

UNIVERSIDADE FEDERAL DO PARANÁ

VICTOR ANTONIO GIL TOSIN

ANÁLISE DE IMPACTO NO CONGESTIONAMENTO DE CANAL DA MENSAGEM  
CPM EM UMA REDE V2X

CURITIBA

2025

VICTOR ANTONIO GIL TOSIN

ANÁLISE DE IMPACTO NO CONGESTIONAMENTO DE CANAL DA MENSAGEM  
CPM EM UMA REDE V2X

Trabalho de Conclusão de Curso apresentado ao curso de Graduação em Engenharia Elétrica com ênfase em Eletrônica e Telecomunicações na Universidade Federal do Paraná, como requisito parcial à obtenção do título de Bacharel em Engenharia Elétrica

Orientador: Prof. Dr. Evelio Martín García Fernández

Coorientador: M.Sc. Leonardo Barbosa da Silva

CURITIBA

2025

## TERMO DE APROVAÇÃO

VICTOR ANTONIO GIL TOSIN

### ANÁLISE DE IMPACTO NO CONGESTIONAMENTO DE CANAL DA MENSAGEM CPM EM UMA REDE V2X

Trabalho de Conclusão de Curso apresentado ao curso de Graduação em Engenharia Elétrica com ênfase em Eletrônica e Telecomunicações na Universidade Federal do Paraná, como requisito parcial à obtenção do título de Bacharel em Engenharia Elétrica, pela seguinte banca examinadora:

---

Prof Dr. Evelio Martín García Fernández  
Orientador – Departamento de Engenharia Elétrica, UFPR

---

Prof. Dr. César Augusto Dartora  
Convidado – Departamento de Engenharia Elétrica, UFPR

---

M.Sc. Diogo Rikio Miyazaki  
Convidado – Departamento de Engenharia Elétrica, UFPR

Curitiba, 3 de julho de 2025.

## **AGRADECIMENTOS**

Dedico esse trabalho aos meus pais, que sempre me apoiaram na minha trajetória e me deram a sustentação para eu sempre alcançar os meus sonhos.

Agradeço aos meus orientadores, Prof. Dr. Evelio Martín García Fernández e M.Sc. Leonardo Barbosa da Silva, por me derem essa oportunidade de estudar um tema inovador e me conduzirem nesse trabalho.

Agradeço aos meus amigos que estiveram comigo nessa jornada da faculdade, pelos momentos de descontração e de solidariedade.

A Nathalia Dalnegro, por estar sempre ao meu lado, me apoiando, me encorajando e por ser o meu refúgio.

## RESUMO

Este trabalho tem como objetivo analisar o impacto da transmissão de mensagens CPM (Collective Perception Message) no congestionamento do canal de comunicação em redes V2X, com base na arquitetura ETSI ITS-G5. Foram desenvolvidos quatro cenários de simulação com diferentes densidades de nós simultâneos, utilizando as ferramentas SUMO, OMNeT++ e o framework Artery. As simulações permitiram a coleta de métricas como latência, jitter, throughput e PER, possibilitando a avaliação do desempenho da rede em ambientes urbanos com alta mobilidade. Os resultados mostraram que o aumento no número de nós impacta diretamente a latência máxima e a taxa de erro de pacotes, mas sem ultrapassar os limites aceitáveis definidos pelas normas técnicas. O estudo também evidenciou a viabilidade da aplicação das mensagens CPM em cenários com densidade moderada a alta, desde que haja controle eficiente do acesso ao meio.

Palavras-chave: V2X, CPM, comunicação veicular, ITS-G5, simulação de redes.

## **ABSTRACT**

This work aims to analyze the impact of CPM (Collective Perception Message) transmission on channel congestion in V2X networks, based on the ETSI ITS-G5 architecture. Four simulation scenarios with different densities of simultaneous nodes were developed using the SUMO, OMNeT++, and Artery tools. The simulations allowed the collection of performance metrics such as latency, jitter, throughput, and PER, enabling the evaluation of network behavior in urban environments with high mobility. The results showed that increasing the number of nodes directly affects maximum latency and packet error rate, although remaining within acceptable technical limits. The study also demonstrated the feasibility of CPM message usage in medium to high-density scenarios, provided that efficient channel access control is in place.

Keywords: V2X, CPM, vehicular communication, ITS-G5, network simulation.

## **LISTA DE ABREVIATURAS OU SIGLAS**

ADHOC – Ad hoc (rede temporária sem infraestrutura fixa)

ARTERY – Framework de simulação para V2X baseado em OMNeT++

BPSK – Binary Phase Shift Keying

BTP – Basic Transport Protocol

CAM – Cooperative Awareness Message

CPM – Collective Perception Message

DENM – Decentralized Environmental Notification Message

EDCA – Enhanced Distributed Channel Access

ETSI – European Telecommunications Standards Institute

GUI – Graphical User Interface

HMI – Human-Machine Interface

ID – Identificador

IEEE – Institute of Electrical and Electronics Engineers

INET – Módulo de simulação de redes para OMNeT++

IP – Internet Protocol

ITS – Intelligent Transportation Systems

IVC – Inter-Vehicle Communication

LTE – Long Term Evolution

MAC – Medium Access Control

OMNET++ – Simulador de eventos discretos orientado a módulos

OSI – Open Systems Interconnection

OSM – OpenStreetMap

PER – Packet Error Rate

PHY – Physical Layer

QPSK – Quadrature Phase Shift Keying

RSU – Road Side Unit

TCP – Transmission Control Protocol

THROUGHPUT – Vazão efetiva de dados

UDP – User Datagram Protocol

VANET – Vehicular Ad Hoc Network

VEINS – Vehicles in Network Simulation

V2X – Vehicle-to-Everything

VRU – Vulnerable Road Users

XML – Extensible Markup Language

## LISTA DE TABELAS

|                                       |    |
|---------------------------------------|----|
| TABELA 1 : RESULTADOS CENÁRIO 1 ..... | 43 |
| TABELA 2 : RESULTADOS CENÁRIO 2 ..... | 46 |
| TABELA 3 : RESULTADOS CENÁRIO 3 ..... | 49 |
| TABELA 4 : RESULTADOS CENÁRIO 4 ..... | 51 |
| TABELA 5 : COMPARAÇÃO LATÊNCIA .....  | 54 |
| TABELA 6 : COMPARAÇÃO JITTER .....    | 55 |
| TABELA 7 : COMPARAÇÃO TROUGHPUT ..... | 55 |
| TABELA 8 : COMPARAÇÃO PER .....       | 56 |

## LISTA DE FIGURAS

|                                                   |    |
|---------------------------------------------------|----|
| FIGURA 1 : MODELO OSI.....                        | 16 |
| FIGURA 2 : CAMADAS ITS-G5 .....                   | 20 |
| FIGURA 3 : ESTRUTURA CPM.....                     | 23 |
| FIGURA 4 : INTERFACE DO NETEDIT .....             | 28 |
| FIGURA 5 : SUMO-GUI .....                         | 29 |
| FIGURA 6 : INTERFACE DA SIMULAÇÃO.....            | 30 |
| FIGURA 7 : FLUXOGRAMA ARQUITETURA ARTERY V2X..... | 32 |
| FIGURA 8 : INTERFACE OSM WEB WIZARD.....          | 33 |
| FIGURA 9 : MAPA CARREGADO NO SUMO .....           | 34 |
| FIGURA 10 : ROTAS NO NETEDIT .....                | 35 |
| FIGURA 11 : SIMULAÇÃO NO SUMO.....                | 36 |
| FIGURA 12 : SENSOR.XML .....                      | 37 |
| FIGURA 13 : CONFIGURAÇÃO DE COMUNICAÇÃO .....     | 38 |
| FIGURA 14 : CONFIGURAÇÃO DO PATH LOSS.....        | 38 |
| FIGURA 15 : CONFIGURAÇÃO DOS SENSORES .....       | 39 |
| FIGURA 16 : ARQUIVOS PARA A SIMULAÇÃO .....       | 40 |
| FIGURA 17 : SIMULAÇÃO NO OMNET++ .....            | 41 |
| FIGURA 18 : LOG DE TRANSMISSÃO .....              | 41 |
| FIGURA 19 : LOG MENSAGEM CPM.....                 | 41 |
| FIGURA 20 : LATÊNCIA CENÁRIO 1 .....              | 44 |
| FIGURA 21 : JITTER CENÁRIO 1 .....                | 45 |
| FIGURA 22 : TROUGHPUT CENÁRIO 1 .....             | 45 |
| FIGURA 23 : PER CENÁRIO 1 .....                   | 46 |
| FIGURA 24 : LATÊNCIA CENÁRIO 2 .....              | 47 |
| FIGURA 25 : JITTER CENÁRIO 2 .....                | 47 |
| FIGURA 26 : TROUGHPUT CENÁRIO 2 .....             | 48 |
| FIGURA 27 : PER CENÁRIO 2 .....                   | 48 |
| FIGURA 28 : LATÊNCIA CENÁRIO 3 .....              | 49 |
| FIGURA 29 : JITTER CENÁRIO 3 .....                | 50 |
| FIGURA 30 : TROUGHPUT CENÁRIO 3 .....             | 50 |
| FIGURA 31 : PER CENÁRIO 3 .....                   | 51 |

|                                           |    |
|-------------------------------------------|----|
| FIGURA 32 : LATÊNCIA CENÁRIO 4 .....      | 52 |
| FIGURA 33 : JITTER CENÁRIO 4 .....        | 52 |
| FIGURA 34 : TROUGHPUT CENÁRIO 4 .....     | 53 |
| FIGURA 35 : PER CENÁRIO 4 .....           | 53 |
| FIGURA 36 : COMPARAÇÃO DE LATÊNCIA.....   | 54 |
| FIGURA 37 : COMPARAÇÃO DE JITTER .....    | 55 |
| FIGURA 38 : COMPARAÇÃO DE TROUGHPUT ..... | 56 |
| FIGURA 39 : COMPARAÇÃO DE PER.....        | 57 |

