urbano pode acarretar uma queda de desempenho quando aplicada em cenário rural, devido à ausência de veículo para continuar o processo de disseminação de dados.

A disseminação dos dados pode ser realizada por meio do mecanismo de difusão, mais conhecido como *broadcast*. No mecanismo de *broadcast* os veículos podem disseminar os dados sem a necessidade de conhecer o endereço e a rota para um destino final. Com isso

elimina a complexidade da descoberta de rotas e da gerência da topologia [27]. Devido as característica das redes veiculares, a disseminação de dados possui alguns desafios, sendo eles [9]:

- **Tempestade de *broadcast*:** ocorre quando inúmeros veículos tentam transmitir simultaneamente, ocasionando elevado tráfego de dados, congestionamento de rede, colisões de pacotes e atraso adicional no controle de acesso ao meio (MAC).
- **Partição de rede:** ocorre quando há grandes obstáculos (prédios, estádios, lagos) entre as comunicações, de modo que os veículos não possam dar continuidade ao processo de disseminação.
- **Fragmentação temporal da rede:** ocorre quando a quantidade de veículos é insuficiente para prover a disseminação de dados para todos os veículos da rede, a partição da rede é temporária e de curta duração. Esse problema ocorre devido à alta mobilidade dos nós da rede e perda de pacotes e colisões.

2.2.2 Sincronização Introduzida pelo Padrão IEEE 802.11p

O padrão IEEE 802.11p não exige a utilização de múltiplas antenas, sendo assim, um mecanismo de salto de canais é proposto. Nesse esquema, a cada T_c segundos é permitido ao rádio saltar do CCH para o SCH e então, após mais T_c segundos, saltar do SCH para o CCH, e assim por diante. O padrão estabelece $T_c = 50$ ms. Diante disso, quando a camada MAC recebe da camada acima uma mensagem para ser transmitida no SCH, mas o CCH está momentaneamente ativo, então a mensagem deve aguardar o SCH se tornar ativo para que a transmissão possa ocorrer. No começo da operação de cada canal existe um intervalo de 5 ms chamado *guard interval* em que o canal é tratado como ocupado. Além disso, assume-se que os *beacons* periódicos, normalmente utilizados pelas aplicações de segurança, irão utilizar o CCH, enquanto as demais aplicações irão utilizar o SCH. Toda mensagem repassada para a camada MAC deve especificar em qual canal ela deve ser transmitida.

A abordagem de salto de canais feita pelo protocolo IEEE 802.11p introduz um efeito de sincronização indesejado. Considere o exemplo mostrado na Figura 2.4. Nesta Figura, há dois veículos tentando retransmitir uma mensagem que eles receberam de um vizinho em comum. No tempo T_1 , ambos os veículos agendam o repasse da mensagem para a camada MAC. Perceba que, o veículo *A* calcula e agenda o repasse com um atraso de 10 ms, enquanto *B* calcula e agenda o repasse com um atraso de 25 ms. Sem o mecanismo de salto de canais, *A* iria transmitir primeiro, *B* iria escutar a transmissão de *A* e consequentemente iria cancelar sua própria transmissão, evitando assim uma transmissão desnecessária. No entanto, não é o que ocorre quando a abordagem de salto de canais é empregada. No tempo T_2 , ou seja, 10 ms depois de T_1 , *A* repassa a mensagem para a camada MAC para transmissão no SCH. Porém, perceba que o CCH está momentaneamente ativo. Logo, a camada MAC do veículo *A* armazena a mensagem até que o SCH fique ativo. No tempo T_3 , ou seja, 25 ms depois de T_1 , *B* repassa a mensagem para a camada MAC, também para transmissão no SCH. Assim como em *A*, a camada MAC do veículo *B* também armazena

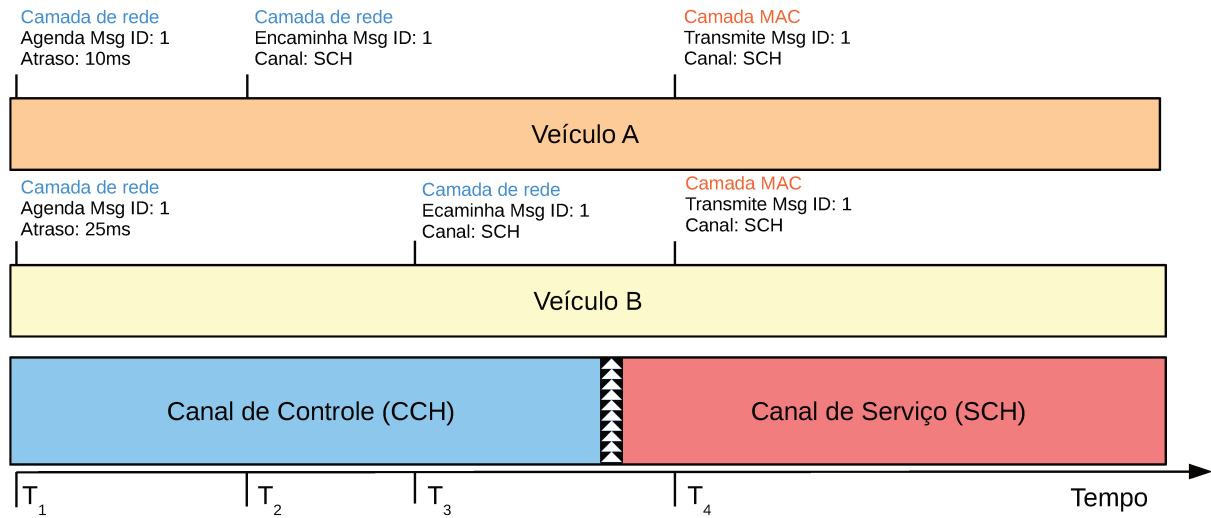


Figura 2.4: Efeito de sincronização introduzido pelo padrão 802.11p

a mensagem, já que o CCH está momentaneamente ativo. Finalmente, no tempo T_4 , quando o SCH se torna ativo, a camada MAC de ambos os veículos escuta e percebe o canal como não ocupado. Assim, os veículos transmitem a mensagem ao mesmo tempo, resultado em uma colisão.

2.3 Cenários

Os cenários em redes veiculares geralmente são classificados em três grupos: urbano, rural e rodoviário [24]. A seguir estão os detalhes de cada um deles:

- **Cenário urbano:** na maioria das vezes a densidade dos veículos tende ser alta e a velocidade com que eles trafegam geralmente é baixa. Nessa situação a comunicação por múltiplos saltos pode ser favorecida. Mas, esse cenário pode variar de acordo com o horário, por exemplo: durante a madrugada o tráfego de veículos geralmente é muito baixo, este cenário pode prejudicar a comunicação por múltiplos saltos. Além disso, acidentes e obras na pista podem influenciar o comportamento do trânsito prejudicando a comunicação;
- **Cenário rodoviário:** os veículos na maioria das vezes trafegam com alta velocidade, uma vez que não há semáforos nem cruzamentos. O tempo de conectividade entre os veículos que estão trafegando na mesma direção, neste tipo de cenário, tende a ser maior do que em cenários urbanos e, também há uma forte tendência de formar grupos de veículos, ou seja, *clusters*;
- **Cenário rural:** a densidade dos veículos é baixa resultando uma rede com conectividade intermitente. Dessa forma a rede é particionada em pequenos grupos de veículos formando *clusters*. Como a conectividade é intermitente, logo há necessidade de elaborar soluções para encaminhar os pacotes de dados, uma delas é o emprego das RSUs.

2.4 Resumo Conclusivo

Neste capítulo, foram apresentados os conceitos básicos sobre as VANETs, os quais incluem as arquiteturas de comunicação (V2V, V2I e comunicação híbrida). Em seguida, foi apresentado o padrão IEEE 802.11p utilizado nas comunicações nas VANETs. Além disso, foram detalhadas as características das redes veiculares e por fim foram apresentados alguns desafios referentes a disseminação de dados nas VANETs e o problema de sincronização introduzido pelo padrão IEEE 802.11p.

Capítulo 3

Trabalhos Relacionados

Neste capítulo serão apresentados alguns trabalhos relacionados utilizados para desenvolvimento do sistema. Na seção 3.1 são apresentados os trabalhos relacionados referentes a disseminação de dados, na seção 3.2 são apresentados os trabalhos relacionados à detecção de congestionamentos. Além disso, na seção 3.2 são apresentados os trabalhos relacionados referentes ao controle de congestionamentos em redes veiculares. Por fim, na seção 3.3 é apresentada uma conclusão dos sistemas propostos em comparação aos trabalhos relacionados.

3.1 Disseminação de Dados

Flooding é a forma básica de disseminar dados ao longo da rede. Basicamente a ideia do Flooding é disseminar os dados para todos os veículos vizinhos, que, em seguida, armazenam e encaminham os dados para seus respectivos vizinhos. Esta abordagem apresenta um bom desempenho para redes esparsas, mas rapidamente se incorre no problema de tempestade de *broadcast* quando a densidade da rede aumenta.

Tempestade de *broadcast* é um grande problema em disseminação de dados em VANETs. Devido a isso, vários estudos na literatura têm sido propostos apenas para lidar com esse problema. Alguns protocolos de disseminação como, DV-CAST, AID e DBRS e DRIVE são descritos abaixo.

DV-CAST (*Distributed Veicular Broadcast*) proposto por Tonguz et al. [31] é um protocolo o qual utiliza uma topologia local de vizinhos mais próximos (vizinhos de um salto) como principal critério para determinar como a mensagem será retransmitida. O DV-CAST implementa dois mecanismos, sendo eles: (i) detecção de veículos vizinhos e; (ii) supressão de *broadcast*. Dessa forma, o mecanismo *detecção de vizinhos* estima a topologia local por meio do monitoramento dos *beacons* trocados periodicamente pelos veículos. Portanto, com a topologia local é possível determinar a direção em que os vizinhos estão trafegando e se os mesmos são bons candidatos para continuar a retransmissão da mensagem. Por outro lado, para realizar a supressão de *broadcast* o protocolo utiliza a técnica de *weighted p-persistence* $P_{ij} = \frac{D_{ij}}{R}$ [38], onde, D_{ij} é a distância relativa entre os veículos transmissor (i) e o receptor (j) da mensagem e R é a média do raio de transmissão entre os veículos. Entretanto, o DV-CAST depende dos *beacons* periódicos trocados

pelos veículos para manter a topologia local em cada veículo, dessa forma em um cenário denso, essa constante troca de *beacons* entre os veículos incorre em uma sobrecarga na rede, a qual já possui uma largura de banda limitada [14, 19].

O protocolo *Approach for Information Dissemination* (AID) é uma abordagem descentralizada e adaptável para a disseminação de informação em VANETs [2]. O AID utiliza um método com base estatística para realizar a disseminação. No AID, cada veículo decide se deve ou não retransmitir a mensagem com base do cálculo do limiar $(\frac{\gamma}{C} - \Delta t_i)$, onde, γ é o tempo de espera aleatório gerado a partir do recebimento de uma mensagem, C é a quantidade de mensagens duplicadas recebidas no intervalo γ e, por fim, Δt_i é o intervalo de cada mensagem duplicada recebida. Quando o tempo de espera γ espira, se o valor de C é baixo, isso significa que há poucos veículos informados e consequentemente o intervalo Δt_i é alto. Dessa forma, o cálculo do limiar será maior que zero $((\frac{\gamma}{C} - \Delta t_i > 0))$, caso contrário o valor será menor que zero. Portanto, se o limiar for maior que zero o veículo transmite a mensagem, caso contrário não retransmite, reduzindo assim o problema de *tempestade de broadcast*. Porém, a utilização do tempo de espera aleatório γ utilizado pelo AID, pode gerar um atraso muito alto nas retransmissões.

DBRS (*Distance Based Relay Selection*) proposto por Kim et al. [15] é um protocolo baseado em distância. O DBRS utiliza um atraso para retransmissão das mensagens baseado na distância inversamente proporcional entre emissor e receptor, portanto, a mensagem sempre será retransmitida pelo veículo mais distante. Por consequência, os veículos mais próximos do veículo emissor, recebem uma mensagem duplicada e, então, os veículos que recebem a mensagem duplicada, cancelam o agendamento da mesma. Entretanto, essa abordagem ao evitar o problema de *tempestade de broadcast* pode acarretar em dois problemas, sendo eles: (i) o atraso pode ser muito alto, pois não há garantia da existência de veículos próximo a borda do raio de comunicação do veículo emissor e, (ii) a cobertura pode ser reduzida, uma vez que os veículos suspenderão as suas transmissões de forma indiscriminada ao receber o mesmo pacote.

O protocolo *Data Dissemination Protocol in Vehicular Networks* (DRIVE) [34], seleciona alguns veículos com maior prioridade, que estão dentro do raio de transmissão do veículo emissor, para dar continuidade ao processo de disseminação. Os veículos que têm maior prioridade para retransmissão da mensagem são aqueles que estão mais próximo da borda do raio de transmissão do veículo emissor, uma região nomeada de *sweet spot*. Portanto, os veículos localizados nessa região têm um tempo de espera inferior aos veículos fora dessa região, dessa forma transmitindo primeiro do que os veículos fora dessa região. Além disso, para todos os veículos dentro do raio de comunicação, um atraso padrão é calculado, baseado na fórmula $T_j = 0.01 \times \frac{D_{ij}}{R}$, onde D_{ij} é a distância entre veículos emissor e receptor e R é o raio de transmissão do veículo. Além disso, em uma rede esparsa, sempre que uma partição da rede é detectada por um veículo de origem, DRIVE usa o conceito de zona de recuperação. Veículos fora da área de interesse são usados para disseminar dados sobre o evento que ocorreu dentro da área de interesse. A principal finalidade do uso de uma zona de recuperação é realizar a disseminação de dados para veículos separados por partições de rede que estão dentro da área de interesse. Entretanto, o DRIVE não trata o problema de sincronização introduzido pelo padrão 802.11p igualmente aos protocolos Flooding, DV-CAST, AID e DBRS.

Para realizar a disseminação de alerta no SPARTAN, foi desenvolvido um protocolo para disseminação de dados nomeado ADD [30], onde o protocolo trata das limitações apresentadas pelos protocolos AID, DV-CAST, DBRS e DRIVE, como: sobrecarga do canal de comunicação [31], alta atraso nas retransmissões das mensagens [2], cobertura reduzida [15] e sincronização introduzida pelo padrão IEEE 802.11p [31, 2, 15, 34].

A Tabela 3.1 apresenta uma comparação quantitativa entre os protocolos de disseminação apresentados, a comparação é feita utilizando: Estratégia de retransmissão, Objetivos, e Cenário.

Tabela 3.1: Uma comparação quantitativa entre as soluções de disseminação

Protocolo	Estratégia de retransmissão				Objetivos			Cenário		
	Posição	Estatístico	Store-Carry-Forward	Distância	Temporizador	Tempestade broadcast	Partição de rede	Alerta da troca de canais	Rodovia	Urbano
AID [2]		✓			✓				✓	
DBRS [15]	✓			✓					✓	
DRIVE [34]	✓		✓	✓	✓	✓		✓	✓	
DV-CAST [31]	✓		✓	✓	✓	✓		✓		
ADD [30]	✓		✓	✓	✓		✓	✓	✓	

3.2 Detecção e Controle de Congestionamentos

Waze é uma aplicação de navegação geográfica baseada em GPS para smartphones que apresenta informações que descrevem informações *turn-by-turn*¹. Além da navegação *turn-by-turn* de voz, tráfego em tempo real, e outros alertas de localização específica, Waze envia simultaneamente informações anônimas, incluindo a velocidade e localização dos usuários de volta à sua base de dados, afim de melhorar a rota para cada veículo, planejando evitar áreas congestionadas. No entanto, uma desvantagem desta abordagem é a possibilidade de mover o congestionamento para outro local. Uma vez que as decisões de replanejamento são realizadas localmente para descobrir caminhos ideais para os indivíduos, em vez de equilibrar o fluxo de tráfego. Além disso, alguns dos problemas e complicações da Waze mais comuns incluem a distância limitada de roteamento, cálculo da rota baseado apenas na distância, limitações ao apoio do modelo do smartphone, problemas de navegação, e falhas do programa em alguns locais físicos [36].

Bauza et al. [3] apresentam CoTEC (*CO*operative *Traffic congestion detECTION*), um sistema veicular cooperativo baseado em comunicações veículo-a-veículo (V2V) para detectar um congestionamento do tráfego usando lógica fuzzy. CoTEC usa CAM (*CO*operative *Awareness Messages*) ou mensagens de *beacons*, periodicamente transmitidas por

¹<https://www.waze.com>

todos os veículos para informá-los sobre as condições de tráfego na estrada. Além disso, o CoTEC utiliza lógica fuzzy para detectar uma potencial condição de congestionamento do tráfego rodoviário localmente em cada veículo. Portanto, quando um congestionamento for detectado, num primeiro momento, cada veículo transmite sua própria estimativa sobre o congestionamento e, em seguida, com todas as estimativas, os veículos de forma colaborativa detectam e caracterizam o congestionamento. No entanto, o mecanismo utilizado para identificar a condição de tráfego pode causar uma sobrecarga na rede devido às mensagens periódicas de *beacons* e as estimativas locais disseminados por todos os veículos. Além disso, apesar de ser capaz de detectar um congestionamento, nenhum mecanismo para minimizar ou controlar o congestionamento é apresentado no seu trabalho. Em adição, o CoTEC foi proposto para operar exclusivamente em cenários de rodovia.

Em [26] é proposto um sistema para predição, detecção e controle de congestionamentos afim de reduzir o tempo de viagem, o consumo de combustível e as emissões de CO. O sistema utiliza uma abordagem centralizada para coleta de informações sobre o tráfego dos veículos (posição, velocidade e direção) em tempo real. A informação coletada é utilizada para detectar ou predizer um congestionamento. Após a detecção ou predição do congestionamento, o sistema possui um mecanismo de controle de congestionamento, onde é realizado o re-roteamento dos veículos que se aproximam das áreas congestionadas. Para o mecanismo de controle de congestionamento, são apresentadas três estratégias de re-roteamento para os veículos, sendo a primeira conhecida como *Dynamic Shortest Path - DSP*, é a estratégia padrão de re-roteamento que associa ao veículo a rota de menor caminho com menor tempo de viagem. A segunda estratégia é a *Random k Shortest Paths - RkSP*, essa estratégia seleciona as k rotas de menor caminho, então uma dessas k rotas é selecionada aleatoriamente. Por fim é apresentada a estratégia *Entropy Balanced k Shortest Paths - EBkSP*, essa estratégia utiliza a mesma ideia da estratégia RkSP, entretanto é aplicado um mecanismo para seleção de rota mais inteligente, de modo que é considerado o impacto que cada k rota tem sobre a densidade futura da pista. Entretanto, o sistema proposto por Pan et al. não implementa nenhum mecanismo de detecção de congestionamentos ou acidentes, dessa forma os congestionamentos só serão detectados na próxima fase de roteamento.