## SUMÁRIO

|       |                                                 |    |
|-------|-------------------------------------------------|----|
| 1     | INTRODUÇÃO .....                                | 14 |
| 1.1   | OBJETIVOS .....                                 | 15 |
| 1.1.1 | OBJETIVO GERAL.....                             | 15 |
| 1.1.2 | OBJETIVO ESPECÍFICO .....                       | 15 |
| 2     | FUNDAMENTAÇÃO TEÓRICA .....                     | 16 |
| 2.1   | MODELO OSI.....                                 | 16 |
| 2.2   | REDES ADHOC E VANET .....                       | 18 |
| 2.3   | IEEE 802.11P .....                              | 19 |
| 2.4   | ARQUITETURA ETSI ITS-G5.....                    | 19 |
| 2.5   | MÉTRICAS DE AVALIAÇÃO DE DESEMPENHO .....       | 24 |
| 2.5.1 | LATÊNCIA.....                                   | 24 |
| 2.5.2 | JITTER .....                                    | 25 |
| 2.5.3 | TROUGHPUT.....                                  | 25 |
| 2.5.4 | PER (PACKET ERROR RATE) .....                   | 25 |
| 2.6   | MODELO DE PROPAGAÇÃO LOG-NORMAL SHADOWING ..... | 26 |
| 3     | MATERIAIS E MÉTODOS.....                        | 27 |
| 3.1   | FERRAMENTAS UTILIZADAS.....                     | 27 |
| 3.1.1 | SIMULAÇÃO VEICULAR (SUMO).....                  | 27 |
| 3.1.2 | SIMULAÇÃO DE REDES (OMNET++) .....              | 29 |
| 3.1.3 | FRAMEWORK ARTERY V2X.....                       | 31 |
| 3.2   | CONSTRUÇÃO DO CENÁRIO VEICULAR.....             | 33 |
| 3.3   | CONFIGURAÇÃO DO CENÁRIO DE COMUNICAÇÃO .....    | 36 |
| 3.4   | SIMULAÇÃO .....                                 | 40 |
| 3.5   | MÉTRICAS.....                                   | 42 |
| 4     | RESULTADOS .....                                | 43 |
| 4.1   | CENÁRIO 1 – 18 NÓS SIMULTÂNEOS .....            | 43 |

|       |                                       |    |
|-------|---------------------------------------|----|
| 4.2   | CENÁRIO 2 – 61 NÓS SIMULTÂNEOS .....  | 46 |
| 4.3   | CENÁRIO 3 – 116 NÓS SIMULTÂNEOS ..... | 48 |
| 4.4   | CENÁRIO 4 – 161 NÓS SIMULTÂNEOS ..... | 51 |
| 4.5   | COMPARAÇÃO ENTRE CENÁRIOS .....       | 54 |
| 4.5.1 | LATÊNCIA.....                         | 54 |
| 4.5.2 | JITTER .....                          | 54 |
| 4.5.3 | THROUGHPUT .....                      | 55 |
| 4.5.4 | PER.....                              | 56 |
| 5     | CONCLUSÃO .....                       | 58 |
| 6     | SUGESTÕES DE TRABALHOS FUTUROS.....   | 60 |
| 7     | BIBLIOGRAFIA .....                    | 61 |

## 1 INTRODUÇÃO

A evolução dos sistemas de transporte inteligente tem impulsionado o desenvolvimento de tecnologias de comunicação voltadas para veículos. Nesse contexto, destaca-se a comunicação V2X (*Vehicle-to-Everything*), que permite a troca de informações entre veículos e diferentes elementos do ambiente. Essa comunicação pode ocorrer entre veículos (V2V), entre veículos e infraestrutura (V2I), com pedestres (V2P) ou com redes externas (V2N), viabilizando aplicações que aumentam a segurança viária, otimizam o fluxo do tráfego e reduzem o número de acidentes.

Cada tipo de comunicação possui aplicações específicas. No caso da V2V, os veículos compartilham dados como posição, velocidade e direção para prevenir colisões e melhorar a consciência situacional. Na comunicação V2I, a interação com semáforos ou sinais inteligentes permite ajustes dinâmicos no tráfego. A V2P permite a detecção de pedestres com dispositivos conectados, enquanto a V2N viabiliza acesso a serviços baseados em nuvem, como atualizações de tráfego em tempo real.

Apesar dos avanços, ainda existem desafios técnicos relacionados à confiabilidade, desempenho e escalabilidade dessas redes. Um dos principais entraves está no congestionamento do canal de comunicação, especialmente em cenários densos e com alto volume de mensagens cooperativas. A introdução das mensagens CPM (Collective Perception Message), que aumentam significativamente o volume de dados transmitidos por veículo, exige uma análise mais profunda sobre seu impacto na rede.

O congestionamento do canal pode gerar diversos efeitos negativos, como o aumento da latência devido ao tempo adicional de espera para acesso ao meio, a elevação do jitter pela variação imprevisível nos atrasos de entrega, a redução do throughput efetivo causada por colisões, além do aumento na taxa de perda de pacotes (PER) quando o canal atinge saturação. Assim, este trabalho tem como objetivo avaliar os efeitos da transmissão de mensagens CPM em cenários V2X, com foco nas métricas de desempenho citadas anteriormente.

## 1.1 OBJETIVOS

### 1.1.1 OBJETIVO GERAL

O objetivo geral deste trabalho é analisar o impacto da transmissão de CPM no congestionamento de canal em redes V2X, avaliando como fatores influenciam a latência e perda de pacotes. Para isso, será criado um cenário de simulação baseado em uma parcela específica de Curitiba, utilizando softwares SUMO, OMNeT++ e Artery V2X, permitindo simular redes veiculares em tráfegos urbanos dinâmicos, onde o congestionamento de canal pode ser crítico.

### 1.1.2 OBJETIVO ESPECÍFICO

- Estudar e implementar as normas da ETSI (European Telecommunications Standards Institute) para redes V2X;
- Construir um ambiente em que seja possível realizar as simulações
- Recriar um cenário de tráfego urbano com envio de mensagens CPM;
- Avaliar impactos na transmissão e recepção da CPM nesse cenário.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 MODELO OSI

O modelo OSI (Open Systems Interconnection) é uma referência teórica amplamente utilizada para descrever como diferentes sistemas de comunicação interagem em uma rede. Ele é estruturado em sete camadas, sendo cada uma responsável por uma função específica que garante a padronização e eficiência da comunicação.

FIGURA 1 : MODELO OSI

|                 |
|-----------------|
| Aplicação       |
| Apresentação    |
| Sessão          |
| Transporte      |
| Rede            |
| Enlace de dados |
| Física          |

FONTE: O autor (2024)

A camada física lida com o meio de transmissão, como cabos e meios de transmissão sem fio. É nessa camada que ocorre a conversão dos bits em sinais eletromagnéticos ou ópticos, permitindo que a comunicação de fato aconteça entre os dispositivos. Ela define características como o tipo de conector, modulação e potência do sinal.

A camada de enlace gerencia conexões confiáveis entre dispositivos próximos. Ela divide os dados em quadros e trata de erros que possam ocorrer durante a transmissão, garantindo que apenas informações corretas cheguem ao próximo nó da rede. Em serviços confiáveis, o receptor envia uma confirmação de recebimento (acknowledgement), mas esse processo ocorre apenas entre dispositivos diretamente conectados, ou seja, em comunicação ponto a ponto. A camada também evita que um transmissor rápido sobrecarregue um receptor lento, por meio de controle de fluxo.

Dentro dessa camada, a subcamada MAC (Medium Access Control) é responsável por controlar o uso do meio de transmissão compartilhado, definindo

quando cada dispositivo pode transmitir e ajudando a evitar colisões. Ela também pode aplicar prioridades no tráfego, o que é especialmente importante em redes V2X, onde mensagens de segurança devem ser tratadas com maior urgência.

Enquanto isso, a camada de rede cuida do roteamento e entrega de pacotes entre redes distintas. Ela determina qual o melhor caminho para que um pacote chegue ao seu destino, mesmo que envolva múltiplas redes intermediárias. Protocolos como o IP operam nessa camada

O protocolo IP (Internet Protocol) é o principal protocolo da camada de rede e é responsável por identificar cada dispositivo conectado à internet por meio de um endereço lógico único. Ele define o formato dos pacotes e trata do encaminhamento desses pacotes de dados entre as redes, permitindo que a informação percorra diferentes caminhos até chegar ao destino.

Já a camada de transporte assegura a entrega confiável de dados entre dispositivos finais. Ela fragmenta as mensagens em segmentos, controla o fluxo, reordena os pacotes quando necessário e garante que tudo chegue sem erros. É onde atuam protocolos como o TCP e o UDP

O TCP funciona estabelecendo uma conexão entre os dois dispositivos antes de começar a troca de dados. Ele divide as informações em segmentos, numera cada um deles e garante que todos cheguem ao destino na ordem correta. Se algum segmento se perder ou chegar com erro, ele é reenviado. Além disso, o TCP controla o fluxo de dados para evitar sobrecarga no receptor. Já o UDP envia os dados diretamente, sem criar uma conexão prévia e sem garantir que os pacotes cheguem corretamente ou na ordem certa. Isso o torna mais leve e rápido, mas menos confiável, não há reenvio em caso de perda.

A camada de sessão é responsável por estabelecer, manter e finalizar sessões de comunicação entre aplicações. Ela organiza o diálogo entre sistemas, controlando quem envia e quem recebe em cada momento, garantindo uma comunicação ordenada. Esse controle de diálogo permite que os processos se comuniquem em modo half-duplex, onde os dados fluem em apenas um sentido por vez, ou em modo full-duplex, com transmissão nos dois sentidos simultaneamente. Além disso, essa camada fornece mecanismos de sincronização, permitindo que pontos de verificação

(checkpoints) sejam inseridos no fluxo de dados. Isso facilita a retomada da transmissão em caso de falha, sem necessidade de reiniciar toda a comunicação.

A camada de apresentação é responsável por preparar os dados para que possam ser interpretados corretamente pela aplicação no destino. Ela cuida da tradução entre diferentes formatos de codificação, garantindo que emissor e receptor compreendam as informações, mesmo que utilizem representações distintas. Também realiza a compressão de dados, otimizando a quantidade de informações que trafegam pela rede. Outra função importante é a criptografia, que protege os dados durante a transmissão, assegurando confidencialidade e segurança na comunicação.

A camada de aplicação é o ponto onde os usuários e os programas interagem diretamente com a rede. Ela fornece os serviços que permitem o envio e recebimento de dados entre sistemas, como acesso à web, e-mail, transferência de arquivos e comunicação entre aplicações distribuídas. É aqui que as informações são geradas ou consumidas, e onde ocorre a interface entre o software do usuário e os serviços oferecidos pela rede.

## 2.2 REDES ADHOC E VANET

As redes Ad Hoc são um tipo de arquitetura de comunicação descentralizada, onde os dispositivos se conectam diretamente entre si sem depender de uma infraestrutura fixa, como roteadores ou pontos de acesso. Cada nó participante atua como transmissor e roteador, encaminhando pacotes até o seu destino. Essas redes são altamente dinâmicas e se adaptam constantemente à entrada e saída de nós, o que as torna ideais para situações em que a infraestrutura convencional está ausente ou indisponível.

Nesse contexto, surge a VANET (Vehicular Ad Hoc Network), um tipo específico de rede Ad Hoc voltada para a comunicação entre veículos, ampliando o conceito para aplicações de mobilidade inteligente e segurança no trânsito. Nessa configuração, os veículos desempenham papéis de hosts ou roteadores móveis, formando uma rede temporária que acompanha o movimento do tráfego.

### 2.3 IEEE 802.11P

O IEEE 802.11p foi desenvolvido para atender às exigências da comunicação em redes veiculares, lidando com a alta mobilidade e a baixa latência exigidas nesses ambientes. Ele opera na faixa de 5,9 GHz e utiliza canais de 10 MHz, mais estreitos que os canais tradicionais do Wi-Fi, com espaçamento entre subportadoras de 156,25 kHz. Essa configuração reduz os efeitos do espalhamento de atraso e melhora a robustez do sinal em ambientes com grande variação de velocidade e distância entre os nós.

Na camada física, o 802.11p utiliza a técnica de modulação OFDM (Orthogonal Frequency Division Multiplexing), que divide o sinal em várias subportadoras ortogonais, transmitidas simultaneamente. Isso permite uma transmissão mais eficiente em canais com múltiplos caminhos de propagação, como ocorre em ambientes urbanos e rodoviários. O OFDM oferece resistência a interferências e ao efeito Doppler.

Na camada de enlace, o acesso ao meio é controlado pelo mecanismo EDCA (Enhanced Distributed Channel Access), que organiza o tráfego em diferentes categorias de prioridade. Cada categoria possui parâmetros específicos de acesso ao canal, permitindo que mensagens mais importantes, como alertas de segurança, sejam transmitidas antes de dados menos sensíveis ao tempo. Esse controle é feito de forma distribuída, sem necessidade de coordenação centralizada.