De forma semelhante ao CoTEC, Araújo et al. [1] propõe um sistema para detecção de congestionamentos de forma colaborativa nomeado *Cooperative vehicular Traffic congestion Identification and Minimization* (CARTIM). CARTIM, faz uso apenas de comunicações V2V para mensurar o nível de congestionamento da via em que o veículo se encontra. Adicionalmente, para mensurar o nível de congestionamento localmente nos veículos o sistema utiliza lógica fuzzy, para tratar informações quantitativas. Por fim, quando um congestionamento é detectado, o sistema utiliza um mecanismo para evitar o ponto do congestionamento utilizando um algoritmo de roteamento de menor caminho. Porém, igualmente ao CoTEC, o CARTIM pode causar uma sobrecarga na rede devido aos *beacons* e mensagens de estimativas locais trocados periodicamente pelos veículos.

Doolan and Muntean [10] apresentam um algoritmo distribuído para re-roteamento de veículos que se baseia em comunicações V2V. O algoritmo também conhecido como EcoTrec faz uso de características da estrada e condições de tráfego para melhorar a eficiência do tráfego e reduzir o consumo de combustível. No algoritmo proposto, cada veículo

dissemina de tempos em tempo a sua rota e seu consumo de combustível. Essas informações são utilizadas para computar as melhores rotas para cada veículo e caso necessário novas rotas são sugeridas para os veículos visando o menor consumo de combustível. O cálculo das melhores rotas para cada veículo é feito de forma distribuída, de modo que cada veículo calcula sua melhor rota utilizando o algoritmo de menor caminho. Entretanto, o fato da utilização de um algoritmo de menor caminho para calcular uma nova rota incorre no mau balanceamento do tráfego, o qual pode mover um congestionamento para outro local.

Brennand et al. [7] propõem um ITS para informações em tempo real para detectar e controlar um congestionamento. O ITS proposto é composto por um conjunto de RSUs distribuídas pelo cenário a fim de prover cobertura total de uma cidade. Cada RSU é responsável por gerenciar os veículos e detectar um congestionamento apenas dentro do seu raio de comunicação. Além disso, o ITS inclui um mecanismo para controle de congestionamentos, o qual periodicamente faz o re-roteamento de todos os veículos, distribuindo os veículos, a fim de alterar as rotas dos mesmos para não passarem por áreas congestionadas. Da mesma forma que Pan et al. [26], o ITS proposto não detecta o congestionamento assim que ele ocorre, de modo que o mesmo será detectado apenas na próxima fase de re-roteamento.

Neste contexto, os sistemas SPARTAN e COLOSSUS, foram propostos para suprir as limitações apresentadas pelos trabalhos da literatura como: sobrecarga da rede [3, 1], roteamento limitado [36], ausência de um mecanismo para detecção de acidentes em tempo real [3, 1, 10, 26, 7] e detecção de congestionamentos durante a fase de roteamento [26, 7].

A Tabela 3.2 apresenta uma comparação quantitativa entre as soluções para controle e detecção de congestionamentos apresentadas, a comparação é feita utilizando: Design, Objetivos, Roteamento e Cenário.

3.3 Resumo Conclusivo

Neste capítulo, foram apresentados os trabalhos relacionados. Em particular, na Seção 3.1, foram apresentados os trabalhos relacionados a disseminação de dados e, na Seção 3.2, foram apresentados os trabalhos relacionados a detecção e controle de congestionamentos.

Para a disseminação de dados, foram apresentados os protocolos Flooding, DV-CAST, AID, DBRS e DRIVE, onde todos os protocolos apresentados com exceção do Flooding, tratam o problema de tempestades de *broadcast*, entretanto apesar dos protocolos apresentarem características distintas e tratar de outros desafios nenhum dos protocolos apresentados tratam o problema de sincronização introduzido pelo padrão IEEE 802.11p.

Para a detecção e controle de congestionamentos, foram apresentados os seguintes sistemas: WAZE, DSP, RkSP, With routing, CoTEC, EcoTrec e CARTIM. Entretanto, os sistemas apresentam algumas limitações, como: sobrecarga do canal de comunicação [3, 1], roteamento limitado [36], ausência de um mecanismo para detecção de congestionamento em tempo real [10, 26, 7]. Além disso, nenhum sistema apresentado possui algum mecanismo para controle de congestionamentos causados por acidentes.

Tendo em vistas essas questões, nos Capítulos 4 e 5, são apresentados SPARTAN e

Tabela 3.2: Uma comparação quantitativa entre as soluções para controle e detecção de congestionamentos

	Design		Objetivos					Roteamento			Cenário	
	Centralizado	Distribuído	Deteção de congestionamentos	Controle de Acidentes	Classificação do tráfego	Controle de congestionamentos	Roteamento periódico	Menor caminho	Balanceamento de tráfego	Cooperativo	Rodovia	Urbano
Protocolo												
DSP [26]	✓		✓			✓	✓	✓				✓
RkSP [26]	✓		✓			✓	✓		✓			✓
With routing [7]	✓		✓			✓	✓		✓			✓
CoTEC [3]		✓	✓		✓						✓	
EcoTrec [10]		✓				✓	✓				✓	
CARTIM [1]		✓	✓		✓	✓		✓				✓
SPARTAN		✓		✓		✓			✓		✓	✓
COLOSSUS	✓		✓		✓	✓		✓	✓	✓		✓

o COLOSSUS, sistemas para tratar as limitações apresentadas pelos trabalhos da literatura.

Capítulo 4

Controle de Acidentes

Neste capítulo, é apresentado o SPARTAN (*System to Prevent congestion caused by Accidents with Real-Time Alert and re-routiNg*). O sistema de transporte inteligente nomeado como SPARTAN visa reduzir os congestionamentos de veículos causados por acidentes de trânsito. Em linhas gerais, o SPARTAN detecta quando ocorre um acidente, alerta os veículos dentro da área de interesse (AoI) ¹ e computa novas rotas para os veículos em que a região do acidente está contida na sua rota atual. O SPARTAN faz uso do sistema OBD 2 (*On Board Diagnostic 2*) para detectar a ocorrência de um acidente, para disseminar a mensagem de alerta dentro da AoI é utilizado uma versão para ambientes urbanos do protocolo de disseminação de dados ADD (*A Data Dissemination Solution for Highly Dynamic Environments*) [30]. Por fim, o SPARTAN faz uso dos algoritmos *Dijkstra*, *A** e *Probabilistic k-Shortest Paths* para computar novas rotas para os veículos evitarem a área congestionada ou que possivelmente ficará congestionada em um futuro próximo.

Problema 1 *Considere uma VANET representada por um grafo orientado com pesos, $G = (V, E)$, onde o conjunto finito $V = \{v_1, v_2, \dots\}$ corresponde ao conjunto de interseções (vértices), enquanto que o conjunto finito $E = \{e_1, e_2, \dots\}$ corresponde ao conjunto de segmentos de estradas (arestas) e $W = \{w_i, w_2, \dots\}$ corresponde ao conjunto de pesos para cada aresta $e \in E^2$. Seja $N = \{n_1, n_2, \dots, n_j\}$ o conjunto de veículos (nós) e $R = \{r_1, r_2, \dots, r_j\}$ o conjunto de rotas para cada veículo em N , onde cada rota em R corresponde a um subconjunto de arestas de E^2 . Na ocorrência de acidente de trânsito, os veículos envolvidos no acidente criam uma mensagem de alerta msg , a qual contém as informações sobre o local do acidente, e alertam os veículos que estão dentro da AoI. O local do acidente LA corresponde a um subconjunto de arestas de E^2 que serão afetadas pelo acidente. Portanto, quando um veículo n recebe a mensagem de alerta msg , o mesmo verifica se ao menos uma aresta de sua rota está contida na área afetada pelo acidente LA , ou seja, ele verifica se irá passar pela região do acidente. Então, caso necessário, o veículo n calcula uma rota alternativa r_{new} visando evitar a área afetada pelo acidente LA .*

¹AoI - é a área onde a mensagem de alerta deve ser propagada, essa área é definida pelo sistema de acordo com o cenário)

As seções a seguir são organizadas da seguinte forma: Seção 4.1 descreve a arquitetura do SPARTAN; Seção 4.2 como é feita a detecção de acidentes; Seção 4.3 o processo para alertar os veículos dentro da AoI sobre o acidente; Seção 4.4 é descrito o mecanismo de controle para evitar o surgimento de congestionamentos no local do acidente. Além disso, na Seção 4.5 é apresentado a avaliação de desempenho do SPARTAN em relação as soluções da literatura. Por fim, na Seção 4.6 é apresentada uma breve conclusão sobre o capítulo.

4.1 Arquitetura SPARTAN

A arquitetura do SPARTAN é apresentada na Figura 4.1. O SPARTAN é composto por três módulos, sendo eles: (i) Detecção de acidentes, (ii) *Alerta de acidentes*, e (iii) Gerenciador de tráfego.

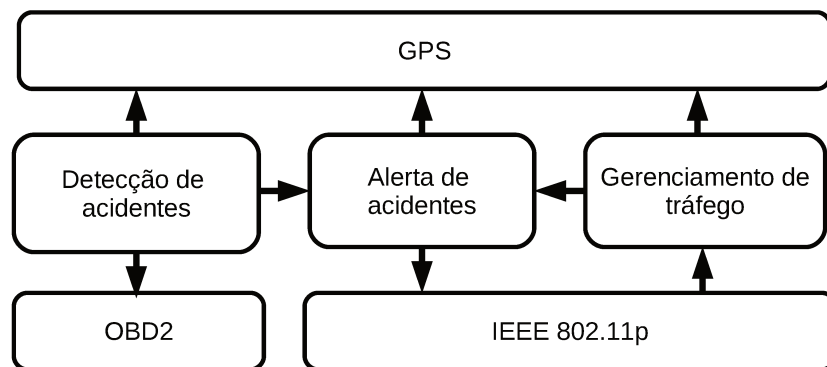


Figura 4.1: Arquitetura do SPARTAN

- **Detecção de acidentes:** responsável por identificar um acidente, coletar informações sobre o mesmo e criar uma mensagem de alerta para notificar os veículos dentro da AoI. Além disso, o módulo também é responsável por encaminhar a mensagem de alerta criada para o módulo de *Alerta de acidentes*.
- **Alerta de acidentes:** recebe a mensagem criada pelo módulo de *Detecção de acidentes* e dissemina a mesma para os veículos que estão dentro da AoI. Para isso, é utilizado uma versão para ambientes urbanos do protocolo de disseminação de dados ADD (*A Data Dissemination Solution for Highly Dynamic Environments*) [30]. O ADD realiza a disseminação em ambientes com diversas condições de tráfego e maximiza a entrega das mensagens de alerta para os veículos dentro da AoI, com baixo *overhead*, pequenos atrasos e baixo número de colisões.
- **Gerenciamento de tráfego:** o módulo é responsável por evitar que surja um congestionamento na região do acidente. Portanto, ele é subdividido em duas partes. A primeira é a *tomada de decisão em tempo real*, a qual faz a verificação se há necessidade da alteração da rota do veículo, para isso, o módulo verifica se a rota atual de veículo irá passar pelo local do acidente e, caso positivo a segunda parte do

módulo *alteração de rota* é notificada. Caso contrário, o veículo permanece com sua rota atual. A segunda parte do módulo *alteração de rota* implementa os algoritmos *Dijkstra*, *A** e *Probabilistic k-Shortest Paths* para computar novas rotas para os veículos evitarem a área congestionada ou que possivelmente ficará congestionada em um futuro próximo.

A Figura 4.2 apresenta o fluxo de operação do SPARTAN para detectar acidentes, alertar e rotear os veículos. A descrição do fluxo de operação do SPARTAN é apresentada em conjunto com a descrição dos módulos *Deteção de Acidentes*, *Alerta de Acidentes* e *Gerenciamento de tráfego* apresentados em detalhes nas sub-seções 4.2, 4.3 e 4.4 respectivamente.

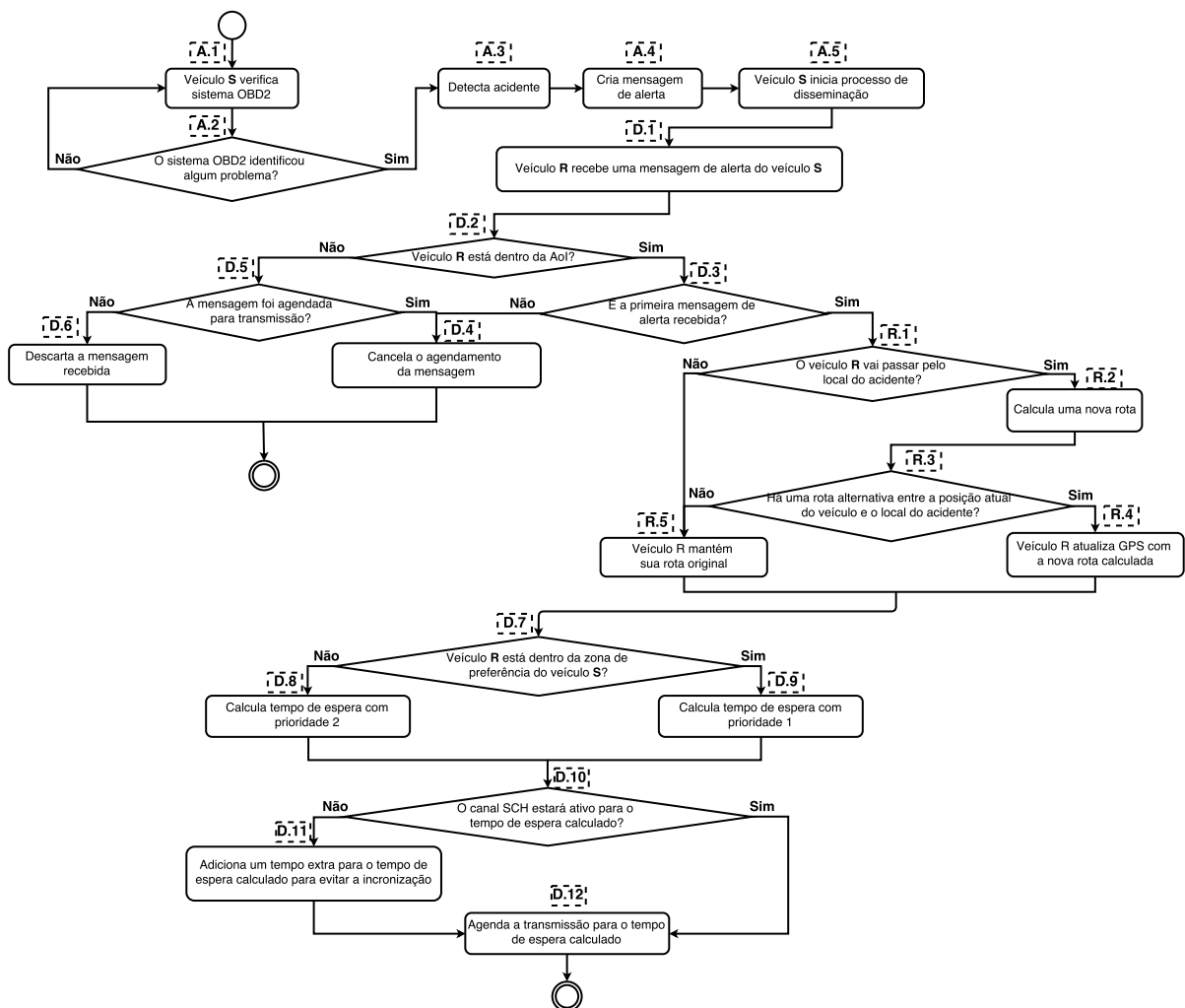


Figura 4.2: Fluxograma de atividades executadas pelo SPARTAN: detecção de acidentes, alerta de acidentes e gerenciamento de tráfego.

4.2 Detecção de Acidentes

Para a detecção de acidentes, o SPARTAN faz uso do sistema OBD 2 (*On Board Diagnostic 2*). O OBD2 já é implantado em todos os veículos fabricados no Brasil após o ano de 2012.

O sistema OBD 2 fornece informações sobre o veículo em tempo real, como por exemplo, velocidade, estado de *air bags* e danos no motor. Portanto, assim que um acidente ocorre, o sistema OBD 2 provê informações sobre o mesmo em tempo real. Após a detecção do acidente, os veículos envolvidos no acidente coletam informações sobre o local do acidente e uma mensagem de alerta é criada (veja Definição 1) para alertar os veículos dentro da AoI sobre o acidente (Figura 4.2 *labela A.1, A.2, A.3 e A.4*). Por fim, o módulo de *detecção de acidentes* encaminha a mensagem de alerta criada para o módulo de *alerta de acidentes*.

Portanto, se inicia o processo de disseminação da mensagem de alerta (Figura 4.2 *label A.5*).

Definição 1 *A mensagem de alerta é uma mensagem de camada 2 transmitida via broadcast pelo veículo que detectou o acidente para os veículos próximos através de múltiplos saltos. Esta mensagem contém as coordenadas geográficas do local do acidente e, as possíveis áreas que podem ser afetadas pelo mesmo (vias adjacentes). O local do acidente e as áreas que possivelmente serão afetadas são obtidas partir do GPS do veículo. Além disso, a mensagem também contém seu próprio time-to-live definida pelo protocolo de disseminação.*

4.3 Alerta de Acidentes

Para realizar a mensagem de alerta de acidente para os veículos dentro da AoI é utilizado uma versão para ambientes urbanos do protocolo de disseminação de dados ADD (*A Data Dissemination Solution for Highly Dynamic Environments*) [30].

O protocolo ADD trata de dois principais desafios da disseminação de dados em redes veiculares, sendo eles: (i) a tempestades de *broadcast*, descrito na Seção 4.3.1 e (ii) o efeito de sincronização introduzido pelo padrão IEEE 802.11p, descrito na Seção 4.3.2.

4.3.1 Supressão Broadcast

Para evitar o problema da tempestade de *broadcast*, é proposto um mecanismo para supressão de *broadcast* baseado no conceito de *zona de preferência*. A zona de preferência (Veja a Definição 2) é apresentada na Figura 4.3. Por questões de simplicidade a área de comunicação é considerada um círculo, mas qualquer outra forma funciona para a solução proposta. A área de comunicação é decomposta em quatro quadrantes, e em cada um, uma subárea do quadrante é definida como uma zona de preferência.

Definição 2 *Zona de preferência é definida como a área em que os veículos que estão dentro dela são os mais propícios para continuar o processo de disseminação da mensagem de alerta. Em outras palavras, dentre todos os veículos que recebem a mensagem de alerta, a retransmissão de apenas um veículo dentro da cada zona de preferência é suficiente para disseminar as mensagens de maneira eficiente. Veículos localizados na zona de preferência são mais susceptíveis de espalhar a mensagem e alcançar um número maior de vizinhos que não foram capazes de receber a mensagem de alerta do retransmissor anterior [33].*

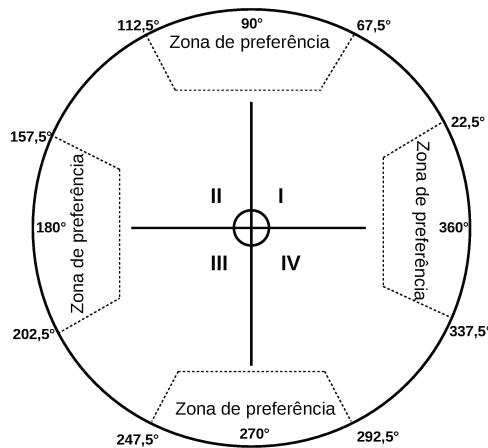


Figura 4.3: Zona de preferência

O ADD é um protocolo baseado em atraso, onde, quando um veículo recebe uma mensagem, o mesmo agenda a retransmissão da mensagem para um determinado instante de tempo. Portanto, o intervalo do recebimento da mensagem até a transmissão da mesma é definido como atraso. Dessa forma, para o cálculo do atraso, o ADD utiliza zona de preferência e a distância euclidiana entre o veículo emissor e o veículo. Além disso, o mecanismo de supressão de *broadcast* implementa um método para evitar a transmissões redundantes, ou seja, se um veículo agenda a retransmissão de uma mensagem para um determinado instante e, antes de retransmitir a mensagem ele recebe uma cópia da mensagem que já foi agendada, o veículo faz o cancelamento da mensagem agendada (Figura 4.2 *label D.4*), pois o recebimento de uma cópia da mensagem agendada para retransmissão informa que outro veículo já realizou a retransmissão da mensagem, portanto o veículo não precisa retransmitir a mesma mensagem.

A zona de preferência minimiza o problema de congestionamento de transmissões. Isto é, os veículos no interior da zona de preferência transmitem a mensagem de alerta com um atraso inferior e cancelam as transmissões da mesma mensagem de veículos que se encontram fora das zonas de preferências. Entre os veículos dentro da zona de preferência, aquele mais distante do veículo emissor irá transmitir primeiro e cancelar as transmissões desnecessárias (redundância) dos outros veículos dentro da mesma zona de preferência. Em cenários esparsos, quando não há nenhum veículo na zona de preferência, o veículo mais distante do emissor em cada quadrante retransmitirá a mensagem de alerta. A zona de preferência também evita que veículos próximos uns dos outros e que pertencem a diferentes quadrantes retransmitem a mesma mensagem alcançando uma área semelhante (redundância desnecessária).

Durante o processo de disseminação, os veículos dentro de cada zona de preferência (veja Figura 4.3) são a melhor escolha para continuar propagando a mensagem de alerta, reduzindo o número de transmissões necessárias. Ao receber uma mensagem (ver Figura 4.2 *label D.1*), cada veículo verifica se está dentro da AoI (Figura 4.2 *label D.2*). Se não for o caso, o veículo descarta a mensagem recebida. Caso contrário, ele verifica se é a primeira vez que recebe a mensagem (ver Figura 4.2 *label D.3*). Se assim for, o veículo

calcula o tempo de espera para continuar o processo de disseminação com base zona de preferência e sua posição (ver D.7, D.8 e D.9 na Figura 4.2). Além disso, após o cálculo do tempo de espera, o veículo verifica se o canal SCH estará ativo durante a transmissão e caso necessário um tempo extra é adicionado para evitar o efeito de ressincronização (ver D.10, D.11 e D.12 na Figura 4.2).

O Algoritmo 1 mostra detalhes sobre a zona de preferência e como o tempo de espera é calculado. A seguir, é apresentado o mecanismo dessincronização proposto para calcular o novo tempo de espera e evitar a ressincronização causada pelo padrão IEE 802.11p.

Algorithm 1: Verifica se o veículo que recebeu a está dentro da zona de preferência e calcula o tempo de espera para a transmissão da mensagem

```

1 Input:  $(x_s, y_s)$  and  $(x_r, y_r)$  // Coordenadas do veículo emissor e receptor
2 Output: Delay // Calcula o tempo de espera para a retransmissão da mensagem.
   // a função atan2 é uma variante da função arctangente.
3  $angle \leftarrow \text{atan2}(y_s - y_r, x_s - x_r)$ ;
4  $distToSender \leftarrow \sqrt{(x_s - x_r)^2 + (y_s - y_r)^2}$ ;
5  $defaultDelay \leftarrow 0.01 \times (distToSender / communicationRadius)$ ;
6 if  $((angle \geq 67.5^\circ \text{ and } angle \leq 112.5^\circ) \parallel (angle \geq 157.5^\circ \text{ and } angle \leq 202.5^\circ) \parallel$ 
    $(angle \geq 247.5^\circ \text{ and } angle \leq 292.5^\circ) \parallel (angle \leq 22.5^\circ \parallel angle \geq 337.5^\circ))$  then
7   // Calcula tempo de espera com prioridade 1
    $Delay \leftarrow defaultDelay + \text{random}(0, 0.01)$ 
8 end
9 else
   // Calcula tempo de espera com prioridade2
10   $Delay \leftarrow defaultDelay + \text{random}(0.02, 0.05)$ 
11 end

```

4.3.2 Mecanismo de Dessincronização

O mecanismo proposto visa reduzir o problema de ressincronização introduzido pelo padrão IEEE 802.11p. O mecanismo utiliza como entrada o valor de atraso computado no módulo *alerta de acidentes* esperado a partir da disseminação de dados (ver Algoritmo 1) e, caso necessário, um atraso adicional T_d é calculado considerando o regime de troca de canais. Algoritmo 2 detalha como este método funciona.

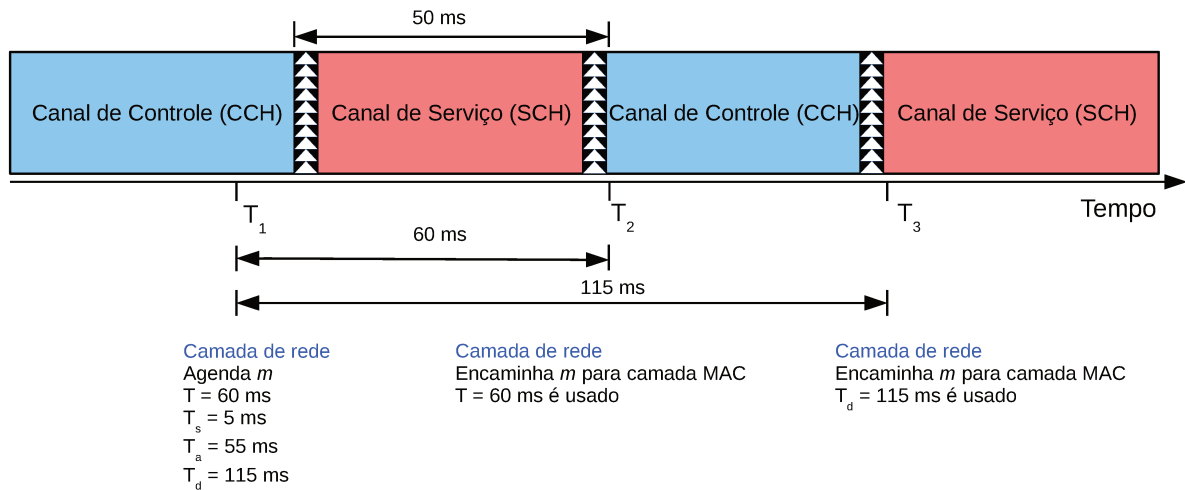


Figura 4.4: Exemplo de como o método de dessincronização funciona

A ideia principal do Algoritmo 2 é adicionar ao atraso inicial um tempo extra T_a antes de retransmitir a mensagem. Este tempo extra T_a (Linhas 5 e 13) é o tempo em que o canal CCH estará ativo novamente (no máximo 55 ms, contanto com o intervalo de 5 ms para a troca de canal CCH para SCH)[REF ADD], se a mensagem for imediatamente passada para a camada física, em seguida, ela aguardaria $Delay$ segundos, antes de ser transmitida pelo canal SCH. Por exemplo, na Figura 4.4, um veículo recebe uma mensagem no momento T_1 . Portanto, o algoritmo calcula um atraso $T = 60\text{ms}$ para retransmitir a mensagem. Note que, no tempo T_1 , o canal CCH está ativo e permanecerá por mais $t_s = 5\text{ms}$. Neste exemplo, o Algoritmo 2 calcula $T_a = 55\text{ms}$, porque, se a camada de rede enviasse a mensagem para a camada MAC no tempo T_1 e a camada MAC esperaria $T = 60\text{ms}$ antes de transmitir a mensagem no canal SCH, então a mensagem vai notar o canal CCH ativo para 55ms. Portanto, o algoritmo agenda a retransmissão da mensagem com um atraso de $T_d = T + T_a$. Por fim, no tempo de T_3 , a camada de rede encaminha a mensagem para a camada MAC, para que possa ser imediatamente transmitida. Ao utilizar o método de dessincronização, é possível garantir que, quando a camada de rede encaminha a mensagem para a camada MAC, o canal SCH estará ativo.

Algorithm 2: Método de dessincronização

```

Input :  $T$  // tempo de espera recebido para transmitir a mensagem
1  $T_c$  // tempo de cada canal (50ms)
2  $T_s$  // tempo restante para a troca de canal

3 if CCH está ativo then
4    $cch\_cycles = \lfloor \frac{T}{T_c} \rfloor$ ;
5    $T_a \leftarrow T_s + (cch\_cycles \times T_c)$ ;
6    $T_d \leftarrow T + T_a$ ;
7 end
8 else
9    $T_d \leftarrow T$ ;
10   $T_{tmp} \leftarrow T - T_s$ ;
11  if  $T_{tmp} > 0$  then
12     $chh\_cycles \leftarrow \lceil \frac{T_{tmp}}{T_c} \rceil$ ;
13     $T_a \leftarrow cch\_cycles \times T_c$ ;
14     $T_d \leftarrow T + T_a$ ;
15  end
16 end
17 return  $T_d$ 

```

4.4 Gerenciamento do Tráfego

Para evitar o surgimento de um congestionamento no local do acidente LA , é necessário alterar a rota dos veículos $n_j \in N$ em que ao menos uma via e_i da sua rota r_j esteja contida na área do acidente $e_i \in r_j \mid e_i \in LA$. Para isso, nessa seção, é apresentado os mecanismos *tomada de decisão* e *alteração de rotas* utilizados pelo módulo de *Gerenciamento do tráfego*.

Primeiramente, quando um veículo dentro da AoI recebe uma mensagem de alerta, o mesmo executa o mecanismo de *Tomada de decisão*, onde é verificado se a rota atual passa pela área do acidente $r_j \subset LA$ (via que contém o acidente e vias adjacentes), baseado nas informações do GPS integrado ao veículo (ver *label R.1* na Figura 4.2), caso a rota atual do veículo não passa pelo local do acidente $r_j \not\subset LA$, o mesmo mantém a rota atual

(*label R.5* na Figura 4.2). Caso contrário, uma nova rota é calculada r_{new} . Entretanto, se não houver uma rota alternativa, a rota original é mantida. Porém, caso exista pelo menos uma nova alternativa, o veículo atualiza sua rota para evitar que o mesmo passe pelo local do acidente $r_j \leftarrow r_{new}$ (ver *label R.2*, *label R.3* e *label R.4* na Figura 4.2).

O cálculo da nova rota é feito utilizando um algoritmo de roteamento. Entretanto, antes de calcular uma nova rota, o SPARTAN remove o conjunto de arestas LA para evitar que a nova rota utilize arestas do conjunto LA que serão afetadas pelo acidente. Além disso, o SPARTAN implementa três algoritmos de roteamento, sendo eles: (i) Dijkstra; (ii) A* e; (iii) *Probabilistic k-Shortest Paths*. Os três algoritmos implementados no SPARTAN são algoritmos de menor caminho. Entretanto, cada algoritmo faz uso de métricas diferentes para calcular uma nova rota.

- **Dijkstra:** encontra o caminho com o menor custo (geralmente refere-se ao caminho mais curto) da posição corrente do veículo até o destino. Dijkstra é um dos algoritmos ótimos baseados no método de marcação. No entanto, quando o tráfego de veículos aumenta, a probabilidade de acontecer congestionamentos em uma nova área é maior, uma vez que o algoritmo usa somente o percurso do menor caminho para rotear os veículos, de modo que, em cenários densos muitos veículos são roteados pela a mesma rota.
- **A*:** é uma variante do algoritmo Dijkstra mas utiliza uma função heurística, em vez de um mecanismo de busca ótima. Assim, o algoritmo de A* restringe o espaço de busca e reduz o tempo de processamento. Em uma aplicação de tráfego, o espaço de busca é limitado a uma zona onde ocorre congestionamento [23]. A* utiliza a distância direta entre o local atual e do destino como uma função heurística.
- **Probabilistic k-Shortest Paths (PkSP):** calcula um conjunto de k -novas-rotas de menor caminho do ponto atual do veículo até seu destino, em seguida, utiliza o algoritmo probabilístico de Boltzmann [16] para selecionar uma rota do conjunto de k -novas-rotas previamente calculado. Selecionar uma rota em um conjunto de k -novas-rotas reduz a probabilidade de criar um congestionamento em um novo local, devido ao fato de não selecionar sempre a mesma rota.

4.5 Avaliação de desempenho

Esta seção descreve a metodologia utilizada nas simulações e nas análises dos resultados. Dessa forma, para avaliarmos a eficiência do SPARTAN. Primeiramente, foi avaliado o módulo de alerta de acidentes e em seguida o módulo de *gerenciamento do tráfego*. A Seção 4.5.1 apresenta os resultados para o protocolo de disseminação de mensagens de alerta proposto em comparação com os quatros protocolos bem conhecidos da literatura – Flooding, AID [2], DBRS [15] e DRIVE [34]. Por fim, a Seção 4.5.2, apresenta os resultados do SPARTAN para controle de congestionamentos de veículos. O SPARTAN foi comparado com três soluções bem conhecidas da literatura – “*With routing*” [7], DSP e RkSP [26]. Além disso, o SPARTAN também foi comparado com a abordagem tradi-

onal de trânsito, ou seja, o trace com a mobilidade de veículos original. A abordagem tradicional é nomeada de OVMT (*Original Vehicular Mobility Trace*).

As simulações foram realizadas utilizando os simuladores de rede OMNeT++ [25] e o de tráfego SUMO [5]. Os aspectos da rede veicular foram simulados usando o *framework* Veins [28]. Adicionalmente, as avaliação foram feitas utilizando um cenário realista obtido com o OpenStreetMap². O cenário avaliado possui uma área de 1 km² referente ao bairro de Manhattan, Nova York, Estados Unidos. Essa área possui diversas quadras, compostos por ruas e avenidas com fluxo de veículos em ambas as direções, conforme mostra a Figura 4.5.

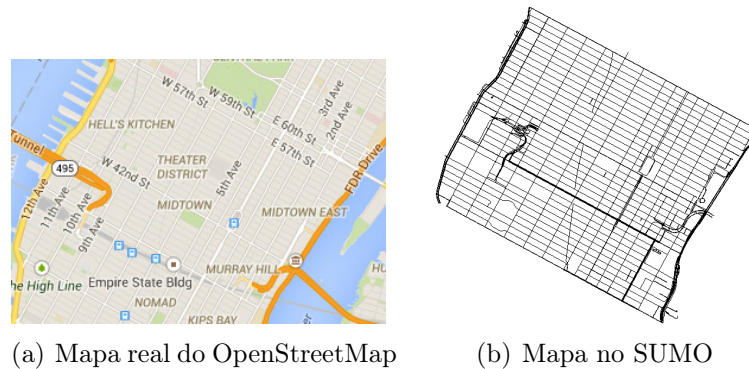


Figura 4.5: Mapa de Manhattam.

Os principais parâmetros foram definidos da seguinte forma: a taxa de bits para 18 Mbit/s na camada MAC; a potência de transmissão foi definida para a 1,6mW, resultando em um alcance de comunicação de cerca de 250m sob um modelo de propagação *two-ray ground* [28]. Finalmente, para todos os resultados apresentados, os pontos nos gráficos representam a média de 33 repetições com um intervalo de confiança de 95%. A Tabela 4.1 mostra os principais parâmetros de simulação utilizados.

Tabela 4.1: Parâmetros de simulação SPARTAN

Parâmetros	Valores
Transmissão	1.6 mW
Alcance	250 m
Taxa de bit	18 Mbit/s
Cenário	Manhattan grid
Tamanho do cenário	1 km ²
Área de interesse	1km
Número de execuções	33
Intervalo de confiança	95%

²<http://www.openstreetmap.com>

4.5.1 SPARTAN vs Soluções de disseminação

Nesta subseção, são apresentados os resultados do ADD, protocolo de disseminação implementado no SPARTAN. Para avaliar os protocolos, uma RSU foi inserida ao centro do cenário, e quando a simulação atinge um estado estável, a mesma gera 100 mensagens de 2048 bytes, que são disseminadas a uma taxa de 1,5 Mbit/s para todos os veículos dentro da AoI. Além disso, o desempenho do ADD foi comparado com AID, DBRS, DRIVE e *Flooding*. Portanto, para analisar o impacto das soluções em diversas condições de tráfego de veículos, as seguintes densidades de veículos foram utilizadas nas simulações: 300, 400, 500, 600 e 700 veículos/km². Por fim, as métricas utilizadas para avaliação dos protocolos foram:

- **Cobertura:** a percentagem de mensagens geradas pela RSU que são efetivamente entregues aos destinatários pretendidos. Espera-se que os protocolos de disseminação devem alcançar uma taxa de entrega de 100%.
- **Mensagens transmitidas:** número total de mensagens transmitidas por todos os veículos na rede durante o processo de disseminação. Um elevado número de transmissões de mensagens é uma forte indicação de que as mensagens estão sendo disseminadas de forma redundante, o que pode resultar no problema de tempestade de *broadcast*.
- **Atraso:** o tempo médio que leva para disseminar as mensagens a partir da RSU para todos os veículos dentro da área de interesse. Um atraso baixo é de particular interesse para aplicações de tempo estrito, como a disseminação de mensagem de alerta.
- **Colisões:** o número médio de colisões de pacotes na camada MAC por veículo para disseminar todas as mensagens. Um elevado número de colisões indicam que um dado protocolo não é capaz de evitar o problema de tempestade de *broadcast*.