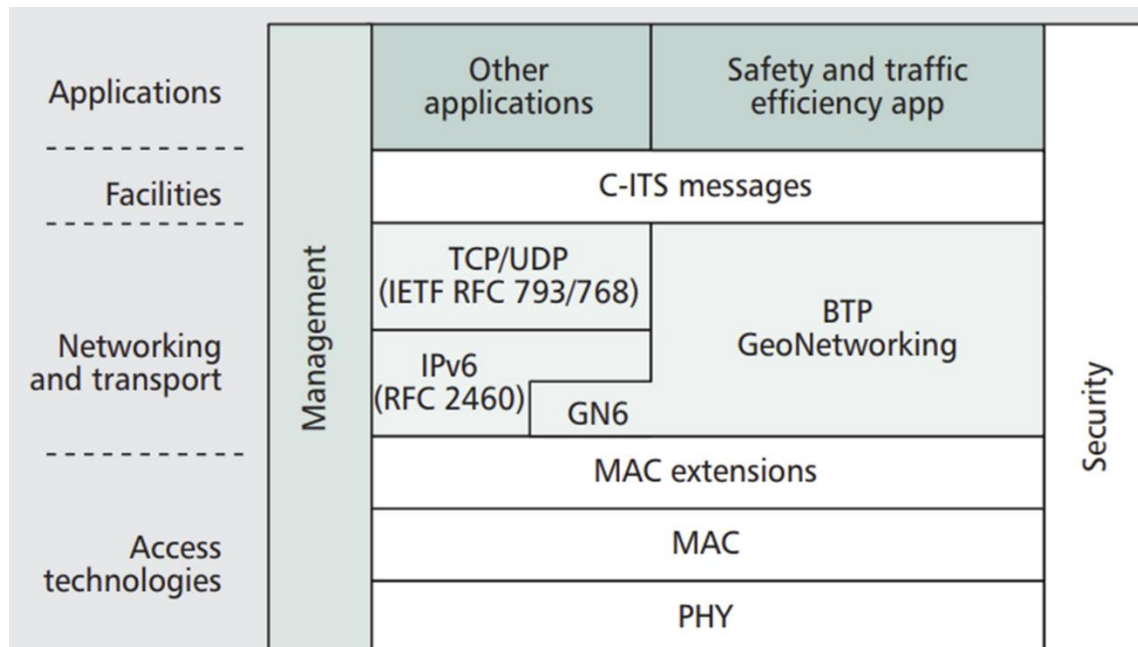
Uma característica marcante do 802.11p é o funcionamento no chamado modo OCB (Outside the Context of a Basic Service Set). Nesse modo, os dispositivos não precisam realizar os procedimentos tradicionais de autenticação e associação antes de trocar dados. Isso permite que os nós iniciem a comunicação de forma imediata, sem negociação prévia, o que é essencial em redes veiculares onde a conexão entre dois dispositivos pode durar apenas alguns segundos.

### 2.4 ARQUITETURA ETSI ITS-G5

O ITS-G5 é uma tecnologia de comunicação desenvolvida especificamente para redes veiculares na Europa, sendo parte dos Sistemas de Transporte Inteligentes (ITS). Baseado no padrão IEEE 802.11p, o ITS-G5 foi adaptado pelo Instituto Europeu

de Normas de Telecomunicações (ETSI) para atender aos requisitos regionais, com foco em aplicações voltadas à segurança viária e eficiência do tráfego.

FIGURA 2 : CAMADAS ITS-G5



FONTE: Adaptado de Špitálová [11] (2022)

Assim como no modelo OSI, a arquitetura do ITS-G5 é organizada em camadas, onde cada uma executa funções específicas na troca de dados entre veículos e infraestrutura. Essa organização modular facilita a padronização e a integração entre sistemas de diferentes fabricantes.

Na camada física (PHY), ocorrem a modulação e a transmissão dos sinais de rádio, definindo a frequência de operação, largura de banda e demais parâmetros de propagação. Essa camada garante a comunicação mesmo em cenários com alta mobilidade e mudanças rápidas de topologia. Acima dela, a camada MAC (Medium Access Control) gerencia o acesso ao canal compartilhado, aplicando mecanismos que evitam colisões e organizam o tráfego entre múltiplos dispositivos. As chamadas MAC extensions ampliam essa funcionalidade para lidar com requisitos mais exigentes em termos de qualidade de serviço e densidade de nós.

Na camada de rede e transporte (Networking and Transport), o ITS-G5 utiliza múltiplos protocolos que operam em paralelo para diferentes tipos de comunicação.

O GeoNetworking (GN) é responsável pelo endereçamento geográfico e pelo roteamento baseado em posição. Ao contrário das redes tradicionais, que utilizam endereços vinculados à identidade dos nós (como IP), o GeoNetworking permite que pacotes sejam enviados para um ponto geográfico específico ou para todos os nós dentro de uma área. Cada nó mantém uma visão parcial da topologia local e, ao receber um pacote, compara o endereço geográfico de destino com sua posição e a dos vizinhos. A decisão de repasse é feita de forma autônoma e instantânea, sem necessidade de tabelas de roteamento, o que torna o protocolo eficiente em ambientes com mobilidade constante.

Já o BTP (Basic Transport Protocol) fornece um serviço de transporte sem conexão e de extremidade a extremidade, funcionando de forma semelhante ao UDP, conforme definido na norma ETSI EN 302 636-3. Ele oferece um serviço mínimo, com entrega não garantida de unidades de dados (BTP-PDUs), o que o torna leve e adequado para aplicações que priorizam tempo de resposta. Além disso, o BTP permite que as entidades da camada Facilities acessem diretamente os serviços do GeoNetworking, criando uma integração eficiente entre as camadas superiores e o roteamento geográfico subjacente.

Para comunicações baseadas em IP, o ITS-G5 também oferece suporte ao IPv6, utilizando o GN6, um mecanismo de encapsulamento que permite transportar pacotes IPv6 sobre o GeoNetworking. Isso permite interoperabilidade com redes IP convencionais e o uso de protocolos como TCP e UDP quando necessário.

A camada Facilities fornece suporte direto às aplicações da arquitetura ITS-G5. Ela é responsável por processar e gerenciar as chamadas mensagens C-ITS (Cooperative Intelligent Transport Systems), que transmitem informações relacionadas à segurança viária, eficiência do tráfego e percepção do ambiente. Essa camada atua como intermediária entre os protocolos de transporte/georreferenciamento e as aplicações finais, organizando os dados recebidos e transmitidos conforme o tipo de serviço oferecido.

Entre as mensagens C-ITS mais utilizadas está a CAM (Cooperative Awareness Message), enviada periodicamente pelos veículos para informar seu estado dinâmico e estático. Essas mensagens contêm dados como posição,

velocidade, aceleração, direção, tipo de veículo e status de dispositivos como luzes de freio ou setas. O objetivo da CAM é manter a consciência situacional cooperativa entre veículos e infraestrutura, permitindo que cada nó tenha conhecimento em tempo real sobre os demais participantes próximos na rede. Isso reduz o risco de colisões e melhora a coordenação do tráfego em ambientes urbanos e rodoviários.

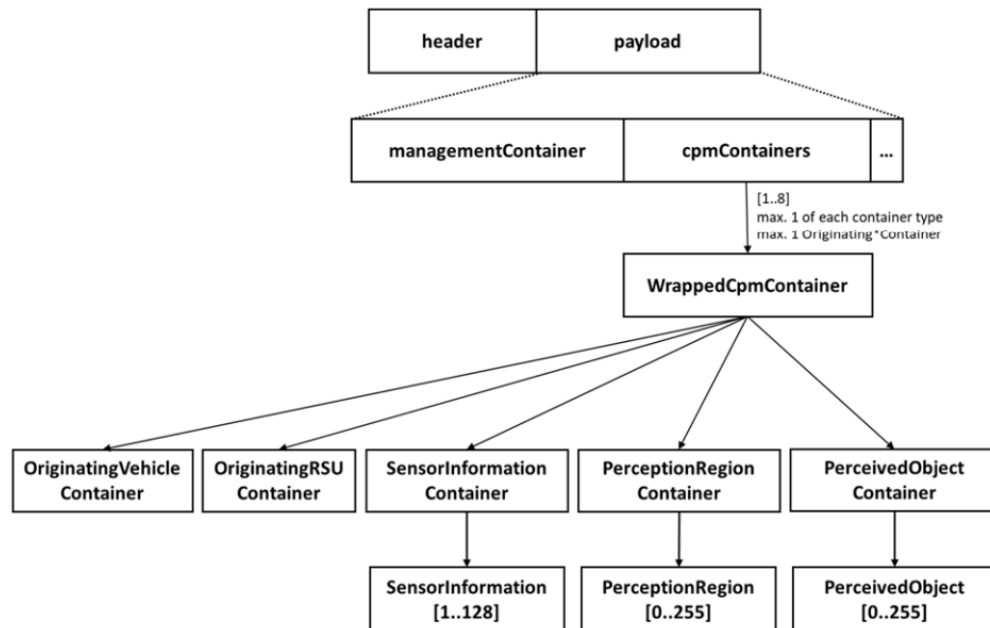
Outra mensagem fundamental da camada Facilities é a CPM (Collective Perception Message). Diferente da CAM, que descreve o próprio veículo, a CPM transmite informações sobre objetos percebidos no entorno, como veículos, pedestres, ciclistas, animais ou obstáculos fixos, mesmo que não estejam equipados com V2X. Além disso, ela também pode descrever regiões desocupadas da via, permitindo que outros veículos determinem áreas livres para manobra ou avanço. Os dados são obtidos a partir dos sensores embarcados no veículo transmissor, como câmeras, radares ou LiDARs, e podem incluir posição, tempo de detecção, velocidade estimada, tipo de objeto e até características do sensor, como campo de visão e tipo.

Ao receber uma CPM, o veículo destinatário passa a ter conhecimento da presença e do estado de objetos ou regiões detectadas por outro ITS-S (Estação de Transporte Inteligente). Como diferentes veículos podem detectar um mesmo objeto por ângulos e sensores distintos, a rede V2X pode realizar fusão de dados, combinando múltiplas percepções para reduzir incertezas quanto ao tamanho, posição ou movimento do objeto. Isso torna a percepção do ambiente mais precisa e confiável, mesmo em casos em que alguma fonte forneça dados corrompidos ou imprecisos. Esse tipo de compartilhamento é especialmente útil em cruzamentos, curvas cegas e áreas com baixa visibilidade, e tem aplicação direta em sistemas de assistência ao condutor, controle autônomo e prevenção de colisões.

A estrutura geral de uma CPM (Collective Perception Message) pode ser observada na FIGURA 3, que ilustra a organização dos principais componentes da mensagem conforme especificado pela norma ETSI TS 103 324 V2.1.1. Essa estrutura é projetada para permitir a comunicação eficiente de informações sensoriais complexas entre veículos e unidades de infraestrutura, com flexibilidade para representar diversos tipos de objetos e sensores em uma única mensagem padronizada.

FIGURA 3 : ESTRUTURA CPM

The general structure of a CPM is illustrated in Figure 7.