A Figura 4.6 mostra todos os resultados para a disseminação de dados para o cenário avaliado. Em particular, a Figura 4.6(a) apresenta os resultados para a métrica de Cobertura, onde pode ser visto que, para densidades baixas, o *Flooding* é o único protocolo que possui uma cobertura maior do que 95% na AoI. Isto se deve ao fato de que o *Flooding*, ao receber uma mensagem retransmite a mensagem para todos os veículos vizinhos, alcançando assim uma melhor cobertura. Para os outros protocolos, apenas o ADD atinge uma cobertura maior que 90% em densidades baixas. Entretanto, conforme a densidade aumenta, o tráfego de dados na rede também aumenta (ver Figura 4.6(b)). Este aumento no tráfego de dados na rede faz com que ocorra maior número de colisões de pacotes, afetando diretamente a cobertura dos protocolos. Como o ADD utiliza um mecanismo para supressão de *broadcast* e um mecanismo de dessincronização para minimizar as colisões de pacotes, ele é o único que consegue manter uma cobertura acima de 90%.

A Figura 4.6(b) apresenta o número total de mensagens transmitidas. Como esperado, *Flooding* é o protocolo com maior sobrecarga, pelo motivo que no *Flooding* todos os veículos retransmitem a mensagem, resultando em retransmissões redundantes. Já o

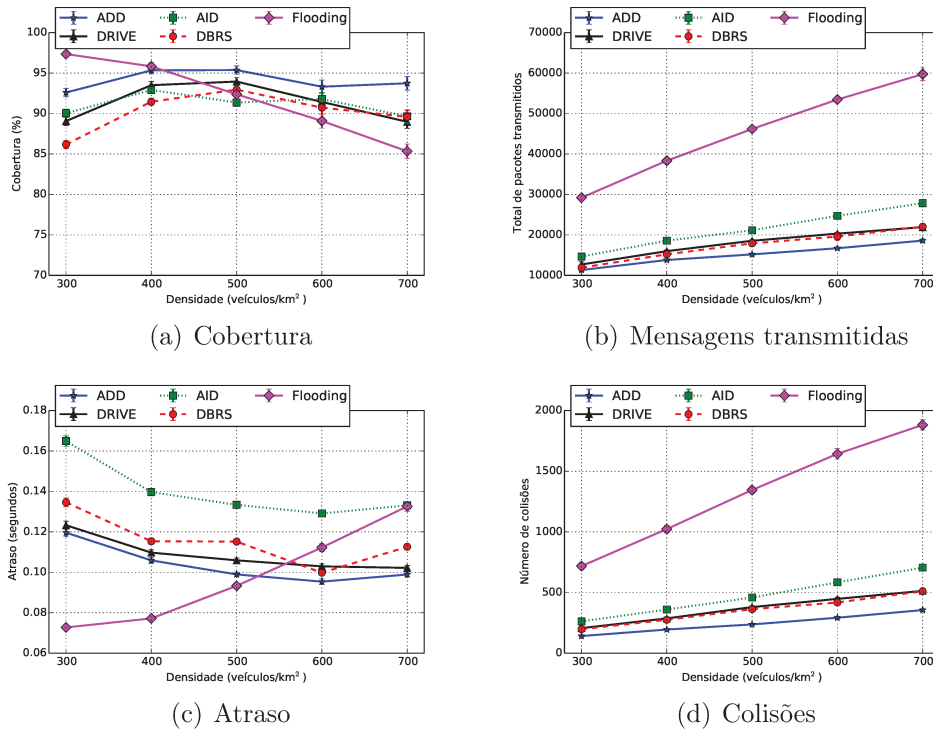


Figura 4.6: Resultado algoritmo de disseminação de dados para cenário urbano

ADD utiliza o mecanismo de supressão de *broadcast* proposto e gera aproximadamente 68% mensagens a menos quando comparado ao *Flooding*. Além disso, ADD dissemina menos mensagens do que AID e DBRS. Portanto, como pode ser observado, a solução proposta não desperdiça a largura de banda com retransmissões desnecessárias.

A Figura 4.6(c) mostra o atraso médio para disseminar as mensagens para os veículos dentro da AoI. Como pode ser observado, para densidades mais baixas, o *Flooding* tem um atraso menor, seguido pelo ADD. No entanto, com o aumento da densidade, o ADD leva menos tempo para disseminar a mensagem para todos os veículos na AoI, em virtude do mecanismo para minimizar o número de colisões de pacotes, evitando retransmissões de pacotes. Isso demonstra que ADD é uma boa solução para aplicações com requisitos de tempo rigorosos, como a disseminação de mensagens de alerta de acidente.

Por fim, a Figura 4.6(d) mostra o número médio de colisões na camada MAC durante a disseminação. O *Flooding* tem o maior número de colisões, o qual aumenta conforme o aumento da densidade. Isso mostra que o *Flooding* não evita o problema de *tempestade de broadcast*, especialmente em densidades mais elevadas. O ADD tem o menor número de colisões para todas as densidades, porque ele implementa o mecanismo dessincronização para minimizar as colisões. Comparado ao *Flooding*, ADD pode reduzir o número de colisões em cerca de 80%. Quando comparado ao AID, DBRS e DRIVE, ADD reduz colisões aproximadamente em 45%, 30% e 30%, respectivamente.

4.5.2 SPARTAN *vs* Soluções para Controle de Congestionamentos

Esta subseção apresenta a avaliação do SPARTAN em comparação as soluções para controle de congestionamentos. Mais especificamente, foi avaliado um cenário em que um veículo escolhido aleatoriamente bloqueia um cruzamento, impedindo o fluxo de veículos em ambas as direções. Por fim, as soluções foram avaliadas com a densidade de 700 veículos/km²

O SPARTAN foi comparado com O OVMT (trace com a mobilidade de veículos original) e com três soluções da literatura: DSP [26], RkSP [26] e “*With routing* [7]”. Todas as soluções foram avaliadas e comparadas utilizando as seguintes métricas:

- **Tempo de viagem:** é a média do tempo que todos os veículos levaram para chegarem ao destino;
- **Distância percorrida:** é média da distância percorrida por todos os veículos até alcançarem seu destino;
- **Tempo de congestionamento:** é a média do tempo que os veículos permanecem parados no congestionamento;
- **Velocidade média:** é a média da velocidade de todos os veículos durante toda a simulação;
- **Consumo de combustível:** é a média do consumo de combustível de todos os veículos na simulação;
- **Emissão de CO:** é a média da emissão de CO de todos os veículos na simulação.

O desempenho das soluções DSP, RkSP e “*With routing*” são dependentes dos parâmetros *intervalo de roteamento* e *k-menores caminhos*. Para determinar os melhores valores, foram realizados vários testes com diferentes *intervalos de roteamento*. Para o cenário avaliado usamos o melhor *intervalo de roteamento* obtido nas avaliações, sendo o valor de 150 segundos. Além disso, as soluções RkSP e “*With routing*” são dependentes do parâmetro *k* para o cálculo dos *k-menores caminhos*. Sendo assim, também foram realizados vários testes no cenário avaliado. O melhor ajuste desse parâmetro foi o valor de $k = 5$ para ambas as soluções. Além disso, o SPARTAN implementa os algoritmos Dijkstra, A* e PkSP para calcular as rotas dos veículos, dessa forma diversos testes foram realizados e o algoritmo PkSP com o valor de $k = 5$ apresentou os melhores resultados, devido ao fato de redistribuir melhor o fluxo de veículos, evitando assim congestionamentos em outros pontos. Por fim, é importante salientar que os valores utilizados para os parâmetros das soluções DSP, RkSP e “*With routing*”, foram obtidos de [26, 7].

A Figura 4.7 mostra todos os resultados para as métricas avaliadas. Em particular, a Figura 4.7(a) apresenta os resultados para o tempo médio de viagem. Conforme esperado, o OVMT apresenta o pior tempo de viagem, pelo motivo que não tem controle de congestionamentos. Como o acidente bloqueia um cruzamento, um congestionamento é iniciado e muitos veículos ficam presos no congestionamento. No OVMT, os veículos ficam preso

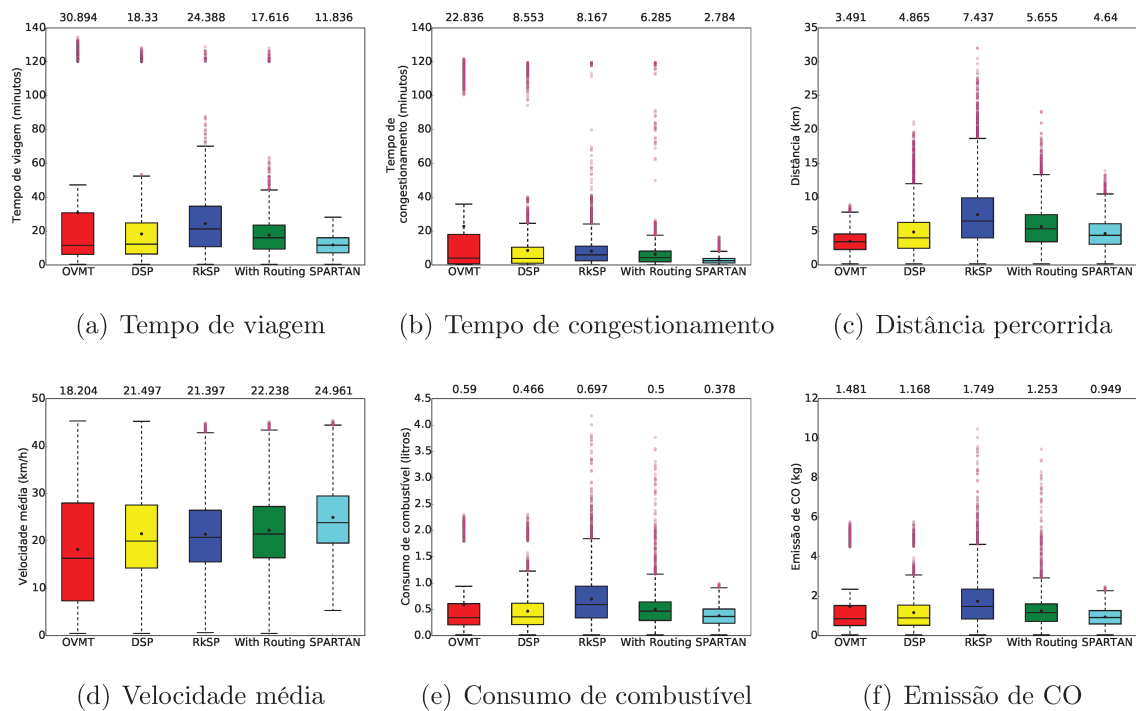


Figura 4.7: Avaliação das Soluções para Controle de Congestionamentos

em um congestionamento em média 22,83 minutos, o qual é aproximadamente 74% de todo o tempo médio de viagem. As soluções DSP, RkSP e “*With routing*” não disseminam nenhuma informação sobre o acidente, consequentemente o congestionamento só será detectado nos intervalos de roteamento dos veículos. É importante notar que sem ter a informação sobre o acidente, muitos veículos entram na área de congestionamento durante o intervalo de roteamento (veja Figura 4.7(a) e Figura 4.7(b)). Mesmo assim, as soluções DSP, RkSP e “*With routing*” são capazes de reduzir o tempo de viagem e o tempo de congestionamento quando comparados com o OVMT. O DSP reduz o tempo de viagem em 40%, enquanto RkSP e “*With routing*” reduzem o tempo de viagem em 21% e 43% respectivamente. Como consequência, o tempo de congestionamento é reduzido em 62% para o DSP, em 64% para o RkSP e em 72% para o “*With routing*” quando comparados com o OVMT. Entretanto, os veículos ainda desperdiçam 46%, 33% e 35% de todo o tempo de viagem em um congestionamento, para DSP, RkSP e “*With routing*”, respectivamente. Além disso, essas soluções aumentam a distância percorrida em 39% para o DSP, 112% para o RkSP e 61% para o “*With routing*” (veja Figura 4.7(c)). Observa-se que a grande distância percorrida pelo RkSP é devido à seleção de rota aleatória, onde muitos veículos podem selecionar rotas mais longas para evitar o congestionamento. Diferentemente da solução “*With routing*”, que usa uma seleção probabilística para evitar esse problema.

Como o SPARTAN é possível controlar a iminência de um congestionamento, logo quando um acidente ocorre. Assim, os veículos não entram nas ruas bloqueadas pelo acidente, de modo que eles não ficam presos no congestionamento (veja os valores na Figura 4.7(a) e Figura 4.7(b)). Além disso, o SPARTAN reduz o tempo de viagem em 62%, quando comparado com o OVMT e 35%, 51% e 32%, quando comparado com DSP, RkSP e “*With routing*”, respectivamente. Por fim, o SPARTAN reduz o tempo de

congestionamento em 88%, quando comparado com o OVMT e os veículos gastam apenas 22% de todo o tempo de viagem no congestionamento. Do mesmo modo, como nas outras soluções de re-roteamento de veículos, o SPARTAN também aumenta a distância percorrida, porém, tem o menor aumento nas comparações, aproximadamente 30%. Estes resultados mostram que o SPARTAN melhora a condição de tráfego e supera em diversos aspectos as soluções da literatura.

Os resultados de consumo de combustível estão diretamente relacionados com o tempo de viagem, tempo de congestionamento e distância percorrida. A Figura 4.7(e) mostra os resultados de consumo de combustível. Como pode ser observado, o OVMT tem um consumo médio de combustível de 0,59 litros. DSP e “*With routing*” diminuem o consumo de combustível em 22% e 15%, enquanto RkSP aumenta o consumo de combustível em 18%. Esse aumento de RkSP é devido à maior distância média percorrida. O SPARTAN também diminui o consumo de combustível, uma redução de 36%, quando comparado com OVMT. Essa redução é maior, por causa da menor distância percorrida e menor tempo que os veículos passam no congestionamento.

A Figura 4.7(f) mostra os resultados para as emissões de CO. Como esperado, as emissões de CO estão relacionadas com o consumo de combustível. Portanto, o DSP e “*With routing*” diminuem a emissão de CO em 21% e 14%, respectivamente, quando comparado com OVMT, enquanto RkSP aumenta as emissões em 18%. Por fim, o SPARTAN diminui as emissões em 36%.

A melhoria da condição de tráfego pode ser vista na Figura 4.7(d), onde o SPARTAN tem a maior velocidade média. Tal resultado deve-se à disseminação da mensagem de alerta sobre o acidente para os veículos e ao mecanismo eficiente de controle congestionamento. O SPARTAN aumenta a velocidade média em 37%, 16%, 16% e 12%, quando comparado com OVMT, DSP, RkSP e “*With routing*”, respectivamente.

4.6 Resumo Conclusivo

Neste capítulo, foi apresentado o SPARTAN, um sistema de transporte inteligente para controle de congestionamentos causados por acidentes de trânsito. Dessa forma, o objetivo principal do SPARTAN é realizar a detecção de acidentes e alertar os veículos dentro da AoI para que os mesmos possam evitar o local com eminência de congestionamento.

Para disseminar a mensagem de alerta na AoI, o SPARTAN implementa a versão para ambientes urbanos do protocolo de disseminação de dados ADD. O protocolo de disseminação de alertas implementado trata os problemas de *tempestade de broadcast* e o *efeito de sincronização*. Os resultados de simulações mostram que o ADD é adequado e eficiente para disseminar mensagens de alerta.

Por fim, para evitar o surgimento de congestionamentos, o SPARTAN faz uso dos mecanismos *tomada de decisão* e *alteração de rotas* implementados no módulo de *Controle de congestionamento*. Os resultados de simulações mostram a eficiência do mecanismo de controle de tráfego. O módulo *controle de congestionamento* contribue diretamente para um menor tempo de viagem, menor tempo em que os veículos ficam presos no congestionamento e aumenta a velocidade média dos veículos.

Entretanto, apesar dos bons resultados alcançados pelo SPARTAN, o sistema consegue controlar apenas os congestionamentos causados por acidentes. Portanto, caso exista congestionamentos mas não há nenhum acidente no cenário, os veículos mantêm suas rotas originais até seu destino, ou seja, o sistema torna-se ineficiente para congestionamentos causados pelo grande fluxo de veículos. Tendo essa necessidade em mente, foi desenvolvido um sistema para detecção de congestionamentos, o qual possa detectar os congestionamentos e controla-los de forma eficiente, o sistema proposto é apresentado no Capítulo 5.

Capítulo 5

Detecção e Controle de Congestionamentos

Neste capítulo, é apresentado COLOSSUS (**C**ongestion detection, **L**evel **O**f Service cla**S**sification and re-ro**U**ting **S**ystem), um sistema para detecção, classificação e controle de congestionamentos de veículos. O objetivo principal do COLOSSUS é detectar a iminência de congestionamento, classificar a condição de tráfego em todas as vias e, por fim, realizar o roteamento dos veículos para distribuir melhor o fluxo de veículos. Para isso, o COLOSSUS faz a coleta periódica das informações dos veículos (id, velocidade, posição e rota) e, então, periodicamente faz a classificação de todas as vias ¹. Além disso, após classificar todas as vias, o COLOSSUS define uma área de roteamento (veja Definição 3), em seguida realiza o roteamento dos veículos que estão dentro dessa área e passarão por alguma via congestionada.