FONTE: ETSI (2023)

A mensagem é composta por duas seções principais: o header e o payload. O header contém informações comuns a todas as mensagens da camada Facilities, como tipo, versão e identificadores. O payload, por sua vez, é dividido em dois blocos: o managementContainer, que inclui dados básicos necessários para interpretar a mensagem, como tempo de envio e posição do transmissor; e o cpmContainers, onde estão localizados os demais dados agrupados sob a estrutura chamada WrappedCpmContainer. Essa estrutura permite incluir de 1 até 8 containers de diferentes tipos, garantindo flexibilidade e expansão futura do formato.

A Como mostrado na FIGURA 3, cada WrappedCpmContainer pode conter um dos seguintes blocos:

- O OriginatingVehicleContainer, presente quando a CPM é enviada por um veículo, contendo dados como orientação e eventuais reboques;
- O OriginatingRSUContainer, usado se a mensagem for originada por uma unidade de beira de estrada (RSU);

- O `SensorInformationContainer`, que descreve os sensores embarcados e suas capacidades;
- O `PerceptionRegionContainer`, que indica áreas de detecção parcial ou limitada;
- `PerceivedObjectContainer`, que descreve os objetos detectados, incluindo dados como tempo de detecção, posição e atributos opcionais como tamanho, velocidade e tipo. Cada mensagem deve conter obrigatoriamente um único container de origem (veículo ou RSU) e pode incluir até 255 objetos percebidos e 255 regiões.

A camada de aplicação da arquitetura ITS-G5 é responsável por utilizar as informações processadas pelas camadas inferiores para fornecer funcionalidades diretamente ligadas à segurança viária, gestão do tráfego e suporte à condução. Ela abriga as aplicações finais, como alerta de colisão frontal, assistência em cruzamentos, notificações de obras na pista, gestão de velocidade recomendada e aviso de presença de pedestres. Essas aplicações consomem dados provenientes das mensagens processadas pela camada `Facilities`, e podem interagir diretamente com o condutor por meio de sistemas HMI (Human-Machine Interface) ou com módulos de controle do veículo, em aplicações automatizadas. A camada de aplicação é também onde novas funcionalidades ITS podem ser implementadas, com foco em mobilidade inteligente, redução de acidentes e eficiência energética no transporte.

## 2.5 MÉTRICAS DE AVALIAÇÃO DE DESEMPENHO

### 2.5.1 LATÊNCIA

A latência é o tempo necessário para que uma mensagem percorra o caminho entre o emissor e o receptor em uma rede. Essa métrica reflete a capacidade de resposta do sistema e é influenciada por fatores como tempo de processamento, enfileiramento de pacotes, acesso ao meio, distância percorrida e congestionamento do canal.

Nas simulações, a latência é geralmente medida ponto a ponto, como a diferença entre o momento de envio e o instante de recebimento da mensagem. Altos

valores indicam atrasos na entrega e podem comprometer o desempenho de aplicações que exigem respostas rápidas.

### 2.5.2 JITTER

O jitter é a variação no tempo de entrega entre pacotes consecutivos em uma rede. Mesmo que a latência média seja aceitável, a flutuação entre os atrasos pode comprometer aplicações que exigem regularidade na transmissão. Em termos estatísticos, essa variação pode ser representada pelo desvio padrão dos tempos de chegada ou dos atrasos, servindo como indicador da instabilidade temporal da rede.

$$Jitter(i) = |D(i + 1) - D(i)|, (1)$$

Essa equação calcula o jitter instantâneo como o valor absoluto da diferença entre os atrasos de dois pacotes sucessivos. Valores elevados indicam maior imprevisibilidade na entrega das mensagens, geralmente causada por congestionamento do canal, contenção de acesso ou processamento irregular nos nós da rede.

### 2.5.3 TROUGHPUT

O throughput representa a taxa efetiva de transmissão de dados em uma rede, ou seja, a quantidade de informação útil entregue com sucesso por unidade de tempo. Essa métrica reflete a capacidade real da rede em transportar dados, considerando perdas, atrasos e retransmissões.

Em simulações, o throughput é geralmente calculado com base no total de dados recebidos divididos pelo tempo de observação. Valores elevados indicam bom aproveitamento do canal, enquanto reduções podem sinalizar congestionamento, interferências ou limitações no protocolo de acesso.

### 2.5.4 PER (PACKET ERROR RATE)

A taxa de erro de pacotes (PER) mede a proporção de pacotes transmitidos que não foram recebidos corretamente. Essa métrica reflete a confiabilidade da comunicação em nível de pacote e é impactada por perdas causadas por interferência, colisões ou degradação do sinal.

$$PER = \frac{P_{perdidos}}{P_{total}} \times 100, (2)$$

$P_{perdidos}$  é o número de pacotes que foram descartados ou corrompidos, e  $P_{total}$  é o total de pacotes transmitidos. Um PER elevado pode indicar problemas no canal ou sobrecarga na rede, comprometendo o desempenho de aplicações sensíveis à perda de dados.

## 2.6 MODELO DE PROPAGAÇÃO LOG-NORMAL SHADOWING

O modelo Log-Normal Shadowing é utilizado para representar a perda média de sinal em ambientes onde há obstáculos e variações imprevisíveis na propagação. Ele combina o decaimento da potência com a distância e uma componente aleatória que simula os efeitos de sombreamento causados por edifícios, veículos e outros elementos do ambiente.

$$P_L(d)_{(dB)} = P_L(d0)_{(dB)} + 10n \log_{10}\left(\frac{d}{d0}\right) + \psi, (3)$$

$P_L(d)_{(dB)}$  é a perda de percurso a uma distância  $d$ ,  $P_L(d0)_{(dB)}$  é a perda em uma distância de referência  $d0$ ,  $n$  é o expoente de perda, e  $\psi$  é uma variável aleatória gaussiana (em dB), de média zero e variância  $\sigma_\psi^2$ , representando o sombreamento. Esse modelo é amplamente adotado em simulações de redes sem fio por oferecer um bom equilíbrio entre realismo e simplicidade.

### 3 MATERIAIS E MÉTODOS

Para o desenvolvimento deste trabalho, foram necessários recursos como um computador com acesso à internet, configurado em um ambiente Linux, que pode ser uma máquina física ou virtual, além de softwares específicos para a simulação do cenário composto pelo tráfego veicular e pela rede de comunicação.

A simulação do ambiente veicular foi realizada utilizando o SUMO (Simulation of Urban Mobility), um software de código aberto desenvolvido para representar ambientes urbanos complexos. O SUMO permite a modelagem de diferentes tipos de veículos e pedestres, bem como a definição de rotas, velocidades e comportamentos de tráfego, sendo amplamente utilizado em pesquisas relacionadas à mobilidade inteligente.

Para simular a rede de comunicação, foi utilizado o OMNeT++ (Objective Modular Network Testbed in C++), uma plataforma de simulação modular baseada em C++, com suporte a diversos tipos de redes de comunicação. A integração entre o OMNeT++ e o SUMO foi realizada por meio do Artery, um framework que estende o OMNeT++ com suporte ao padrão ETSI ITS-G5, permitindo a simulação de redes V2X em conformidade com os protocolos europeus. O Artery é responsável por sincronizar os eventos do tráfego com a pilha de protocolos veiculares, suportando o envio e recebimento de mensagens definidas pelo modelo C-ITS.

#### 3.1 FERRAMENTAS UTILIZADAS

##### 3.1.1 SIMULAÇÃO VEICULAR (SUMO)

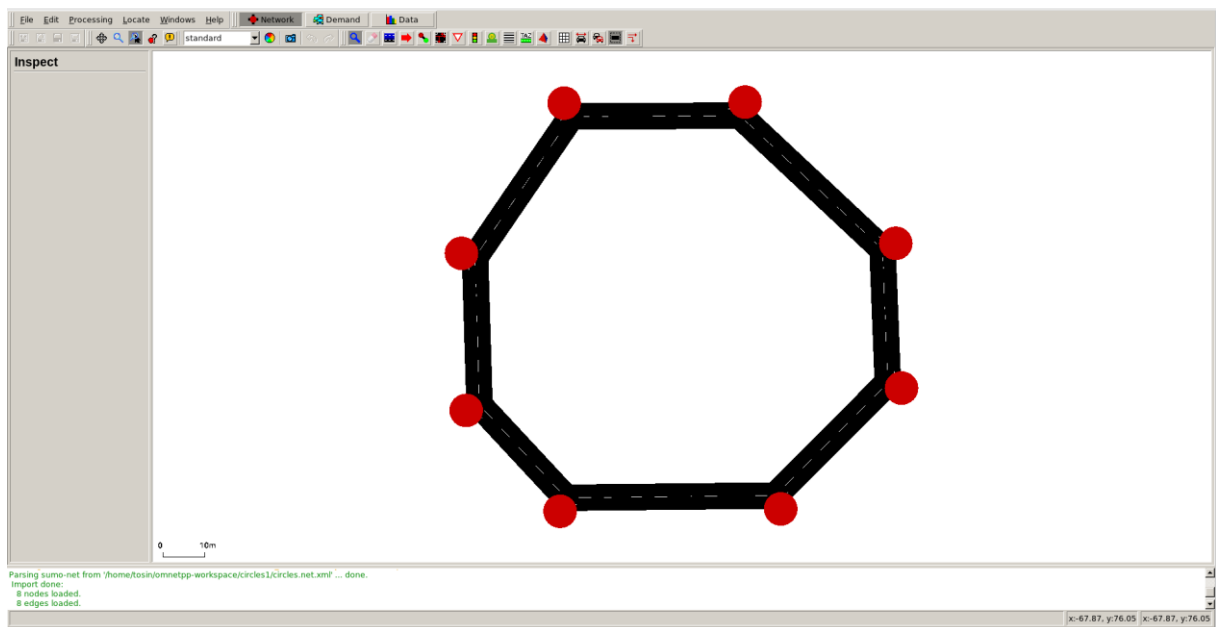
A simulação veicular foi conduzida utilizando o SUMO (Simulation of Urban Mobility), um software de código aberto projetado para representar ambientes urbanos com diferentes níveis de complexidade. No processo de simulação, são definidos o mapa da área urbana, os veículos participantes e os elementos de infraestrutura.

Para que o SUMO execute a simulação, é necessário preparar três arquivos principais: dois no formato .xml e um arquivo de configuração. O formato XML é utilizado para estruturar os dados de forma hierárquica e interpretável pelo simulador.

O arquivo rou.xml descreve as rotas dos veículos, seus tipos, acelerações, horários de partida e destinos. Já o arquivo net.xml define a topologia viária do cenário, incluindo ruas, interseções, faixas de tráfego, prioridades e limites de velocidade. O terceiro arquivo, de configuração, informa ao simulador quais arquivos utilizar e como integrar os elementos definidos, permitindo a execução coordenada da simulação.

Esses arquivos podem ser criados manualmente ou com o auxílio de ferramentas integradas ao SUMO. O Netedit é uma interface gráfica que permite construir mapas personalizados, criar ruas, definir conexões e ajustar o comportamento dos veículos. A ferramenta também gera automaticamente os arquivos XML necessários para a simulação. A FIGURA 4 apresenta a interface do Netedit.

FIGURA 4 : INTERFACE DO NETEDIT



FONTE: O Autor (2024)

Outra ferramenta disponível é o OSMWebWizard, um script que utiliza dados reais da plataforma OpenStreetMap (OSM). Com ele, é possível selecionar diretamente uma área geográfica real e importar sua malha viária. Além disso, o usuário pode definir o número de veículos, tipos de tráfego e opções de exportação. Ao finalizar a configuração, o cenário é gerado automaticamente e carregado no SUMO para execução.