Definição 3 *Área de roteamento (RA - Routing area) é definida como uma área onde os veículos que estão localizados dentro dela têm maior urgência de serem roteados, ou seja, os veículos que estão mais próximos do congestionamento e têm maior probabilidade de passar pela via congestionada. O objetivo da utilização a área de roteamento é evitar o roteamento desnecessário de veículos, como veículos distantes do congestionamento. Por fim, essa área de roteamento é definida pelo sistema.*

O restante do capítulo é dividido da seguinte forma: a arquitetura do COLOSSUS é descrito na Seção 5.1. A Seção 5.2 apresenta a rede e o modelo de comunicação. A Seção 5.3 descreve o método utilizado para classificar a condição de tráfego e detectar as vias congestionadas. A Seção 5.4 apresenta os algoritmos de roteamento de veículos propostos e utilizados no COLOSSUS, onde os algoritmos são responsáveis por realizar o balanceamento do tráfego de veículos para minimizar e controlar congestionamentos de acordo com a condição de tráfego em cada via. Por fim, a avaliação do COLOSSUS em comparação com os sistemas apresentados na literatura é feita na Seção 5.5.

¹o intervalo para classificar todas as vias é definido como *intervalo de classificação - IC*

5.1 Arquitetura do COLOSSUS

O COLOSSUS é dividido em quatro módulos, sendo eles: (i) Coleta de informações; (ii) Classificação do tráfego; (iii) Análise de rota e; (iv) Sugestão de rota. A Figura 5.1 descreve a arquitetura do COLOSSUS e como cada módulo interage entre si.

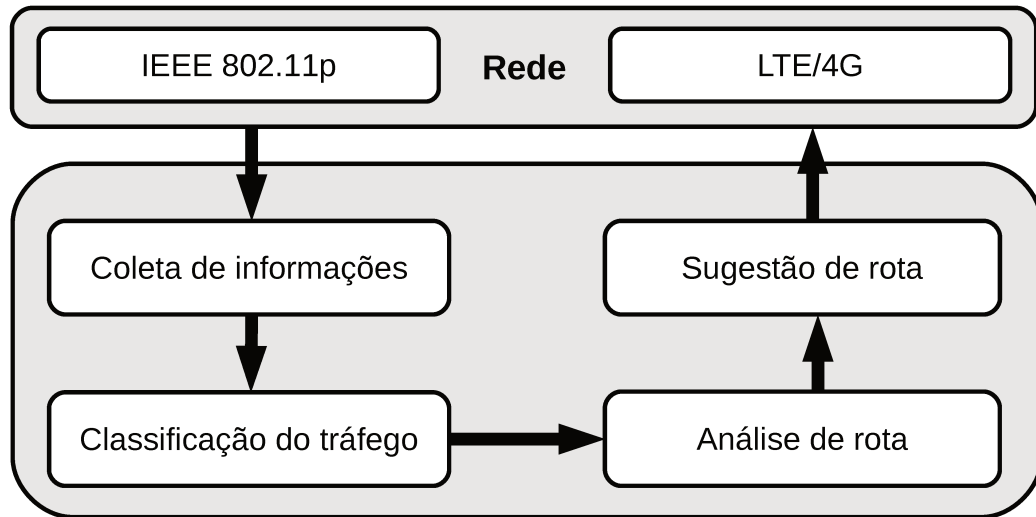


Figura 5.1: Arquitetura do COLOSSUS.

1. **Coleta de informações:** módulo responsável por receber as informações dos veículos (ID, posição atual, rota e velocidade) e manter os dados sobre o tráfego (velocidade média e densidade) atualizados. No módulo *coleta de informações* as RSUs dispostas no cenário são responsáveis por receber as informações dos veículos, as quais são enviadas periodicamente através da rede, a qual pode implementar não só o protocolo IEEE 802.11p, mas também LTE/4G;
2. **Classificação do tráfego:** recebe as informações atualizadas obtidas pelo módulo de *coleta de informações* e periodicamente faz a *classificação do tráfego* de todas as vias. O intervalo para realizar a classificação é definido pelo COLOSSUS;
3. **Análise de rota:** após a *classificação do tráfego* de todas as vias, o módulo *análise de rota* analisa quais veículos devem ser re-roteados. Para isso, para cada via congestionada o COLOSSUS define uma área de roteamento. Além disso, é verificado se o veículo está dentro de uma área de roteamento e se ao menos uma via congestionada está contida na rota do veículo. Caso essa condição for satisfeita, o veículo precisa ser roteado, caso contrário o veículo permanece com sua rota original;
4. **Sugestão de rota:** calcula uma nova rota para os veículos selecionados no módulo *análise de rota*. Ademais, dois algoritmos para roteamento são implementados,

sendo o primeiro uma abordagem de roteamento de melhor caminho, o qual calcula os k menores caminhos e faz o balanceamento do tráfego nos k caminhos calculados. O segundo algoritmo, apresenta uma abordagem de roteamento cooperativa com o objetivo de melhorar o fluxo dos veículos, assim, não são calculadas as melhores rotas para cada veículo, mas sim as melhores rotas para manter um tráfego contínuo. Por fim, a nova rota computada é enviada para os veículos através da rede utilizando o protocolo IEEE 802.11p ou LTE/4G pelas RSUs.

5.2 Representação da Rede e modelo de Comunicação

A rede é representada por um grafo ponderado e dirigido $G = (V, E)$, onde os conjuntos finitos $V = \{v_1, v_2, \dots\}$ representa o conjunto de cruzamentos do cenário (vértices), enquanto o conjunto $E = \{e_1, e_2, \dots\}$ corresponde ao conjunto de vias do cenário (arestas). Além disso, $W = \{w_1, w_2, \dots\}$ é o conjunto de pesos de cada via $e \in E^2$. Seja $N = \{n_1, n_2, \dots, n_j\}$ o conjunto de veículos (nós) na rede e $R = \{r_1, r_2, \dots, r_j\}$ o conjunto de rotas para cada veículo $n_j \in N$.

Todos os veículos n_j na rede, enviam uma mensagem $msg^j \leftarrow \{\text{id}, \text{posição}^j, \text{rota}^j, \text{velocidade}^j\}$ para o COLOSSUS utilizando as RSUs. Assim que novas informações estão disponíveis, o COLOSSUS calcula a velocidade média s_i e a densidade d_i de cada via $e \in E^2$. Depois disso, atualiza o peso de cada via, $w_i \leftarrow (s_i, d_i, ms_i)$, onde ms_i é a velocidade máxima da via e_i . Além disso, baseado nos pesos de cada via, o COLOSSUS classifica o nível de congestionamento de todo conjunto E periodicamente e cria um conjunto das vias congestionadas $C = \{c_1, c_2, \dots, c_i\}$, onde C é um subconjunto de E^2 . Por fim, para cada via congestionada c_i é criada uma área de roteamento RA_i e, para todos os veículos n_j dentro de uma RA_i onde pelo menos uma via e_i de sua rota r_j esteja contida no conjunto C , é calculada uma nova rota para que o veículo evite o congestionamento.

O peso w para cada via e é calculado utilizando a Equação 5.1. A equação foi desenvolvida para o peso ser inversamente proporcional a condição de tráfego de veículos. Dessa forma, quanto melhor a condição do tráfego, menor é seu peso. Portanto, vias congestionadas possuem um peso maior do que rua com fluxo livre de veículos. A Equação 5.1 é definida da seguinte forma:

$$w = 1 - \frac{s_i}{d_i \cdot ms_i} \mid d_i > 0 \quad (5.1)$$

onde s_i , d_i e ms_i são definidas pela via $e \in E^2$ e, representam respectivamente a velocidade média, densidade e velocidade máxima da via e .

5.3 Detecção e Classificação de Congestionamentos

Devido à complexidade de lidar com uma enorme quantidade de informações, tais como, acidentes, mau tempo, problemas de sinalização, pontos de gargalos em horário de pico e grandes eventos, há várias abordagens para detectar congestionamentos [3, 11, 40, 29, 7]. Muitos estudos encontrados na literatura focam na identificação de status de cada via e

utilizam dados tais como a velocidade média dos veículos, para detectar os congestionamentos. Nestes casos, a técnica mais aplicada para classificar o nível de congestionamento é a lógica fuzzy [3, 1, 11]. O COLOSSUS difere destes estudos porque emprega um classificador que utiliza o algoritmo de K -Nearest Neighbor (KNN) [17] para a classificação de níveis de congestionamento.

Em geral, o k -NN precisa de uma base de dados de amostra para ser treinado. Para o efeito, foram empregados um conjunto de dados sintético construído de acordo com o *Highway Capacity Manual* (HCM) [6]. O HCM contém conceitos, diretrizes e procedimentos para medir a capacidade, o desempenho e a qualidade do tráfego, com base na velocidade e densidade de veículos nas vias. O conjunto de dados foi construído com base nos níveis de serviços *Level-of-Service* (LOS) presentes no HCM. Os *LOS* representam uma classificação para descrever a qualidade do tráfego de veículos. Além disso, o HCM define seis diferentes níveis de serviço, *LOS A* representa condições de fluxo livre e *LOS F* representa congestionamento intenso. Cada nível do *LOS*, define as velocidades mínima e máxima com base na velocidade máxima permitida na via. Com o conjunto de dados de treinamento empregado, é possível identificar quatro condições de tráfego, onde cada uma delas é baseado em limites de velocidade apresentados nos LOS do HCM. Além disso, para eliminar falsos positivos, utilizou-se a densidade de veículos nas vias em combinação com a velocidade média para definir uma condição de tráfego. A densidade é baseada na porcentagem de veículos na estrada de acordo com a sua capacidade máxima. Assim, a classificação de congestionamento é definida como *fluxo livre* = $1/4$, *congestionamento leve* = $1/2$, *congestionamento moderado* = $3/4$ e *congestionamento intenso* = 1. A Tabela 5.1 mostra a classificação da condição de tráfego.

Tabela 5.1: Classificação da condição de tráfego

Velocidade	Densidade			
	Baixa	Média	Alta	Muito Alta
Rápida	Fluxo livre	Fluxo livre	Fluxo livre	Leve
Média	Fluxo livre	Fluxo livre	Leve	Moderado
Lenta	Fluxo livre	Fluxo livre	Moderado	Moderado
Muito lenta	Leve	Moderado	Moderado	Intenso

A Figura 5.2 mostra como é o processo de classificação, onde a cada *intervalo de classificação* o COLOSSUS classifica todas as vias. Para isso, é utilizada a velocidade média s_i e densidade dos veículos d_i de cada via $e_i \in E$. Então, a via e_i é classificada utilizando o classificador KNN, dessa forma, depois de treinado, faz a classificação com base nos seus k -vizinhos mais próximos, utilizando a métrica de distância euclidiana. Por fim, após verificar os k -vizinhos mais próximos, o KNN pode classificar a via e_i , a qual pode ser classificada em: (i) Fluxo livre; (ii) Congestionamento leve; (iii) Congestionamento moderado e; (iv) Congestionamento intenso. Sendo assim, a saída do classificador é o nível de congestionamento da via.

Para realizar a detecção de congestionamento, é verificada a classificação de todas as vias, onde todas as vias com classificação *Congestionamento moderado* e *Congestionamento intenso* são adicionadas a um conjunto de vias congestionadas C a qual é um subconjunto de E . Por fim, para cada via $e_i \in C$ é definida uma área de roteamento para

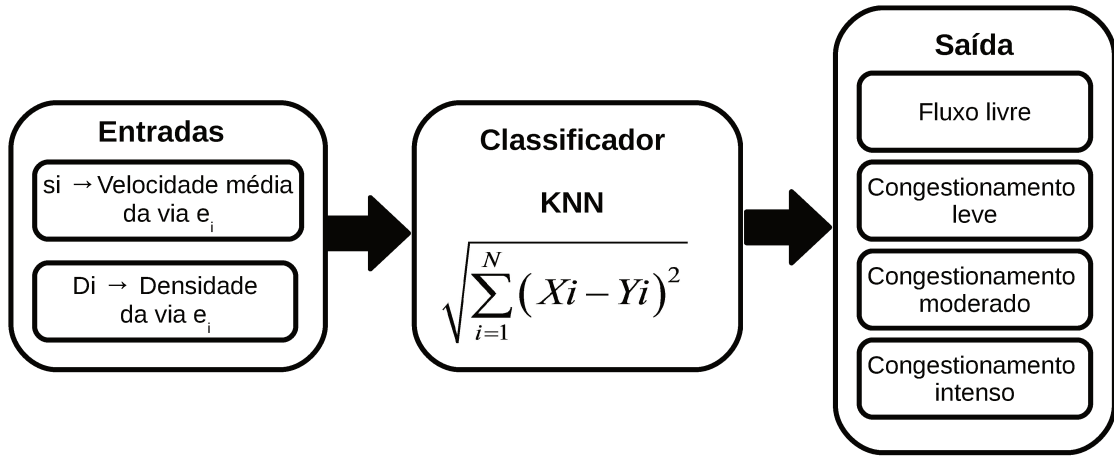


Figura 5.2: Processo de classificação implementado pelo COLOSSUS, mostrando as entradas utilizadas, o método implementado pelo classificador, e a saída da classificação.

rotear os veículos dentro dessa área que possuem ao menos uma via de sua rota contida em C .

A Figura 5.3 mostra um exemplo, onde o COLOSSUS já coletou as informações dos veículos e realizou a classificação de todas as vias do cenário. Após detectar uma via congestionada, é definida uma área de roteamento (RA) para a mesma (representada pelo círculo laranja), além disso, usando o módulo de *análise de rota*, é detectado que há um veículo dentro da RA que vai passar pela via congestionada (representada pela linha vermelha), dessa forma, o módulo de sugestão de rota é utilizado para calcular uma nova rota para o veículo, de modo que o mesmo evite de passar pela via congestionada (representada pela linha verde).

Os passos para realizar a detecção de congestionamento e classificação são mostrados no Algoritmo 3.

Algorithm 3: .

```

Input  :  $N$  // Conjunto de veículos na rede
        1  $G$  // Grafo criado por cada RSU

2 for  $e_i \in E$  do
3    $s_i \leftarrow \text{getVelocidadeMedia}(e_i)$ 
4    $d_i \leftarrow \text{getDensidade}(e_i)$ 
5    $\text{atualizaPesos}(s_i, d_i, e_i)$ 
6 end
7 if  $\text{intervaloDeClassificacao}()$  then
8    $C \leftarrow \text{classifyRoads}()$ 
9   for  $n_j \in N$  do
10     $r_j \leftarrow n_j.\text{getRoute}()$ 
11    for  $e_i \in r_j$  do
12      if  $n_j$  esta dentro da RA da via  $e_i \in C$  then
13        if  $e_i \in C$  then
14           $\text{posAtual} \leftarrow n_j.\text{getPosAtual}()$ 
15           $\text{destino} \leftarrow \text{getDestino}(r_j)$ 
16           $\text{alteraRota}(n_j, \text{posAtual}, \text{destino})$ 
17        end
18      end
19    end
20  end
21 end

```

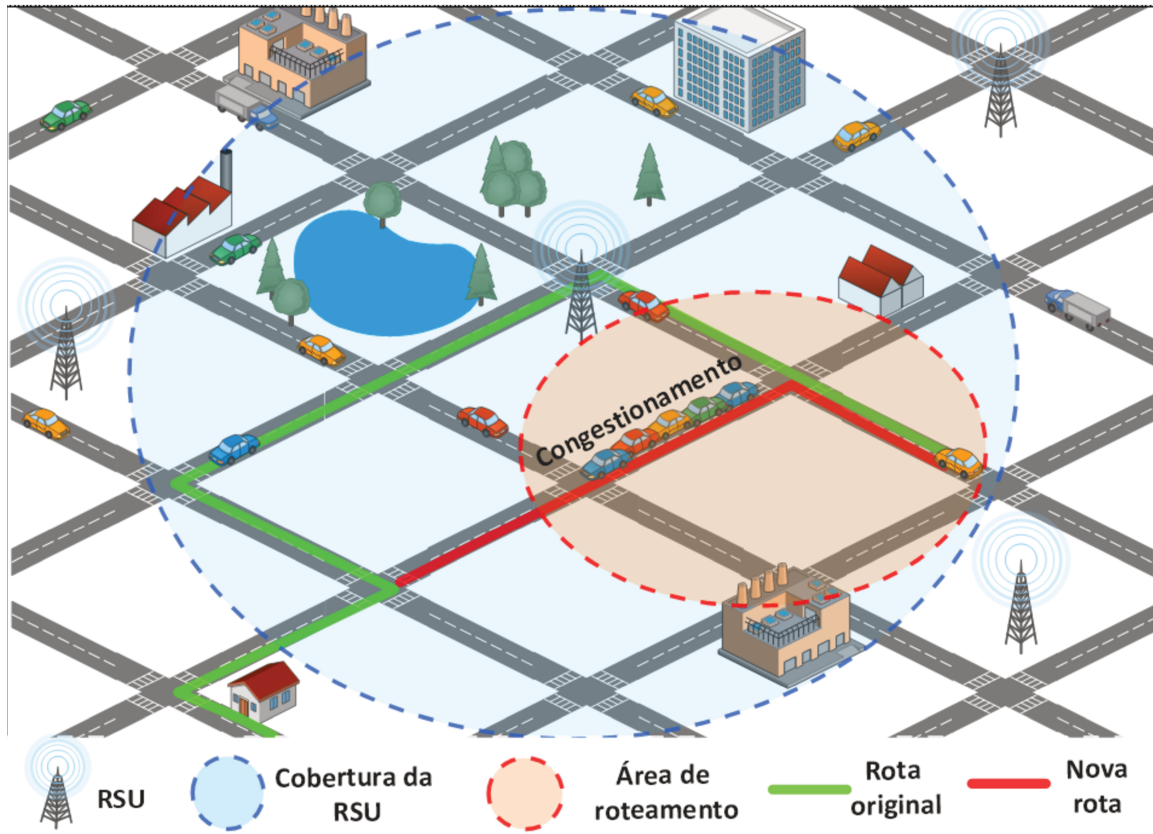


Figura 5.3: As principais entidades implementadas pelo COLOSSUS. Um veículo no interior da RA vai passar pelo congestionamento e é roteado.

O COLOSSUS faz a classificação das vias que estão sob a cobertura de cada RSU (veja Algoritmo 3 - Linhas 7 e 8). Ademais, cada RSU sabe a rota de cada veículo em sua própria cobertura, de modo que cada RSU pode determinar qual veículo vai passar por uma via congestionada (Algoritmo 3 - Linha 10 e 13). Por outro lado, para cada via congestionada, uma área de roteamento (RA) é definida para priorizar veículos que têm uma maior probabilidade de passar por esta via congestionada em um futuro próximo. Posteriormente, quando a RSU detecta que um veículo está dentro de um RA de qualquer via congestionada e este veículo vai passar por essa via congestionada (Algoritmo 3 - Linha 13), uma rota alternativa precisa ser calculada para evitar um congestionamento. Na Figura 5.3, é mostrado o momento em que a RSU detecta um veículo dentro de uma RA de uma via congestionada (veja o retângulo amarelo que representa a RA) e vai passar pela via congestionada (veja o seta vermelha que representa o percurso original do veículo). Assim, uma rota alternativa é calculada e enviada para o veículo, para evitar a via congestionada (ver a seta verde que representa o novo percurso do veículo).

5.4 Controle de Congestionamentos

Esta seção mostra o mecanismo de sugestão de rota utilizado para controlar os congestionamentos detectados. Dessa forma, primeiramente o COLOSSUS analisa a rota de todos os veículos dentro da área de roteamento, e verifica quais veículos passarão pelo

congestionamento, onde posteriormente uma nova rota é calculada para que os veículos evitem a área congestionada. Para calcular uma nova rota, o COLOSSUS utiliza duas abordagens, sendo elas: (i) Roteamento de melhor caminho (COLOSSUS-RMC), descrito na Subseção 5.4.1 e; (ii) Roteamento cooperativo (COLOSSUS-RC), apresentado na Subseção 5.4.2.

5.4.1 Roteamento de Melhor Caminho – COLOSSUS-RMC

Uma das principais preocupações do roteamento de veículos é como diminuir o congestionamento em uma área específica e evitar o surgimento de um novo congestionamento em um futuro próximo em outras áreas como uma resposta às mudanças no padrão de tráfego. Isto é, como evitar o problema de mover o congestionamento de uma via para outra. Para isso, o COLOSSUS-RMC utiliza um mecanismo para distribuir melhor o tráfego de veículos.

Para calcular uma nova rota utilizando o COLOSSUS-RMC, primeiro é verificado todos os veículos que estão dentro da área de roteamento e que vão passar por alguma via congestionada e, então é calculado os k menores caminhos para cada veículo utilizando o algoritmo de k shortest paths (onde k é definido pelo algoritmo), os k menores caminhos são calculados com base na condição do tráfego definido pela Equação. 5.1, onde depois de obter um conjunto R_j com os k menores caminhos, uma dessas rotas é atribuída para o veículo utilizando o algoritmo probabilístico [16].

Para escolher a rota entre os k menores caminhos, a fim de equilibrar o tráfego, o COLOSSUS-RMC calcula a soma dos pesos de todos os k menores caminhos. Os caminhos com os menores pesos totais são os mais adequados. Além disso, o conjunto R_j é usado para diminuir a probabilidade de sempre ser escolhida a mesma rota, equilibrando assim o tráfego entre as vias. As regras de decisão para escolher a nova rota são as seguintes: onde N é o conjunto de veículos no mapa; R_j é o conjunto de rotas do veículo n_j ($n_j \in N$); r_j^i é a rota i do veículo n_j ($n_j \in N$) e ($r_j^i \in R_j$); W_j^i é o peso da rota r_j^i .

$Norm(w_j^i)$ é o valor normalizado de w_j^i ($w_j^i \in [0, 1]$) definida pela Equação. 5.2.

$$Norm(w_j^i) = \frac{W(r_j^i)}{\max\{W(r_j^i) \mid \forall r_j^i \in R_j\}}. \quad (5.2)$$

$K_T^{n_j}$ é a constante Boltzmann do veículo n_j para temperatura T , de acordo com a Equação. 5.3:

$$K_T^{n_j} = \sum_{i \in R_j} e^{-(Norm(W_j^i)/T)}. \quad (5.3)$$

$P_T^{n_j}(r_j^i)$ é a probabilidade de selecionar a rota i para o veículo n_j com parâmetro de temperatura T , de acordo com a Equação. 5.4:

$$P_T^{n_j}(r_j^i) = \frac{1}{K_T^{n_j}} e^{-(N(w_j^i)/T)}. \quad (5.4)$$

Quando $T \rightarrow \infty$, todas as rotas candidatas têm a mesma probabilidade de serem

escolhidas, ou seja, o processo de escolha é uma distribuição aleatória uniforme. Quando $T \rightarrow 0$, a rota com menor peso tem maior probabilidade de ser escolhida.

$E(R_j)$ é a rota escolhida ($E(R_j \in R_j)$), a escolha é feita de acordo com a Equação. 5.5:

$$E(R_j) = \max\{X \times P_T^{n_j}(r_j^i) | \forall r_j^i \in R_j, X \sim \cup([0, 1])\}. \quad (5.5)$$

5.4.2 Roteamento Cooperativo – COLOSSUS-RC

No COLOSSUS-RC, o cálculo das novas rotas é feito em uma perspectiva global (visando o melhor fluxo da rede) para melhorar o fluxo no cenário todo. Diferente do COLOSSUS-RCM, onde a melhor rota é calculada para cada veículo individualmente, o COLOSSUS-RC visa evitar os congestionamentos e equilibrar o tráfego em vez de descobrir melhores rotas para cada veículo. Assim alguns veículos podem ser penalizados, mas minimiza a probabilidade de ocasionar novos congestionamentos o que é comum nas abordagens de melhor caminho.

A Figura 5.4 mostra todo o procedimento realizado pelo COLOSSUS-RC. Primeiramente, os veículos enviam suas informações pela rede e assim que essas informações são recebidas, o peso das arestas do grafo $G = (V, E)$ são atualizados. Além disso, é verificado se deve realizar a classificação (Figura 5.4 *label A*). Caso não for necessário, o COLOSSUS-RC continua a receber as informações dos veículos, caso contrário o mesmo faz a classificação de todas as vias e em seguida define as respectivas áreas de roteamento.

Logo após, é verificado quais veículos estão dentro de alguma área de roteamento RA_i (Figura 5.4 *label B*). Para os veículos que não estão dentro de uma área de roteamento, a rota original é mantida. Entretanto, para os veículos que estão dentro de uma área de roteamento $n_j \in RA_i$, é verificado se os mesmos irão passar por alguma via congestionada $e \in r_j \mid e \in C$ (Figura 5.4 *label C*). Se a condição for satisfeita, uma nova rota é calculada para cada veículo.

Durante o cálculo de uma nova rota, o sistema verifica a rota, a posição atual e o destino de todos os veículos. Além disso, para todos os veículos $n_j \in N$ que irão passar por uma via congestionada $r_j \subset C$ e estão dentro de uma área de roteamento RA_i , todos os possíveis caminhos a partir da posição atual do veículo até seu destino são calculados. O roteamento cooperativo implementa um algoritmo de busca gulosa. A partir daí, para construir uma nova rota, o COLOSSUS-RC não só verifica todos os caminhos possíveis que o veículo pode percorrer a partir da sua posição atual, com exceção dos caminhos que impedem o veículo de chegar ao seu destino, mas também seleciona a via com menor peso como próxima via da nova rota. Além disso, depois de selecionar a próxima via da nova rota, o peso da aresta selecionada é atualizado e o procedimento seleção continua até que construir toda a rota (Figura 5.4 *label D e label E*).

A atualização do peso da via selecionada é feito de acordo com o impacto que aquele veículo terá na mesma. Dessa forma, a atualização do peso é feita, utilizando a Equação 5.6

$$w_i = w_i + (w_i \cdot \frac{l_{n_j}}{l_{e_i}}) \quad (5.6)$$

onde, w_i é o peso atual da via e_i , l_{n_j} é o comprimento do veículo n_j e l_{e_i} é o comprimento

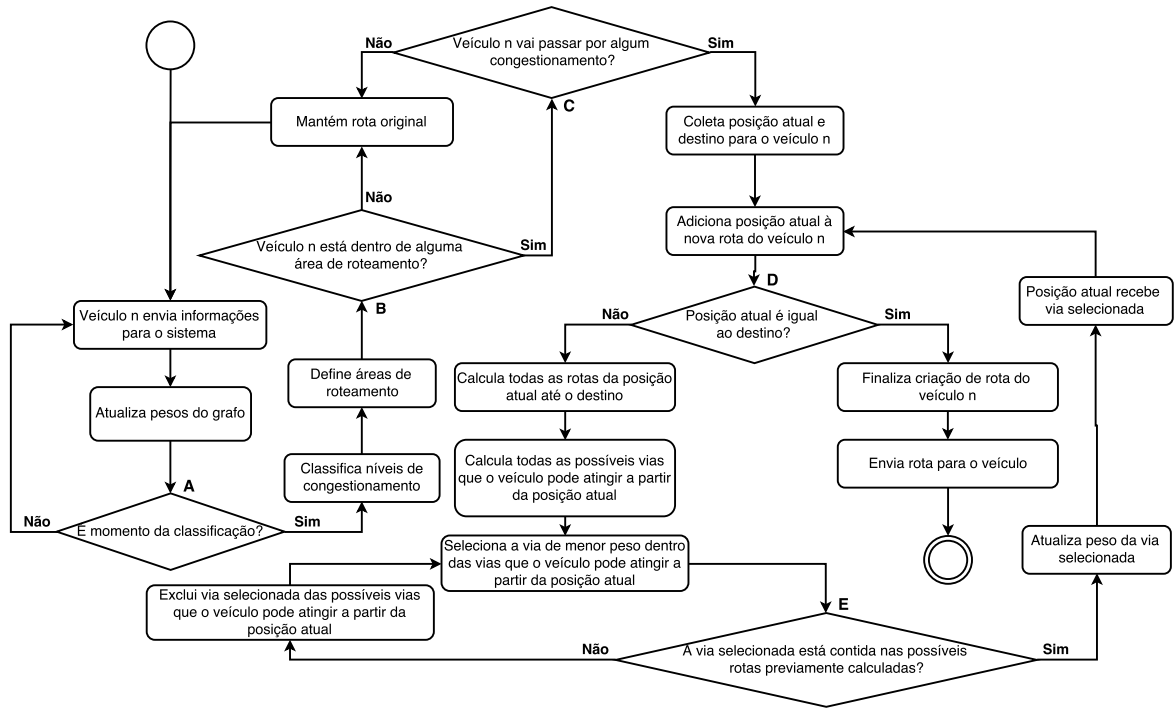


Figura 5.4: Fluxo de operações do COLOSSUS-RC para classificação do tráfego, detecção de congestionamentos e roteamento dos veículos

da via e_i . Selecionar vias com menor peso melhora a eficiência do tráfego evitando vias congestionadas e o processo de atualizar o peso da via selecionada é importante para balancear o tráfego e contribuir para um fluxo de tráfego contínuo.

A Figura 5.5 descreve como o cálculo de rota é feita pelo o COLOSSUS-RC. A figura mostra o planejamento de rota para um veículo que vai passar por uma via congestionada com a posição atual no vértice A e destino no vértice H . Na Figura 5.5(a) depois de calcular todas as possíveis rotas de A até H , o COLOSSUS-RC inicia o planejamento da rota. A partir do vértice A o veículo pode alcançar B e E (ambos estão dentro de todos os caminhos possíveis calculados anteriormente). No entanto, a aresta que conecta vértice A e E tem o menor peso, portanto, esta via é selecionada como a próxima aresta da nova rota do veículo e seu peso é atualizado. Na Figura 5.5(b) a partir do vértice E o veículo pode atingir os vértices C , D , F e G , mas os vértices D e F são excluídos dos possíveis próximos vértices porque eles não fazem parte das possíveis rotas calculadas anteriormente. Além disso, entre os vértices C , G e H a aresta que conecta vértice E e o vértice C tem o menor peso, por isso é selecionada como a próxima aresta e seu peso é atualizado. Por fim, na Figura 5.5(c), a partir do vértice C o veículo pode chegar aos vértices G e H , ambos estão em vias possíveis, no entanto, a aresta que conecta vértice C e H tem o menor peso, portanto, é selecionado como próxima aresta que conecta vértice C e H e o seu peso é atualizado e, a nova rota calculada é $r_j \leftarrow \{A, E, C, H\}$. Além disso, como os pesos das estradas que formam a nova rota foram atualizados, o novo percurso para outro veículo com a mesma posição atual e destino pode ser $r_j \leftarrow \{A, B, E, C, G, H\}$. No entanto, usando outro algoritmo, sem perspectiva global para construir a nova rota, como *Shortest Path*, a rota para dois ou mais veículos com origem no vértice A e destino

no vértice H seriam $r_j \leftarrow \{A, E, H\}$.

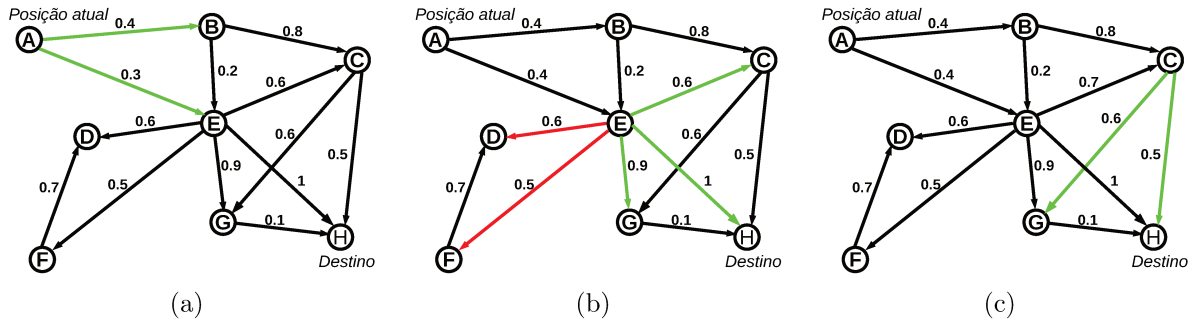


Figura 5.5: Cálculo de uma rota para um veículo com posição atual no vértice A e com destino no vértice H

5.5 Avaliação de Desempenho

Esta seção descreve a metodologia aplicada, as ferramentas utilizadas, o cenário de simulação e os resultados das simulações comparando o COLOSSUS (utilizando as duas abordagens de roteamento) com as soluções apresentadas da literatura DPS, RkSP e With routing. Por fim, uma análise exploratória é feita sobre o COLOSSUS-RC.

Para nossa avaliação, foi utilizado o simulador de mobilidade SUMO (*Simulator for Urban MObility*) [5], versão 0.17.0. A implementação do gerenciamento do tráfego no SUMO é feita pelo TraCI (*Traffic Control Interface*) [37], o qual permite o gerenciamento em tempo real durante a simulação. Além disso, ele provê acesso e controle a objetos de simulação, alterando seu comportamento em tempo de execução. Para calcular as emissões de CO e o consumo de combustível, foi utilizado o modelo EMIT implementado pelo SUMO. EMIT é um modelo estatístico baseado na fórmula HBEFA², a qual calcula as emissões de CO e o consumo de combustível baseado na aceleração e velocidade dos veículos.

É importante salientar que foi utilizado um cenário real para as simulações, retirado do OpenStreetMap³. Foi selecionado uma parte do bairro de Manhattan, Nova York, Estados Unidos, com um tamanho de 4 km², como mostra a Figura 4.5. A mobilidade dos veículos foi gerada de forma aleatória com as seguintes densidades: 1000 e 1500 veículos/km².

A Tabela 5.2 apresenta os parâmetros de simulação utilizados. Por uma questão de simplicidade, foi utilizado o algoritmo de KNN como classificador com $k=5$, valor no qual obteve-se melhores resultados em nossas avaliações preliminares. Por fim, para cada análise, os resultados representam a média de 33 replicações com intervalo de confiança de 95%.

Além da comparação com as soluções encontradas na literatura DSP, RkSP [26] e With routing [7] o sistema também foi comparado com a abordagem tradicional OVMT (*Original Vehicular Mobility Trace*). Além disso, para analisar a performance das soluções, as seguintes métricas foram utilizadas:

²<http://www.hbefa.net>

³<http://www.openstreetmap.org/>

Tabela 5.2: Parâmetros de simulação para detecção e controle de congestionamentos

Parâmetros	Valores
Cenário	4 km ²
Intervalo de classificação	15, 30, 60, 120, 240 s
Área de roteamento (RA)	200, 400, 600, 1000 m ²
Densidade	1000, 1500 veículos/km ²
# vizinhos mais próximos	5
Intervalo de Confiança	95%

- **Eficiência:** relação da velocidade média e da densidade de todos os veículos durante cada segundo da simulação;
- **Velocidade média:** velocidade média dos veículos durante a simulação;
- **Tempo médio de viagem:** tempo médio em que os veículos levaram para atingir seu destino;
- **Tempo de congestionamento:** tempo médio em que os veículos ficaram presos em um congestionamento;
- **Distância percorrida:** distância média percorrida por todos os veículos;
- **Consumo de combustível:** consumo médio de combustível de todos os veículos;
- **Emissão de CO:** emissão média de CO emitida por todos os veículos na simulação.

5.6 Eficiência das soluções de controle de congestionamentos

Nesta seção é avaliada a eficiência de cada solução para controle de congestionamentos, a eficiência foi medida com a correlação entre a velocidade média de todos os veículos e a densidade de veículos a cada segundo de simulação. Dessa forma, o quão mais eficiente for o controle de congestionamento, a densidade de veículos diminui e a velocidade média do cenário aumenta mais rapidamente.

A Figura 5.6 mostra a correlação entre a velocidade média e densidade para cada solução durante todo o tempo de simulação com densidade de 1500 v/km². A Figura 5.6(a) representa o tráfego OVMT e mostra as características do congestionamento, como pode ser observado durante quase toda a simulação a velocidade média é inferior a 10 km/h. Além disso, a simulação dura aproximadamente 55 minutos e a velocidade média torna-se maior do que a densidade apenas em aproximadamente 35 minutos. Os resultados do DSP, na Figura 5.6(b), mostram o quão eficiente é o roteamento periódico dos veículos, onde pode ser observado não só o aumento da velocidade média quando comparado com o OVMT, mas também a redução do tempo de simulação para aproximadamente 45 minutos. No entanto, no DSP, quando os veículos são redirecionados, o melhor caminho é sempre selecionado para cada veículo. Portanto, com este mecanismo de roteamento

simples, outros congestionamentos podem surgir em outros locais na rede e o mesmo será detectado somente na próxima fase de roteamento de acordo com o intervalo de roteamento.