Durante a execução da simulação no SUMO, o cenário previamente configurado pode ser observado, com os veículos representados como carrinhos amarelos se movimentando no mapa, respeitando as infraestruturas e rotas definidas.

FIGURA 5 : SUMO-GUI



FONTE: O autor (2024)

### 3.1.2 SIMULAÇÃO DE REDES (OMNET++)

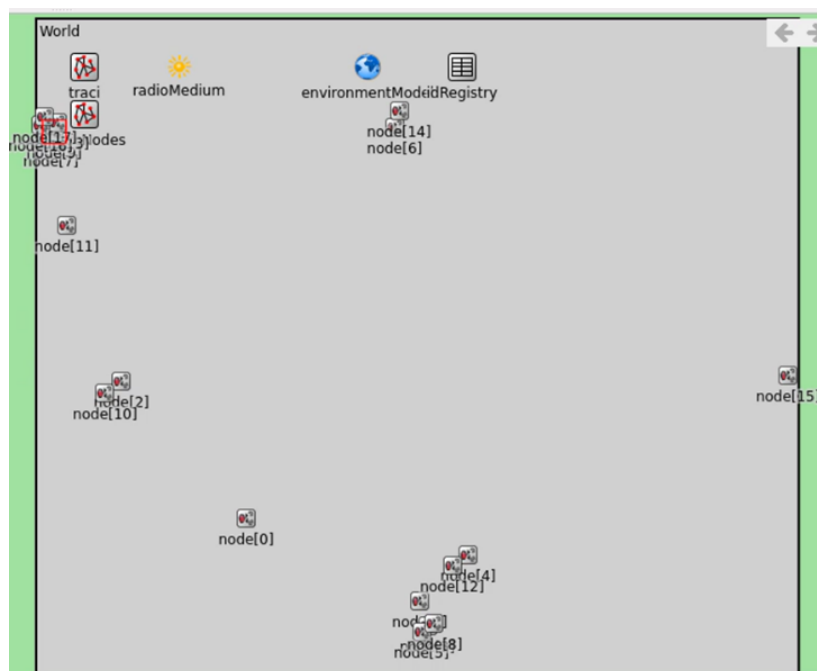
A simulação de redes neste trabalho foi realizada com o OMNeT++, uma ferramenta de simulação de eventos discretos amplamente utilizada em pesquisas acadêmicas, especialmente nas áreas de redes de computadores, comunicações sem fio e redes veiculares. Seu principal diferencial está na arquitetura modular, baseada em orientação a objetos em C++, combinada com a linguagem NED (Network Description), que permite descrever de forma estruturada a topologia da rede simulada.

Com o OMNeT++, é possível modelar sistemas de comunicação complexos, por meio da composição de módulos simples e compostos, representando elementos como rádios, protocolos, canais de transmissão, pilhas de rede e entidades móveis. Esses módulos podem ser configurados graficamente ou por arquivos de texto, oferecendo controle detalhado sobre a simulação. O OMNeT++ também suporta bibliotecas externas, como o INET Framework, e frameworks específicos como o

Artery V2X, utilizado neste trabalho para simular a pilha de protocolos ETSI ITS-G5 e as mensagens C-ITS, incluindo CAM e CPM.

A configuração da simulação é centralizada em um arquivo .ini, que define parâmetros como tempo de simulação, número de veículos, modelo de rádio, frequência de operação, largura de banda e tipo de mensagens. Durante a execução, o OMNeT++ oferece uma interface gráfica que permite acompanhar a simulação em tempo real, visualizando a troca de pacotes, movimentação dos nós e o funcionamento dos módulos internos.

FIGURA 6 : INTERFACE DA SIMULAÇÃO



FONTE: O Autor (2025)

A FIGURA 6 mostra a interface do OMNeT++ durante uma simulação com o Artery, onde é possível observar os nós veiculares (em vermelho).

Os resultados gerados incluem dados escalares e vetoriais, que registram métricas como latência, taxa de transmissão, perdas de pacotes e intensidade de sinal, podendo ser exportados para ferramentas como Python ou MATLAB para análise posterior.

### 3.1.3 FRAMEWORK ARTERY V2X

O Artery é um framework desenvolvido para o OMNeT++, projetado especificamente para simulações de redes veiculares seguindo o padrão europeu ETSI ITS-G5. Assim como o Veins, o Artery é uma ferramenta de código aberto e modular, permitindo a criação de cenários personalizados que combinam mobilidade urbana e comunicação V2X em ambientes dinâmicos. Ele fornece suporte nativo para mensagens C-ITS e para os protocolos GeoNetworking e BTP, conforme especificações do ETSI.

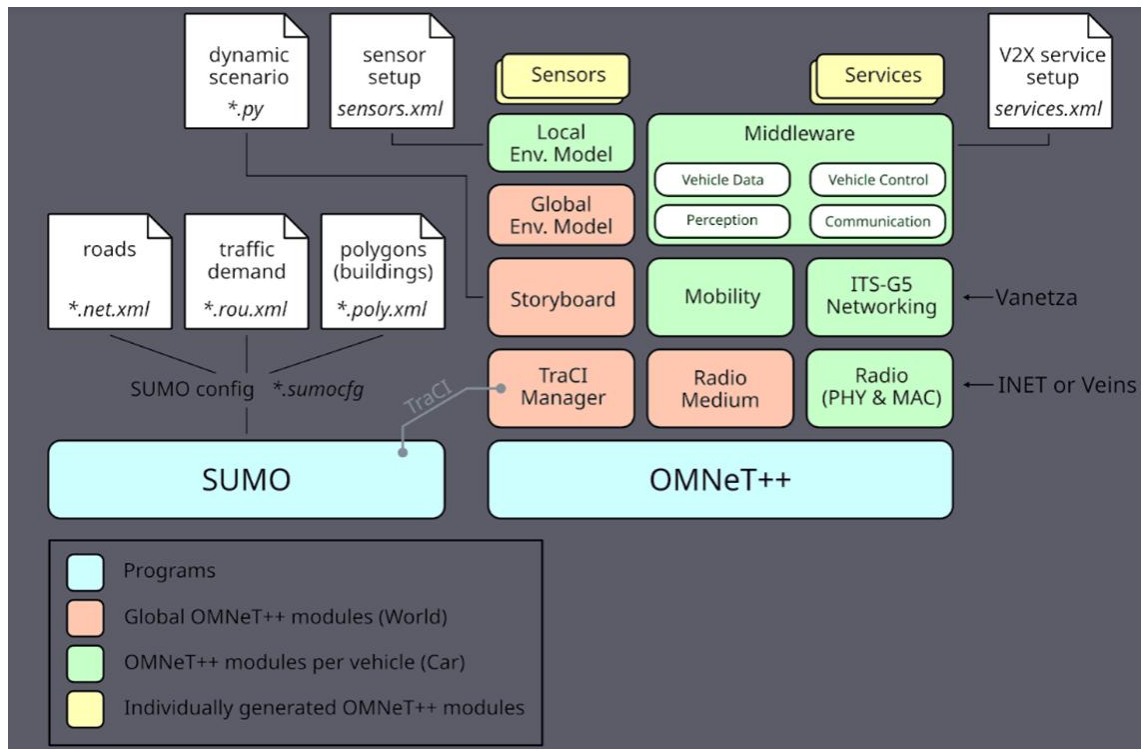
Para iniciar um projeto com Artery, recomenda-se clonar o repositório diretamente do GitHub utilizando o comando `git clone`. Esse procedimento traz todos os arquivos necessários, incluindo os cenários de exemplo, estruturas de simulação, módulos de rede e integração com o SUMO. A construção do projeto requer o uso do CMake, uma ferramenta de automação que gera os arquivos de build necessários para compilar os módulos C++ do OMNeT++ em conjunto com os módulos do Artery. O CMake facilita a configuração do ambiente de compilação, especialmente em sistemas Linux, organizando bibliotecas, caminhos e dependências do projeto de forma padronizada.

Após a configuração e compilação, a simulação é controlada a partir de um arquivo `.ini`, onde são definidos diversos parâmetros do experimento, como o tempo total de simulação, ativação de coleta de métricas, configuração dos sensores, modelo de canal, frequência de operação e localização dos arquivos auxiliares. Nesse arquivo também são referenciados arquivos XML externos como o `services.xml`, responsável por definir os serviços ativos em cada nó (por exemplo, quais mensagens C-ITS são transmitidas), e o `sensors.xml`, que descreve os sensores embarcados aos veículos, como radares de curto e longo alcance.

Durante a execução, a comunicação entre os veículos e a troca de mensagens são coordenadas com a movimentação no SUMO por meio do TraCI (Traffic Control Interface), uma interface que permite a troca de informações entre os dois simuladores em tempo de execução. O TraCI garante que os eventos de mobilidade no SUMO estejam sincronizados com os eventos de rede no OMNeT++, assegurando consistência entre a posição dos veículos e os dados trocados na rede.

A FIGURA 7 ilustra a arquitetura funcional da simulação integrada entre o SUMO e o OMNeT++ com suporte ao framework Artery. À esquerda, estão representados os arquivos de entrada do SUMO, incluindo o mapa viário (.net.xml), a demanda de tráfego (.rou.xml) e elementos visuais como edifícios (.poly.xml). Esses arquivos são utilizados pelo arquivo de configuração. sumocfg, que consolida os dados necessários para a simulação de tráfego. No topo, há os arquivos adicionais como sensors.xml e services.xml, citados anteriormente. O script de cenário dinâmico (.py) também pode ser usado para gerar condições específicas de tráfego.

FIGURA 7 : FLUXOGRAMA ARQUITETURA ARTERY V2X



FONTE: Riebl (2023)

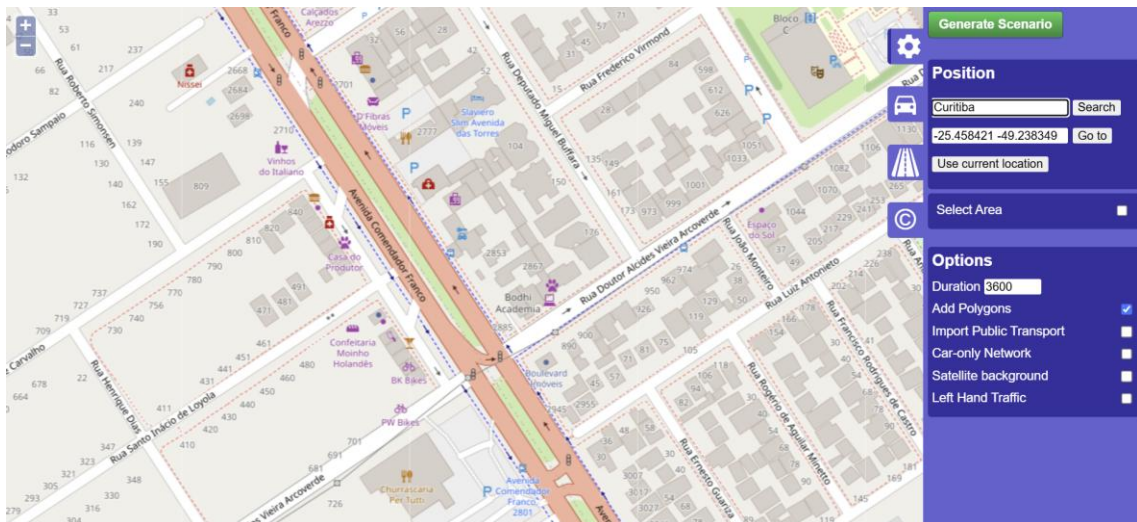
O lado direito da figura mostra os módulos internos do OMNeT++ organizados por tipo: módulos globais, módulos por veículo e módulos gerados dinamicamente. O TraCI Manager atua como intermediário entre os dois simuladores, mantendo sincronizadas a mobilidade veicular (gerada pelo SUMO) e os eventos de rede (gerenciados pelo OMNeT++). A pilha de comunicação é organizada por camadas, incluindo sensores, modelo ambiental, middleware e rádio, permitindo que cada veículo processe dados sensoriais, tome decisões e troque mensagens conforme os

protocolos definidos. A estrutura modular facilita a personalização do comportamento de cada entidade da simulação, garantindo flexibilidade no estudo de aplicações V2X.

### 3.2 CONSTRUÇÃO DO CENÁRIO VEICULAR

O cenário veicular simulado neste trabalho foi baseado em um trecho urbano da cidade de Curitiba (PR), mais especificamente na região da Avenida Comendador Franco, conhecida popularmente como Avenida das Torres, nas proximidades do cruzamento com a Rua Santo Inácio de Loyola. A escolha dessa localização se deu devido ao seu intenso fluxo de veículos ao longo do dia, sendo uma das principais vias de ligação entre os bairros e o centro da cidade. Além disso, essa rota é amplamente utilizada por estudantes e funcionários da Universidade Federal do Paraná (UFPR), o que a torna um ambiente relevante para a análise de aplicações V2X.

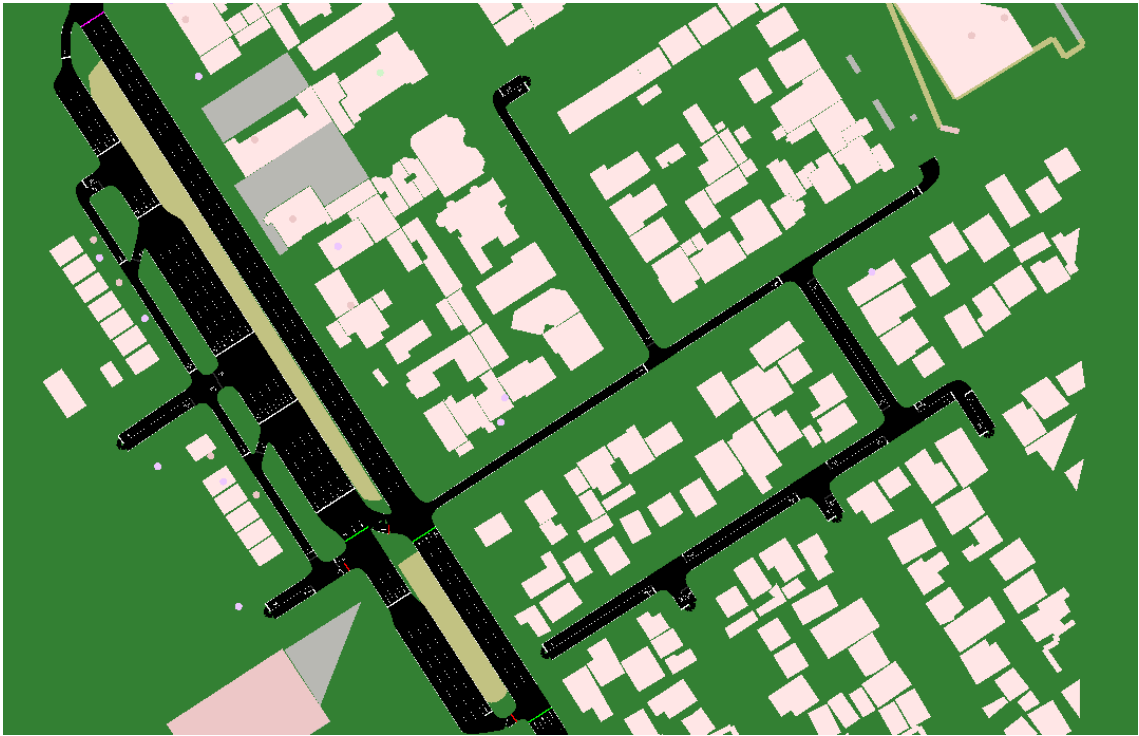
FIGURA 8 : INTERFACE OSM WEB WIZARD



FONTE: O autor (2025)

A FIGURA 8 mostra a interface da ferramenta OSM Web Wizard, utilizada para selecionar a área desejada diretamente a partir dos dados do OpenStreetMap, permitindo a geração automática do mapa viário e de elementos de infraestrutura para uso no SUMO.

FIGURA 9 : MAPA CARREGADO NO SUMO



FONTE: O autor (2025)

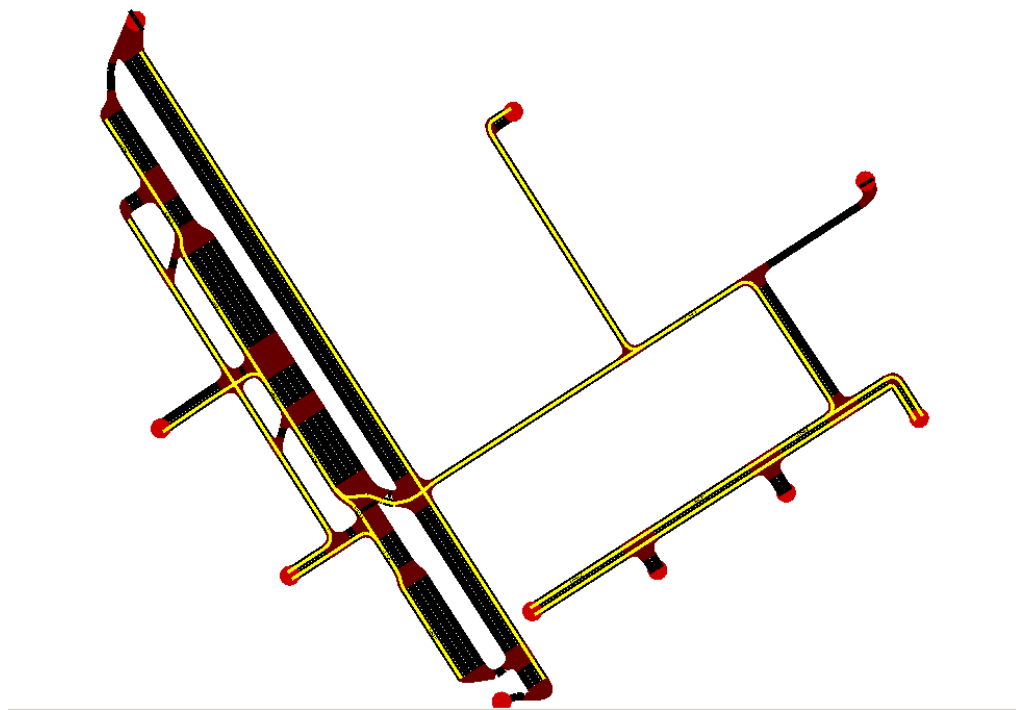
A FIGURA 9 apresenta o resultado da importação do cenário no ambiente do SUMO, após a conversão automática realizada pelo OSM Web Wizard. É possível observar a malha viária gerada a partir dos dados do OpenStreetMap, com ruas, edifícios e áreas verdes já renderizados conforme a geografia da região selecionada. Essa visualização permite verificar se todos os elementos do mapa foram corretamente interpretados e se os cruzamentos e vias estão conectados de maneira funcional para a simulação.

Durante esse processo, no entanto, o OSM Web Wizard pode inserir alguns parâmetros no arquivo .net.xml que não são reconhecidos pela versão do SUMO utilizada em conjunto com o Artery. Esses parâmetros podem gerar erros na etapa de simulação e, portanto, devem ser removidos manualmente.

Após a geração do mapa viário, o próximo passo é a criação das rotas que os veículos deverão seguir durante a simulação. Essa etapa pode ser realizada por meio do NetEdit, interface gráfica do SUMO que permite inserir manualmente os veículos e definir seus parâmetros, como tipo, aceleração, velocidade máxima, também os

pontos de origem e destino de cada rota. O NetEdit oferece uma visualização intuitiva da rede, facilitando a seleção de vias e a atribuição de caminhos aos veículos. Alternativamente, esse processo também pode ser feito diretamente por meio da criação e edição do arquivo rou.xml, que organiza essas informações em formato XML.

FIGURA 10 : ROTAS NO NETEDIT



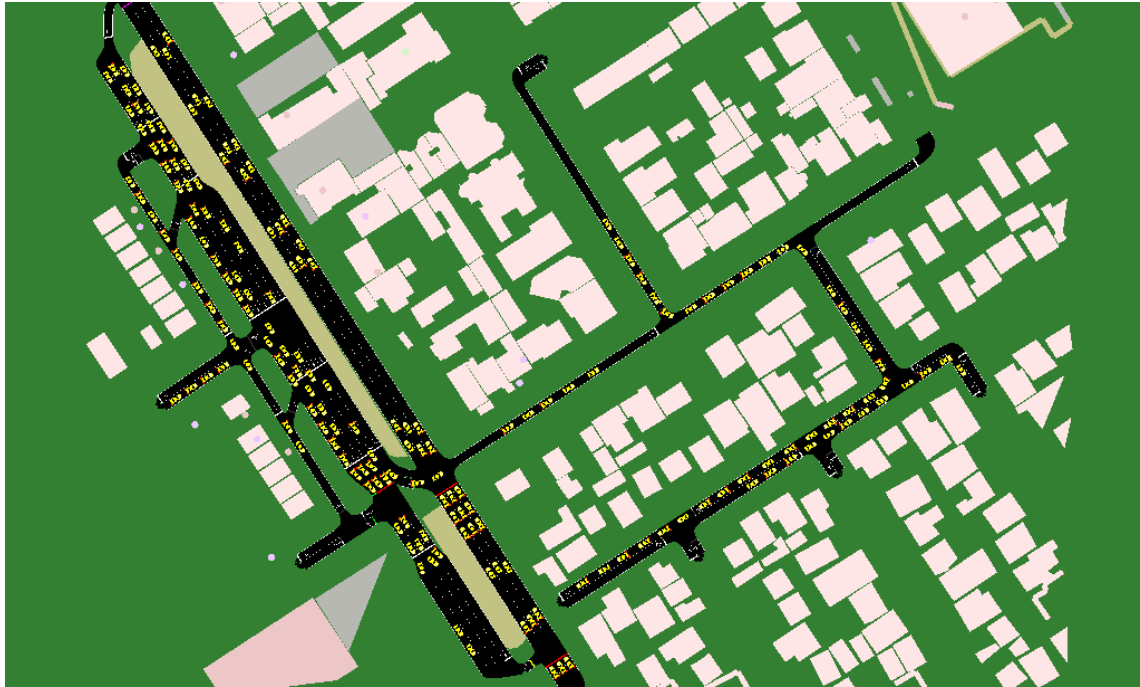
FONTE: O autor (2025)

A FIGURA 10 ilustra a malha viária após a inserção das rotas no NetEdit, destacando os trajetos definidos para os veículos no cenário urbano simulado. Salvando essa etapa é gerado o arquivo, onde se encontram as rotas dos veículos; é possível adicionar os veículos a partir de scripts do próprio SUMO ou utilizando o NetEdit mesmo.