A solução RkSP minimiza o problema apresentado pelo DSP selecionando uma rota entre as k rotas possíveis para cada veículo, equilibrando assim o tráfego e diminuindo a probabilidade do surgimento de um novo congestionamento em outro local. Entretanto, rotas com maiores distâncias podem ser selecionadas, aumentando assim o tempo de viagem, como pode ser observado onde o tempo de simulação atinge 60 minutos. Diferente do RkSP, que seleciona uma rota de maneira aleatória, a solução *With routing*, Figura 5.6(d), seleciona uma nova rota, usando um método probabilístico. O impacto desta seleção pode ser visto na velocidade média, que se comporta de forma semelhante ao RkSP, e o tempo de simulação que diminui quando comparado com RkSP. A Figura 5.6(e) mostra o desempenho do COLOSSUS-RMC, onde pode ser visto a eficiência do mecanismo para detecção e controle de congestionamentos. O tempo de simulação foi reduzido em aproximadamente 50% quando comparado com OVMT e a velocidade média é maior do que todas as outras soluções, chegando a 25 km/h. Por fim, na Figura 5.6(f) é mostrado os resultados para o COLOSSUS-RC, onde o tempo de simulação foi reduzido em aproximadamente 13% quando comparado com o COLOSSUS-RMC, essa redução se deve ao fato dos veículos serem melhores distribuídos no cenário, reduzindo assim o surgimento de novos congestionamentos em um curto período de tempo.

5.7 COLOSSUS *vs* DSP, RkSP e With Routing

A Figura 5.7 mostra os resultados de todas as métricas avaliadas para todas as soluções. Como pode ser observado, o OVMT tem uma distância média percorrida de 3,52 km e um tempo médio de viagem 15,09 minutos, Figuras 5.7(c) e 5.7(a), respectivamente. Isso significa que os veículos passam cerca de 49% do tempo médio de viagem em congestionamentos (ver Figura 5.7(b)). Enquanto no DSP, o tempo médio de viagem reduz aproximadamente 5%, quando comparado com o OVMT. Esta redução deve-se ao roteamento periódico de todos os veículos, no entanto, esse roteamento aumenta a distância média percorrida em 22% (ver Figura 5.7(c)). Além disso, o DSP faz o roteamento de todos os veículos utilizando o menor caminho, resultando em muitos veículos roteados pela mesma via, criando assim congestionamentos em outras áreas. Portanto, os veículos passam 37% de todo o tempo médio de viagem em um congestionamento quando é usado o DSP (ver a Figura 5.7(b)).

As soluções RkSP e *With routing* distribuem melhor o tráfego de veículos, selecionando uma rota a partir de conjunto de k rotas possíveis, diminuindo assim a probabilidade do surgimento de novos congestionamentos em outras áreas. No entanto, a solução RkSP aumenta o tempo médio de viagem e a distância média percorrida em 6% e 36% respectivamente, quando comparado com o OVMT. Por razão da seleção aleatória de uma rota em um conjunto de k rotas possíveis, as rotas com maiores distâncias podem ser selecionadas, aumentando assim o tempo médio de viagem e a distância média percorrida. Porém, os veículos passam 37% do tempo médio de viagem presos em um congestionamento.

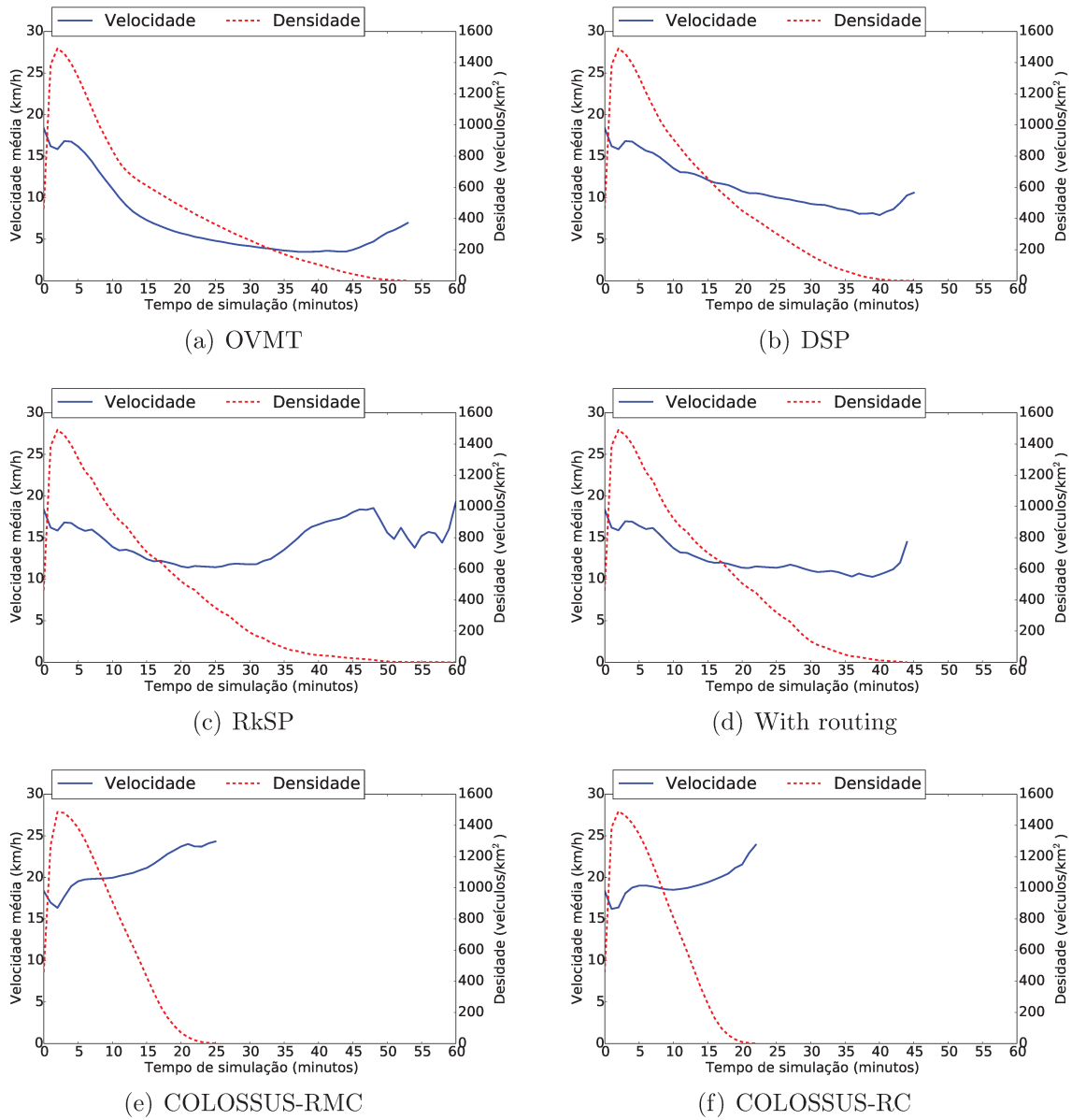


Figura 5.6: Eficiência do controle de congestionamento das soluções, correlação da velocidade média e densidade total do cenário a cada instante da simulação.

Para evitar os problemas associados com a seleção de rota aleatória, o *With routing* utiliza um método probabilístico para selecionar uma nova rota e equilibrar o tráfego. Essa solução aumenta o tempo médio de viagem em 2% e aumenta a distância média percorrida em 30% quando comparado com o OVMT. Igualmente ao RkSP, os veículos passam 36% do tempo médio de viagem em um congestionamento. Por outro lado, os resultados do COLOSSUS-RMC mostram a eficiência do sistema. Portanto, no COLOSSUS-RMC, logo que ocorre um congestionamento, o mesmo é detectado e já pode ser controlado, diferentemente das outras soluções que apenas detectam um congestionamento durante intervalos de roteamento. Os principais resultados são as reduções do tempo médio de viagem em 24% (ver Figura 5.7(a)) eo tempo médio de congestionamento em 67% (ver Figura 5.7(b)), quando comparado com OVMT. Por fim, são apresentados os resultados do COLOSSUS-RC, a eficiência do roteamento cooperativo é devido ao planejamento das

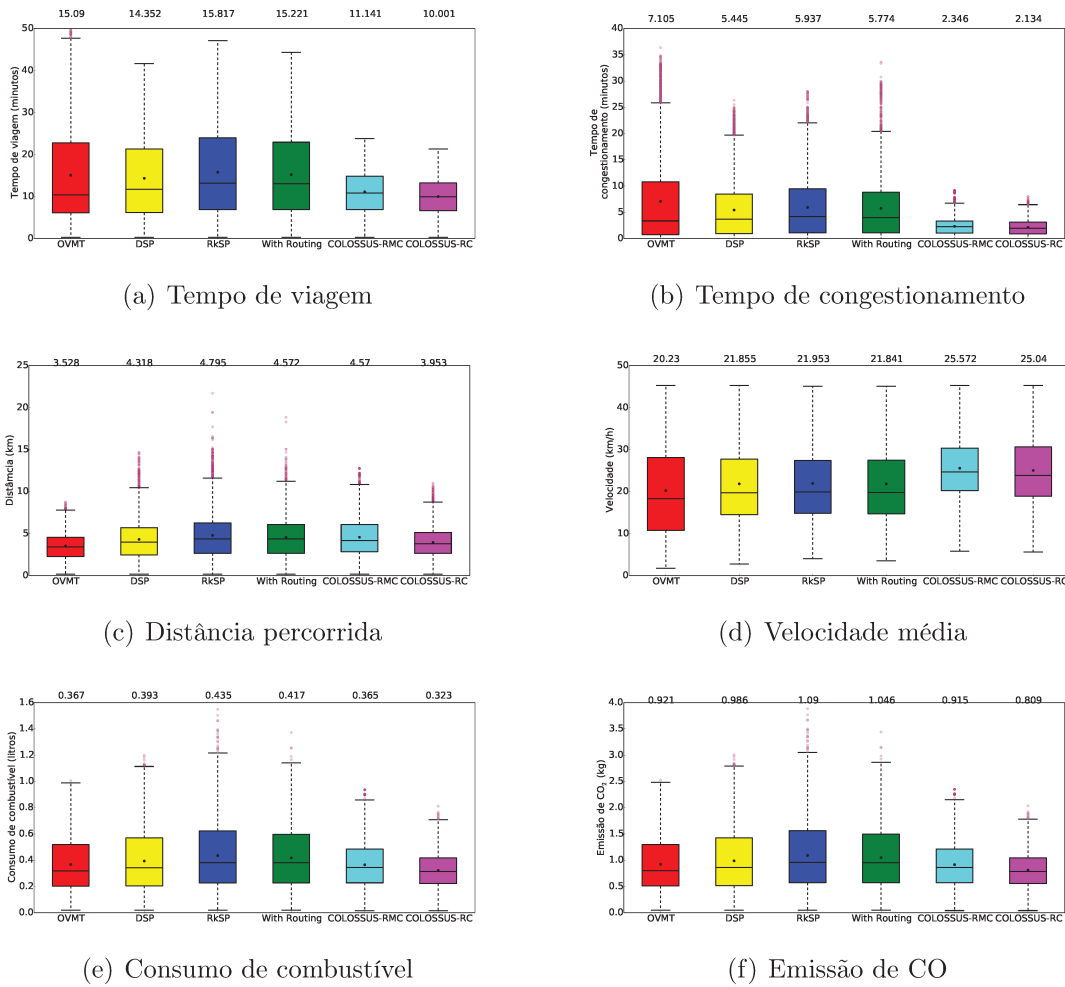


Figura 5.7: Avaliação das Soluções para Controle de Congestionamentos

rotas, visando o melhor fluxo da rede como um todo, portanto, menos congestionamentos se formam, assim liberando o tráfego nas vias mais rapidamente. Em linhas gerais, ao comparar o COLOSSUS-RC com a abordagem tradicional, o mesmo reduz o tempo de viagem em 33%, tempo de congestionamento em 70% e aumenta a distância em apenas 12%. Por outro lado, comparando com o COLOSSUS-RMC, o COLOSSUS-RC reduz o tempo de viagem em 10%, tempo de congestionamento em 9% e distância em 13%, isso mostra a eficiência do roteamento cooperativo.

Os resultados de consumo de combustível estão diretamente relacionados com o tempo de viagem, tempo de congestionamento e distância percorrida. A Figura 5.7(e) mostra os resultados para o consumo de combustível. Como pode ser observado, o OVMT tem um consumo médio de combustível de 0.36 litros. DSP, RkSP e *With routing* aumentam o consumo de combustível em 5%, 16% e 11%, respectivamente, quando comparado com OVMT. Este aumento é devido à maior distância percorrida destas soluções. No entanto, tanto para o COLOSSUS-RMC, quanto para o COLOSSUS-RC reduz-se o consumo de combustível, o consumo é reduzido em aproximadamente 1% para o COLOSSUS-RMC e em aproximadamente 12% para o COLOSSUS-RC devido ao melhor controle dos congestionamentos. Pois os veículos passam menos tempo presos em congestionamentos. Por fim,

a Figura 5.7(f) mostra os resultados para as emissões de CO. Como esperado, as emissões de CO estão relacionadas com o consumo de combustível. Portanto, o DSP, RkSP e *With routing* aumentam as emissões em 6%, 17% e 11%, respectivamente, quando comparados com OVMT e o COLOSSUS-RCM diminui as emissões em aproximadamente 1% e o COLOSSUS-RC reduz em 12%.

5.8 Análise Exploratória do COLOSSUS

Está seção apresenta uma análise exploratória sobre o COLOSSUS, onde primeiramente é avaliado o impacto do parâmetro de intervalo de classificação sobre os resultados do COLOSSUS e, por fim, é avaliado o impacto do tamanho da área de roteamento. Ademais, para avaliação dos parâmetros, foi utilizado apenas o COLOSSUS-RC, pois foi a solução que obteve melhores resultados. Por fim, a análise dos dois parâmetros foi feita sob o mesmo cenário da Seção 5.5 e, além disso, foi utilizado uma densidade de 1000 veículos/km².

5.8.1 Análise do Intervalo de Classificação

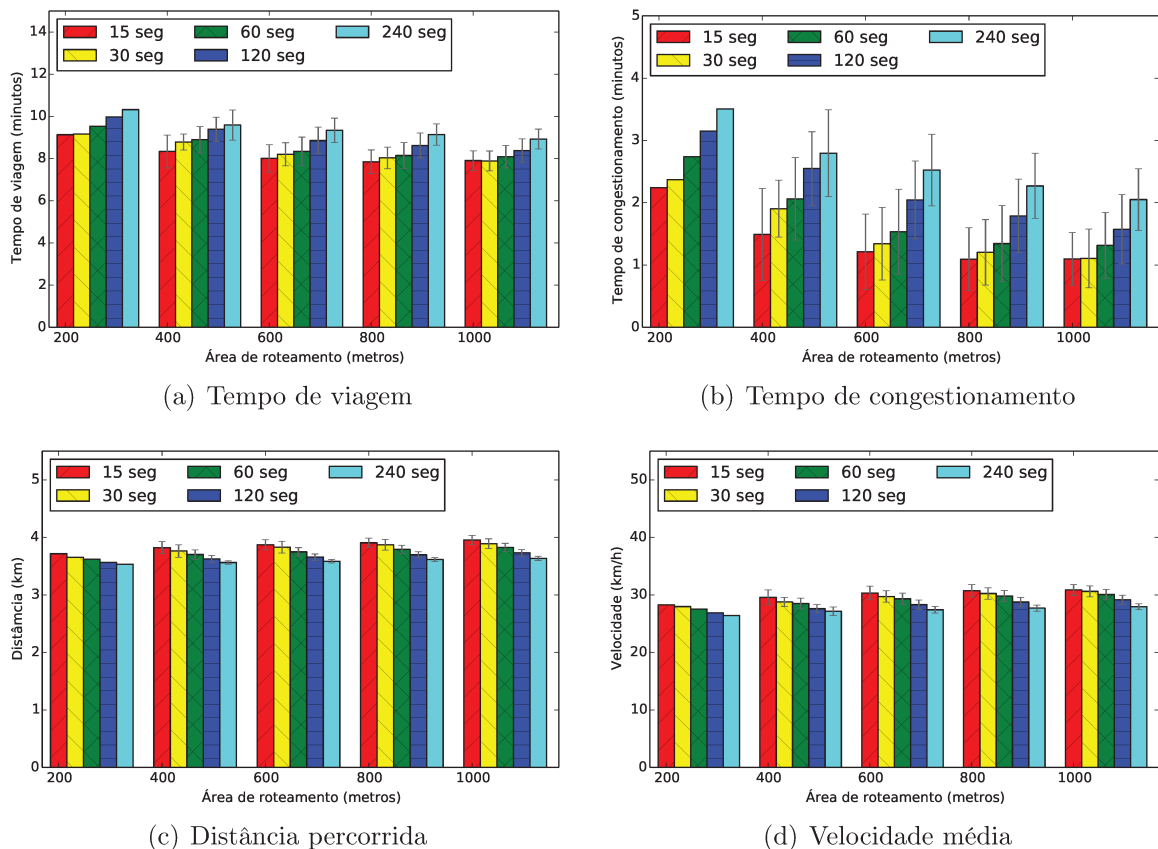


Figura 5.8: Análise do intervalo de classificação do COLOSSUS-RC em relação à área de roteamento.

Nesta subseção são apresentados os resultados sobre a análise do intervalo de classificação implementado pelo COLOSSUS-RC. A Figura 5.8, mostra os resultados dos intervalos

de classificação em relação ao tamanho da área de roteamento, onde os intervalos de classificação implementados no COLOSSUS-RC são: 15, 30, 60, 120, 240 segundos. Em particular, a Figura 5.8(a) apresenta os resultados para o tempo de viagem, onde pode ser visto que quanto maior o intervalo de classificação, maior é o tempo de viagem, isto se deve ao fato de quanto maior o intervalo de classificação, mais tarde serão detectados os congestionamentos, dessa forma, mais veículos entrarão nas áreas congestionadas, como pode ser visto na Figura 5.8(b), onde o tempo de congestionamento aumenta aproximadamente 40% quando comparado o intervalo de classificação de 15 e 240 segundos. Entretanto, quanto menor o intervalo de classificação, os veículos serão roteados com maior frequência para evitar os congestionamentos detectados, portanto, como consequência a distância percorrida aumenta cerca de 10% quando comparado o intervalo de classificação de 240 e 15 segundos. Por fim, A velocidade média para os intervalos de classificação são apresentados na Figura 5.8(d), onde, quanto menor o intervalo de classificação, maior é a velocidade média, porque com intervalos de classificação menores, os congestionamentos serão detectados primeiro, dessa forma o COLOSSUS-RC pode controlar os congestionamentos detectados antes do que quando usando grandes intervalos de classificação, como pode ser visto na Figura 5.8(b).