O arquivo de configuração .sumocfg é responsável por reunir todos os componentes necessários para a execução da simulação no SUMO. Nele são especificados os caminhos para os arquivos de rede (.net.xml), de rotas (.rou.xml), além de parâmetros opcionais como o tempo de simulação, inserção de polígonos e visualização gráfica. Esse arquivo funciona como o ponto central da simulação,

garantindo que todos os elementos definidos anteriormente sejam carregados corretamente durante a execução.

FIGURA 11 : SIMULAÇÃO NO SUMO



FONTE: O autor (2025)

A FIGURA 11 apresenta uma visualização da simulação executada no SUMO, já com os veículos inseridos e trafegando pelas rotas previamente definidas. Cada ponto amarelo representa um veículo em movimento, possibilitando o acompanhamento em tempo real do comportamento do tráfego urbano no trecho selecionado. Essa interface gráfica permite verificar o fluxo nas vias, identificar possíveis gargalos ou cruzamentos críticos, além de auxiliar na validação da coerência do cenário gerado antes de sua integração com o ambiente de comunicação no OMNeT++.

### 3.3 CONFIGURAÇÃO DO CENÁRIO DE COMUNICAÇÃO

Para habilitar a troca de mensagens entre os veículos no ambiente simulado, alguns arquivos adicionais precisam ser configurados corretamente no projeto baseado no Artery. Esses arquivos complementam a integração entre a mobilidade gerada pelo SUMO e o comportamento dos nós de rede em OMNeT++.

O primeiro deles é o arquivo CMakeLists.txt, que coordena o processo de compilação dos módulos do OMNeT++ utilizando o CMake. Já o arquivo .ini, específico do OMNeT++, atua como principal ponto de configuração da simulação. Nele são definidos parâmetros como tempo total de execução, controle de saída dos resultados (escalares e vectors), modelo de propagação utilizado (como o LogNormal Shadowing), configuração do middleware, ativação de sensores, entre outras opções que determinam o comportamento da rede e dos veículos.

O arquivo services.xml é responsável por declarar quais mensagens e serviços V2X serão utilizados pelos veículos durante a simulação. É nele que se ativa, por exemplo, a geração de mensagens CAM (Cooperative Awareness Message) ou CPM (Collective Perception Message), associando cada serviço ao middleware de cada nó.

No arquivo sensors.xml, são definidos os sensores presentes nos veículos simulados. Neste exemplo, estão incluídos radares de longo e curto alcance na parte frontal, traseira, também possuindo câmeras. Cada sensor é identificado por um nome e um tipo, que determina seu comportamento dentro do ambiente de simulação, podendo ser observado na figura a seguir.

FIGURA 12 : SENSOR.XML

```
osm2 > sensors.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <sensors>
3      <sensor name="FrontShortRangeRadar" type="artery.envmod.sensor.FrontRadar" />
4      <sensor name="FrontLongRangeRadar" type="artery.envmod.sensor.FrontRadar" />
5      <sensor name="RearLongRangeRadar" type="artery.envmod.sensor.RearRadar" />
6      <sensor name="RearShortRangeRadar" type="artery.envmod.sensor.RearRadar" />
7      <sensor name="SeeThrough" type="artery.envmod.sensor.SeeThroughSensor" />
8      <sensor type="artery.envmod.sensor.CamSensor" />
9  </sensors>
```

FONTE: O autor (2025)

Para a simulação realizada neste trabalho, alguns parâmetros foram definidos no arquivo .ini do OMNeT++. A seguir, são apresentados trechos do .ini utilizados, acompanhados de uma breve explicação sobre sua função no contexto do cenário simulado.

FIGURA 13 : CONFIGURAÇÃO DE COMUNICAÇÃO

```
*.node[*].wlan[*].typename = "VanetNic"  
*.node[*].wlan[*].radio.channelNumber = 180  
*.node[*].wlan[*].radio.carrierFrequency = 5.9 GHz  
*.node[*].wlan[*].transmitter.power = 200 mW
```

FONTE: O autor (2025)

A FIGURA 13 mostra a configuração da interface de comunicação dos nós veiculares. Define-se o pacote de interface "VanetNic", ajustando o canal de comunicação para 180, com frequência de portadora de 5.9 GHz, e uma potência de transmissão de 200 mW. Esses parâmetros são típicos em redes veiculares que utilizam o padrão IEEE 802.11p, operando na faixa de 5.9 GHz destinada às comunicações ITS.

FIGURA 14 : CONFIGURAÇÃO DO PATH LOSS

```
*.radioMedium.analogModelClass = "ScalarAnalogModel"  
*.radioMedium.pathLossType = "LogNormalShadowing"  
*.radioMedium.pathLossAlpha = 3.5  
*.radioMedium.shadowingSigma = 6
```

FONTE: O autor (2025)

A imagem acima refere-se à configuração do modelo de propagação utilizado, o "LogNormalShadowing", representado por uma classe analógica "ScalarAnalogModel". Os valores atribuídos aos parâmetros pathLossAlpha = 3.5 e shadowingSigma = 6 foram definidos de maneira arbitrária, porém com base nas orientações do documento ETSI TR 103 257-1 V1.1.1 (2019-05), o qual fornece parâmetros típicos para ambientes urbanos em simulações de redes veiculares.

FIGURA 15 : CONFIGURAÇÃO DOS SENSORES

```

*.node[*].environmentModel.*.drawSensorCone = true
*.node[*].environmentModel.*.drawLinesOfSight = true
*.node[*].environmentModel.*.drawDetectedObjects = true
*.node[*].environmentModel.*.drawBlockingObstacles = true

*.node[*].environmentModel.FrontLongRangeRadar.fovRange = 250m
*.node[*].environmentModel.FrontLongRangeRadar.fovAngle = 18
*.node[*].environmentModel.FrontLongRangeRadar.numSegments = 6

*.node[*].environmentModel.FrontShortRangeRadar.fovRange = 100m
*.node[*].environmentModel.FrontShortRangeRadar.fovAngle = 120
*.node[*].environmentModel.FrontShortRangeRadar.numSegments = 6

*.node[*].environmentModel.RearLongRangeRadar.fovRange = 250m
*.node[*].environmentModel.RearLongRangeRadar.fovAngle = 18
*.node[*].environmentModel.RearLongRangeRadar.numSegments = 6

*.node[*].environmentModel.RearShortRangeRadar.fovRange = 100m
*.node[*].environmentModel.RearShortRangeRadar.fovAngle = 120
*.node[*].environmentModel.RearShortRangeRadar.numSegments = 6

*.node[*].environmentModel.SeeThrough.fovRange = 50m

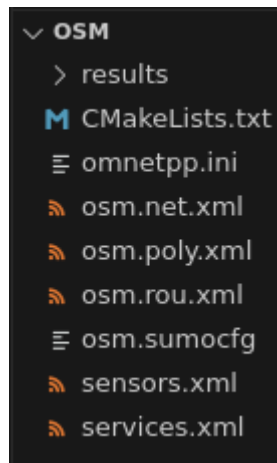
```

FONTE: O autor (2025)

A FIGURA 15 apresenta as configurações relacionadas aos sensores embarcados e ao modelo de ambiente. Ativa-se a visualização de cones de detecção, linhas de visada, objetos detectados e obstáculos bloqueantes. Em seguida, são especificadas as propriedades de alcance (fovRange), ângulo de campo de visão (fovAngle) e o número de segmentos para diferentes radares de curto e longo alcance, tanto na parte frontal quanto traseira dos veículos, além do sensor “SeeThrough”. Esses ajustes são essenciais para garantir a fidelidade da percepção veicular no cenário simulado.

Assim, os arquivos necessários para a criação do cenário completo estão representados na figura a seguir.

FIGURA 16 : ARQUIVOS PARA A SIMULAÇÃO



FONTE: O autor (2025)

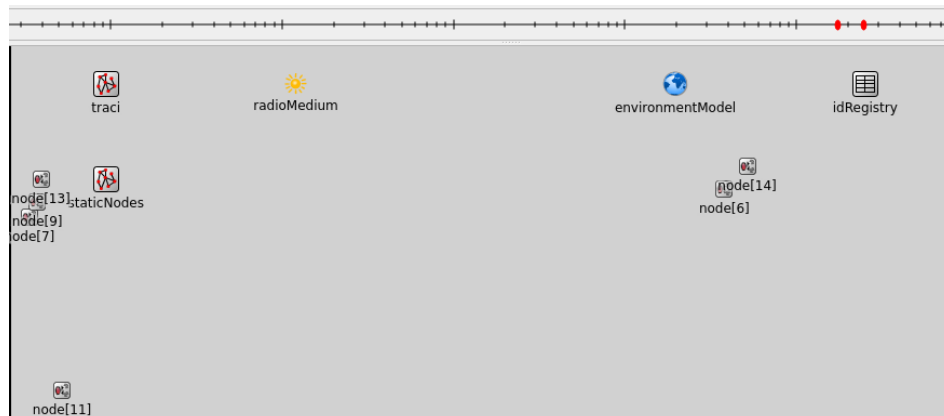
### 3.4 SIMULAÇÃO

A simulação é iniciada por meio do terminal, utilizando o comando:

```
cmake --build build --target run_osm
```

Esse comando compila os arquivos necessários com o CMake e executa a interface gráfica do OMNeT++, como ilustrado em FIGURA 17.

FIGURA 17 : SIMULAÇÃO NO OMNET++



FONTE: O autor (2025)

Durante a execução, é possível observar o processo de transmissão entre os nós da rede. Na FIGURA 18, o nó 13 inicia uma transmissão no evento #98718. Em seguida, verifica-se se o canal está livre e, caso o acesso seja concedido, a transmissão é realizada. Os demais nós alteram seu estado de recepção de “idle” para “receiving”, e o log exibe informações como a potência do sinal recebido, frequência da portadora (carrier frequency), largura de banda (bandwidth) e o identificador da transmissão.

FIGURA 18 : LOG DE TRANSMISSÃO

```

** Event #98718 t=16.764797611814 World.node[13].wlan[0].mac.hcf.edca[1].contention (Contention, id=1102) on selfmsg startTx (omne
INFO: Transmitting frame = (inet::ieee80211::Ieee80211DataFrameWithSNAP)GeoNet packet
DETAIL: Channel access granted to the Best effort queue
INFO: Transmission started: (inet::physicalayer::RadioFrame)GeoNet packet WHOLE as Ieee80211ScalarTransmission, mode = (Ieee80211OF
INFO: Changing radio transmission state from IDLE to TRANSMITTING.
INFO: Changing radio transmitted signal part from NONE to WHOLE.
...
** Event #98723 t=16.764797625758 World.node[9].wlan[0].radio (VanetRadio, id=740) on GeoNet packet (inet::physicalayer::RadioFra
INFO: Reception started: attempting (inet::physicalayer::RadioFrame)GeoNet packet WHOLE as ScalarReception, power = 1.27559e-08 W,
INFO: Changing radio reception state from IDLE to RECEIVING.
INFO: Changing radio received signal part from NONE to WHOLE.

```

FONTE: O autor (2025)

Posteriormente, a mensagem é processada pelo nó de destino, como demonstrado pelo nó 10 na FIGURA 19. O log mostra a quantidade de objetos detectados por cada sensor e, em caso de detecção, são apresentados detalhes como o ID do objeto, posição, velocidade e tempo da detecção.