5.8.2 Análise da Área de Roteamento

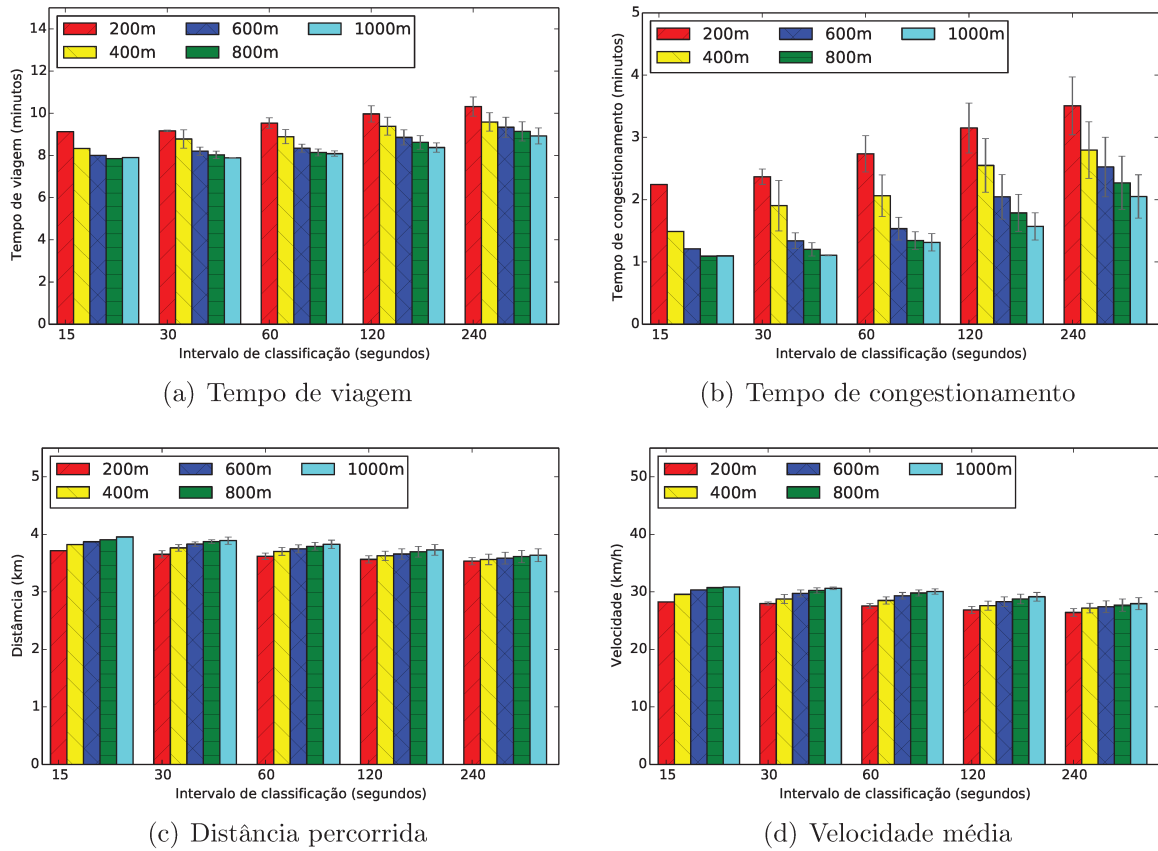


Figura 5.9: Análise da área de roteamento do COLOSSUS-RC em relação ao intervalo de classificação.

Esta subseção apresenta os resultados para a análise do tamanho da área de roteamento implementado no COLOSSUS-RC. A Figura 5.9, apresenta os resultados do tamanho da área de roteamento em relação ao intervalo de classificação, onde foram analisados os seguintes tamanhos para a área de roteamento: 200, 400, 600 e 100 m². Em particular, a Figura 5.9(a), apresenta os resultados para o tempo de viagem, onde pode ser observado que, quanto maior o tamanho da área de roteamento, menor é o tempo de viagem, isto é devido ao fato da utilização do roteamento cooperativo, onde com uma maior área de roteamento, mais veículos serão roteados, melhorando assim a condição do tráfego, ou seja, distribuindo melhor o tráfego dos veículos e reduzindo os congestionamentos. Os resultados para a redução dos congestionamentos pode ser visto na Figura 5.9(b), onde a área de roteamento de 1000 metros, reduz o tempo de congestionamento em até aproximadamente 60% quando comparado com a área de roteamento de 200 metros. Por outro lado, como consequência, quanto maior o tamanho da área de roteamento, mais veículos serão roteados para evitar áreas congestionadas, portanto a distância média percorrida tende a aumentar (ver Figura 5.9(c)). Por fim, a Figura 5.9(d), apresenta os resultados para a velocidade média, onde, quanto maior a área de roteamento, maior é a velocidade média, pois com uma área de roteamento maior, melhor será a distribuição do tráfego pelo roteamento cooperativo, reduzindo assim os congestionamentos 5.9(b).

5.9 Resumo Conclusivo

Neste capítulo foi apresentado COLOSSUS, um sistema para detecção, classificação e controle de congestionamentos, onde o objetivo principal do COLOSSUS é coletar os dados de todos os veículos e periodicamente fazer uma classificação das condições de tráfego em todas as vias e detectar áreas congestionadas, onde em seguida, o COLOSSUS analisa as rotas dos veículos dentro de uma área de roteamento e faz a alteração das rotas dos veículos que iriam passar por áreas congestionadas, a fim de distribuir melhor o tráfego de veículos no e reduzir os congestionamentos.

Para classificar a condição do tráfego em todas as vias e detectar congestionamentos, foi utilizado o classificador KNN, além disso, foi construída uma base de dados sintética para o classificador com base no HCM. Ademais, quando um congestionamento é detectado, o COLOSSUS define uma área de roteamento para priorizar o roteamento dos veículos mais próximos do congestionamento. Portanto o COLOSSUS analisa as rotas dos veículos que estão dentro dessa área e para os veículos que irão passar por alguma área congestionada, uma nova rota é calculada.

Para o cálculo da nova rota, o COLOSSUS implementa duas abordagens, sendo elas: (i) roteamento de melhor caminho (COLOSSUS-RMC), a qual visa distribuir o tráfego no cenário utilizando uma das k menores rotas para o veículo e; (ii) roteamento cooperativo (COLOSSUS-RC), a qual visa balancear melhor o tráfego de veículos como um todo, dessa forma, não são calculadas as melhores rotas para cada veículo, mas sim as melhores rotas que contribuirão para um melhor tráfego no cenário todo.

Os resultados mostraram a eficiência do COLOSSUS, onde tanto para o roteamento de melhor caminho, quanto para o roteamento cooperativo, foram apresentados melhores

resultados do que as soluções da literatura. Entretanto, é importante salientar que o roteamento cooperativo, obteve melhores resultados em todas as métricas avaliadas. Por fim, uma análise exploratória foi realizada para avaliar o impacto do intervalo de classificação e o tamanho da área de roteamento para o COLOSSUS.

Capítulo 6

Considerações Finais

Este capítulo está dividido em duas partes. A Seção 6.1 apresenta as conclusões e a Seção 6.2 apresenta os trabalhos futuros.

6.1 Conclusões

Com os avanços das tecnologias sem fio especialmente com o desenvolvimento das VANETs tornou-se possível o desenvolvimento de sistemas de transporte inteligentes, os quais permitem o gerenciamento do trânsito em tempo real. Nessa dissertação de mestrado foram propostos dois sistemas de transporte inteligentes, nomeados como SPARTAN e COLOSSUS, para minimizar os congestionamentos em centros urbanos.

O SPARTAN faz uso do sistema OBD 2 (*On Board Diagnostic 2*) para detectar a ocorrência de um acidente, para disseminar a mensagem de alerta dentro da AoI foi proposto uma versão para ambientes urbanos do protocolo de disseminação de dados ADD (*A Data Dissemination Solution for Highly Dynamic Environments*) [30]. Por fim, o SPARTAN faz uso dos algoritmos *Dijkstra*, A^* e *Probabilistic k-Shortest Paths* para computar novas rotas para os veículos evitarem a área congestionada ou que possivelmente ficará congestionada em um futuro próximo. Os resultados mostram que o SPARTAN é uma boa opção para controle de congestionamentos causados por acidentes, onde o mesmo reduz o tempo de viagem, tempo de congestionamento e aumenta a velocidade média, além de reduzir o consumo de combustível e emissões de CO. Além disso, os resultados mostram que a versão do ADD para ambientes urbanos é uma boa opção para disseminação de alertas em redes veiculares, uma vez que o mesmo maximiza a cobertura, com um menor número de mensagens enviadas, baixo atraso e com um número inferior de colisões, quando comparado com os protocolos da literatura.

Além do SPARTAN, foi apresentado o COLOSSUS, um sistema para detecção, classificação e controle de congestionamentos. Para isso, o COLOSSUS faz a coleta das informações dos veículos que estão sob a cobertura das RSUs, em seguida faz uma classificação periodicamente da condição de tráfego em todas as vias para detectar as vias congestionadas. Após detectar as vias congestionadas, o COLOSSUS define uma área de roteamento para as mesmas, em seguida, novas rotas são computadas para os veículos dentro da área de congestionamento e que passarão pelas vias congestionadas. Para computar uma nova

rota, foram apresentadas duas abordagens, uma de melhor caminho e uma abordagem de roteamento cooperativo. Os resultados mostram que, apesar de alguns veículos serem penalizados no roteamento cooperativo, o mesmo contribui para uma melhor distribuição do tráfego qual maximiza o fluxo de veículos. Os resultados mostram que o COLOSSUS com roteamento cooperativo reduz o tempo de viagem, tempo de congestionamento, consumo de combustível e emissões de CO, além de aumentar a velocidade média dos veículos e otimizar o fluxo de veículos.

6.2 Trabalhos Futuros

Os trabalhos futuros são:

- Realizar a integração dos sistemas SPARTAN e COLOSSUS;
- Avaliar as questões relacionadas à rede LTE/4G no COLOSSUS;
- Avaliar a viabilidade da utilização de uma rede LTE/4G, ou uma rede heterogênea (LTE/4G + 802.11p) em relação a densidade da rede.
- Projetar uma solução distribuída para detecção e controle de congestionamentos para operar sem a necessidade de infraestruturas.

Referências Bibliográficas

- [1] G.B. Araujo, M.M. Queiroz, F.L.P. Duarte-Figueiredo, A.I.J. Tostes, and A.A.F. Loureiro. Cartim: A proposal toward identification and minimization of vehicular traffic congestion for vanet. In *Computers and Communication (ISCC), 2014 IEEE Symposium on*, pages 1–6, June 2014.
- [2] M. Bakhouya, J. Gaber, and P. Lorenz. An adaptive approach for information dissemination in vehicular ad hoc networks. *Journal of Network and Computer Applications*, In Press, Uncorrected Proof, 2011.
- [3] R. Bauza and J. Gozalvez. Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications. *Journal of Network and Computer Applications*, 36(5):1295 – 1307, 2013.
- [4] R. Bauza, J. Gozalvez, and J. Sanchez-Soriano. Road traffic congestion detection through cooperative vehicle-to-vehicle communications. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 606–612, Oct 2010.
- [5] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO - Simulation of Urban MObility: An Overview. In *International Conference on Advances in System Simulation (SIMUL '11)*, pages 63–68, 2011.
- [6] Transportation Research Board. Hcm 2010 - highway capacity manual, 2000.
- [7] Celso Brennand, Allan Souza, Guilherme Maia, Azzedine Boukerche, Antonio Loureiro, and Leandro Villas. An intelligent transportation system for detection and control of congested roads in urban centers. In *Computers and Communications (ISCC), 2015 IEEE Symposium on*, July 2015.
- [8] Marcos Cintra. A crise do trânsito em são paulo e seus custos. *GVExecutivo*, 12(2):58–61, Jul/Dez 2013.
- [9] Felipe Domingos Da Cunha, Azzedine Boukerche, Leandro Villas, Aline Carneiro Viana, and Antonio A. F. Loureiro. Data communication in vanets: A survey, challenges and applications. Research Report RR-8498, INRIA Saclay ; INRIA, March 2014.
- [10] R. Doolan and G.-M. Muntean. Vanet-enabled eco-friendly road characteristics-aware routing for vehicular traffic. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–5, June 2013.

- [11] S. Dornbush and A. Joshi. Streetsmart traffic: Discovering and disseminating automobile congestion using vanet's. In *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pages 11–15, April 2007.
- [12] Mohammad Ilyas. *The handbook of ad hoc wireless networks*. CRC press, 2014.
- [13] D. Jiang and L. Delgrossi. Ieee 802.11p: Towards an international standard for wireless access in vehicular environments. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2036–2040, May 2008.
- [14] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *Communications Surveys Tutorials, IEEE*, 13(4):584–616, Fourth 2011.
- [15] Tae-Hwan Kim, Won-Kee Hong, Hie-Cheol Kim, and Yong-Doo Lee. Information networking. towards ubiquitous networking and services. chapter An Effective Data Dissemination in Vehicular Ad-Hoc Network, pages 295–304. Springer-Verlag, Berlin, Heidelberg, 2008.
- [16] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.
- [17] Daniel T. Larose. *k-Nearest Neighbor Algorithm*, pages 90–106. John Wiley and Sons, Inc., 2005.
- [18] Uichin Lee and Mario Gerla. A survey of urban vehicular sensing platforms. *Comput. Netw.*, 54(4):527–544, March 2010.
- [19] Fan Li and Yu Wang. Routing in vehicular ad hoc networks: A survey. *Vehicular Technology Magazine, IEEE*, 2(2):12–22, June 2007.
- [20] Joseph Macker and Scott Corson. *Mobile ad hoc networks (MANET)*. 1997.
- [21] Guilherme Maia, Cristiano Rezende, Leandro A. Villas, Azzedine Boukerche, Aline C. Viana, Andre L. Aquino, and Antonio A. Loureiro. Traffic aware video dissemination over vehicular ad hoc networks. In *Proceedings of the 16th ACM International Conference on Modeling, Analysis, Simulation of Wireless and Mobile Systems, MSWiM '13*, pages 419–426, New York, NY, USA, 2013. ACM.
- [22] Valery Naumov, Rainer Baumann, and Thomas Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '06*, pages 108–119, New York, NY, USA, 2006. ACM.
- [23] Vi Tran Ngoc Nha, S. Djahel, and J. Murphy. A comparative study of vehicles' routing algorithms for route planning in smart cities. In *Vehicular Traffic Management for Smart Cities (VTM), 2012 First International Workshop on*, pages 1–6, Nov 2012.

- [24] Stephan Olariu and Michele C Weigle. *Vehicular networks: from theory to practice*. CRC press, 2019.
- [25] OMNET++". Omnet++ network simulation framework.
- [26] J. Pan, M.A. Khan, I. Sandu Popa, K. Zeitouni, and C. Borcea. Proactive vehicle re-routing strategies for congestion avoidance. In *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on*, pages 265–272, May 2012.
- [27] S. Panichpapiboon and W. Pattara-Atikom. A review of information dissemination protocols for vehicular ad hoc networks. *Communications Surveys Tutorials, IEEE*, 14(3):784–798, Third 2012.
- [28] C. Sommer, S. Joerer, and F. Dressler. On the applicability of two-ray path loss models for vehicular network simulation. In *Vehicular Networking Conference (VNC), 2012 IEEE*, pages 64–69, Nov 2012.
- [29] Allan Mariano de Souza, Azzedine Boukerche, Guilherme Maia, Rodolfo Ipolito Menequette, Antonio A.F. Loureiro, and Leandro Aparecido Villas. Decreasing greenhouse emissions through an intelligent traffic information system based on inter-vehicle communication. In *Proceedings of the 12th ACM International Symposium on Mobility Management and Wireless Access, MobiWac '14*, pages 91–98, New York, NY, USA, 2014. ACM.
- [30] A.M. Souza, G. Maia, and L.A. Villas. Add: A data dissemination solution for highly dynamic highway environments. In *Network Computing and Applications (NCA), 2014 IEEE 13th International Symposium on*, pages 17–23, Aug 2014.
- [31] Ozan K. Tonguz, Nawaporn Wisitpongphan, and Fan Bai. DV-CAST: a distributed vehicular broadcast protocol for vehicular ad hoc networks. *IEEE Wireless Communications*, 17(2):47–56, 2010.
- [32] Y. Toor, P. Muhlethaler, and A. Laouiti. Vehicle ad hoc networks: applications and related technical issues. *Communications Surveys Tutorials, IEEE*, 10(3):74–88, Third 2008.
- [33] L.A. Villas, A. Boukerche, R.B. Araujo, A.A.F. Loureiro, and J. Ueyama. Network partition-aware geographical data dissemination. In *Communications (ICC), 2013 IEEE International Conference on*, pages 1439–1443, 2013.
- [34] Leandro Aparecido Villas, Azzedine Boukerche, Guilherme Maia, Richard Werner Pazzi, and Antonio A.F. Loureiro. Drive: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks. *Computer Networks*, 75, Part A(0):381 – 394, 2014.
- [35] M. Wang, H. Shan, R. Lu, R. Zhang, X. Shen, and F. Bai. Real-time path planning based on hybrid-vanet-enhanced transportation system. *Vehicular Technology, IEEE Transactions on*, May 2015.

- [36] WAZE. Waze app bugs and issues.
- [37] Axel Wegener, Michał Piórkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean-Pierre Hubaux. Traci: an interface for coupling road traffic and network simulators. In *Proceedings of the 11th communications and networking simulation symposium*, pages 155–163. ACM, 2008.
- [38] N. Wisitpongphan, O.K. Tonguz, J.S. Parikh, P. Mudalige, F. Bai, and V. Sadekar. Broadcast storm mitigation techniques in vehicular ad hoc networks. *Wireless Communications, IEEE*, 14(6):84–94, December 2007.
- [39] S. Yousefi, M.S. Mousavi, and M. Fathy. Vehicular ad hoc networks (vanets): Challenges and perspectives. In *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 761–766, June 2006.
- [40] Ting Zhong, B. Xu, and O. Wolfson. Disseminating real-time traffic information in vehicular ad-hoc networks. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 1056–1061, June 2008.