FIGURA 19 : LOG MENSAGEM CPM

```

(--- Radar Sensor Object List (1 objects) ---
station ID: 176 TraCI ID: object2 lon: -49.2397 deg lat: -25.4586 deg speed: 0.641004 m s-1 when: 16
(--- CAM Sensor Object List (15 objects) ---
station ID: 334 TraCI ID: object4 lon: -49.2382 deg lat: -25.4593 deg speed: 0.893839 m s-1 when: 16
station ID: 650 TraCI ID: object8 lon: -49.2383 deg lat: -25.4595 deg speed: 0.172347 m s-1 when: 16
station ID: 729 TraCI ID: object14 lon: -49.24 deg lat: -25.4576 deg speed: 0.790983 m s-1 when: 16
station ID: 571 TraCI ID: object10 lon: -49.24 deg lat: -25.4577 deg speed: 0.913425 m s-1 when: 16
station ID: 966 TraCI ID: object5 lon: -49.2383 deg lat: -25.4593 deg speed: 0.620459 m s-1 when: 16
station ID: 492 TraCI ID: object7 lon: -49.2385 deg lat: -25.4576 deg speed: 0.80567 m s-1 when: 16
station ID: 413 TraCI ID: object6 lon: -49.2384 deg lat: -25.4595 deg speed: 0.00112435 m s-1 when: 16
station ID: 1045 TraCI ID: object19 lon: -49.24 deg lat: -25.4576 deg speed: 0.794293 m s-1 when: 16
station ID: 1203 TraCI ID: object21 lon: -49.2368 deg lat: -25.4586 deg speed: 0 m s-1 when: 16
station ID: 887 TraCI ID: object17 lon: -49.2399 deg lat: -25.458 deg speed: 0.780668 m s-1 when: 16
station ID: 255 TraCI ID: object3 lon: -49.2384 deg lat: -25.4595 deg speed: 0 m s-1 when: 16
station ID: 97 TraCI ID: object1 lon: -49.2384 deg lat: -25.4594 deg speed: 0.607369 m s-1 when: 16
station ID: 176 TraCI ID: object2 lon: -49.2397 deg lat: -25.4586 deg speed: 0.641004 m s-1 when: 16
station ID: 18 TraCI ID: object0 lon: -49.2391 deg lat: -25.4591 deg speed: 0.933248 m s-1 when: 16
station ID: 1124 TraCI ID: object12 lon: -49.2385 deg lat: -25.4575 deg speed: 0 m s-1 when: 16
(--- By name ---
(--- FrontShortRangeRadar Object List (1 objects) ---
station ID: 176 TraCI ID: object2 lon: -49.2397 deg lat: -25.4586 deg speed: 0.641004 m s-1 when: 16
(--- FrontLongRangeRadar Object List (0 objects) ---
(--- RearLongRangeRadar Object List (0 objects) ---
(--- RearShortRangeRadar Object List (0 objects) ---
(--- SeeThrough Object List (1 objects) ---
station ID: 176 TraCI ID: object2 lon: -49.2397 deg lat: -25.4586 deg speed: 0.641004 m s-1 when: 16

```

FONTE: O autor (2025)

### 3.5 MÉTRICAS

Após a simulação, os resultados são armazenados em arquivos .sca (dados escalares) e .vec (dados vetoriais). Utilizando a interface do OMNeT++, esses arquivos podem ser exportados para o formato .csv, permitindo a filtragem das métricas de interesse.

A análise das métricas foi realizada com ferramentas externas como Python, utilizando as bibliotecas pandas e SciPy, possibilitando o tratamento estatístico e visualização dos dados obtidos.

## 4 RESULTADOS

Para a análise das métricas de latência, jitter, throughput e PER, foram realizados quatro cenários distintos de simulação. Todos os cenários mantêm o número total de veículos, variando apenas a porcentagem de veículos inteligentes (ITS), o que altera a quantidade de nós simultâneos atuando na rede no período analisado.

A fim de limitar o volume de dados processados, foi definido um intervalo de amostragem entre os instantes de 200 s a 220 s. Esse intervalo foi escolhido por representar um momento de alta densidade veicular na simulação, com aproximadamente 63 veículos por quilômetro.

Os cenários foram organizados da seguinte forma:

- Cenário 1: 18 nós simultâneos
- Cenário 2: 61 nós simultâneos
- Cenário 3: 116 nós simultâneos
- Cenário 4: 160 nós simultâneos

As métricas avaliadas foram: latência, jitter, throughput e PER (Packet Error Rate), com os dados processados em Python por meio das bibliotecas pandas e SciPy.

### 4.1 CENÁRIO 1 – 18 NÓS SIMULTÂNEOS

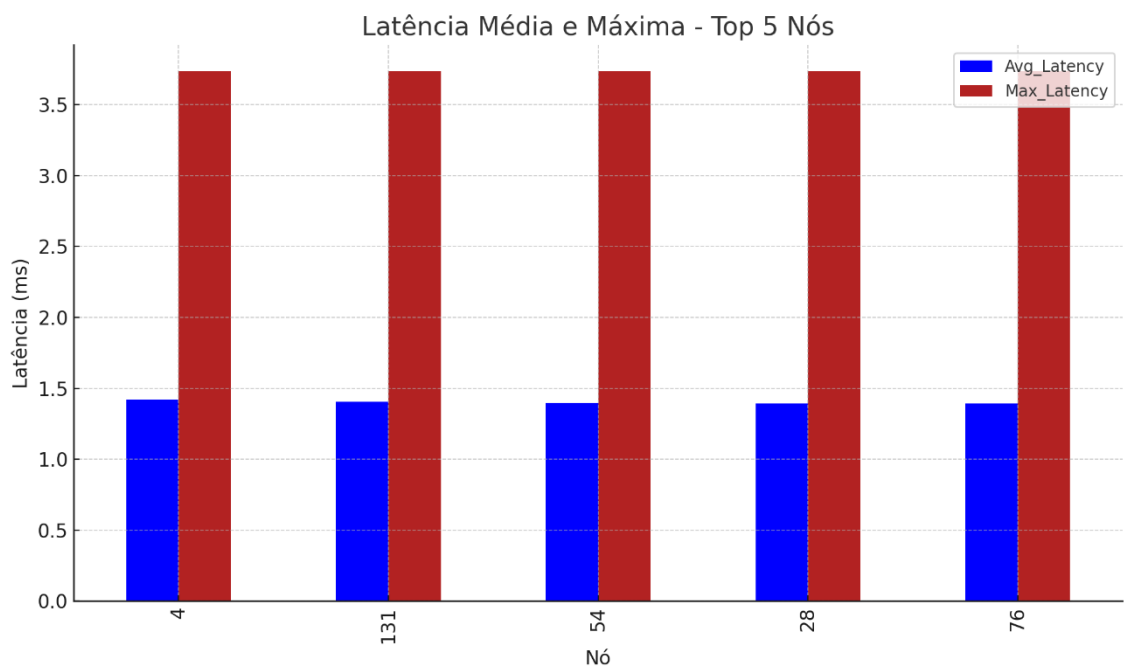
TABELA 1 : RESULTADOS CENÁRIO 1

|                   | Max.  | Med.  | Min.  |
|-------------------|-------|-------|-------|
| Latência (ms)     | 3,736 | 1,384 | 0,208 |
| Jitter Médio (ms) | 2,095 | 0,663 | 0,075 |
| Throughput (Mbps) | 0,748 | 0,624 | 0,511 |
| PER Médio (%)     | 0,337 | 0,098 | 0     |

FONTE: O autor (2025)

Nesse cenário, com baixa densidade de nós, observam-se variações discretas nos valores médios das métricas. A FIGURA 20 apresenta a latência média e máxima, com destaque para cinco nós que registraram uma latência máxima de 3,74 ms e média de 1,42 ms, acima da média geral.

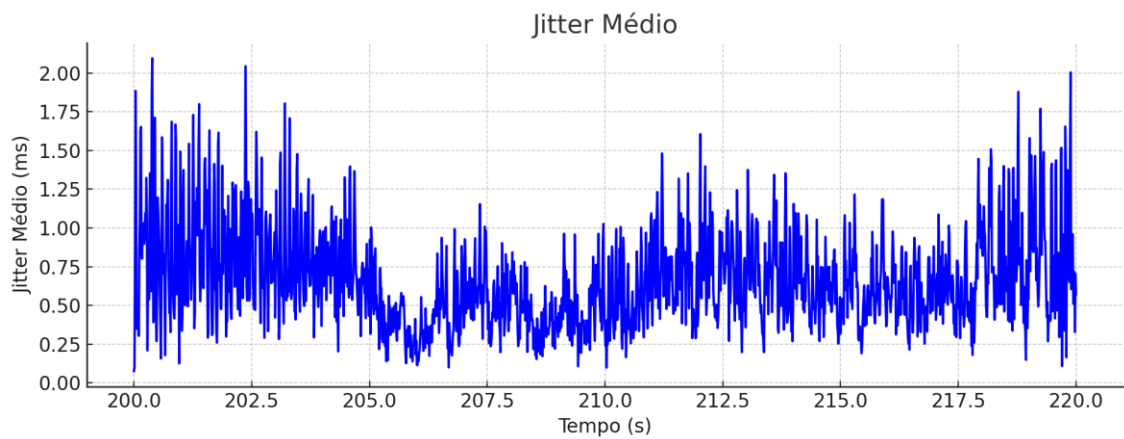
FIGURA 20 : LATÊNCIA CENÁRIO 1



FONTE: O autor (2025)

A FIGURA 21 mostra o comportamento do jitter médio ao longo do tempo, que reflete as variações na latência observadas entre os pacotes.

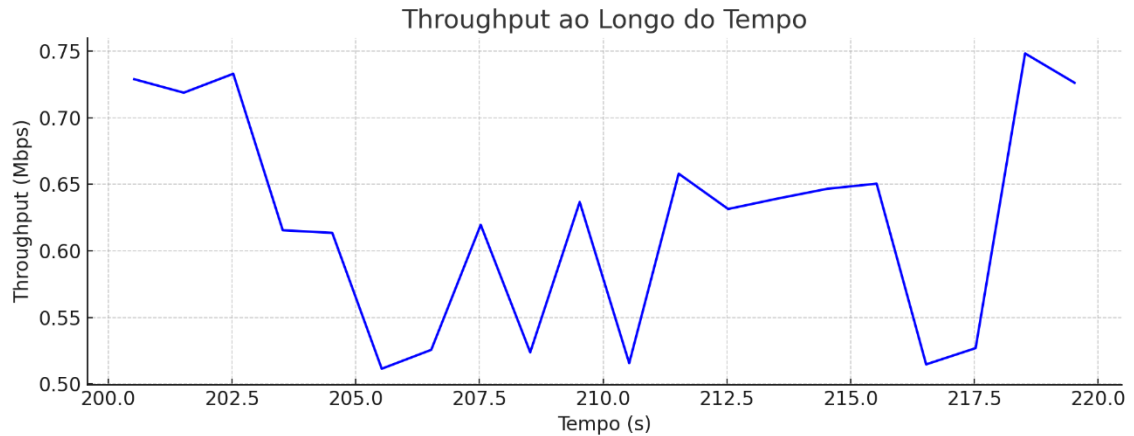
FIGURA 21 : JITTER CENÁRIO 1



FONTE: O autor (2025)

O throughput apresenta pequenos vales em alguns instantes, como visto na FIGURA 22, o que indica baixa troca de mensagens em determinados momentos.

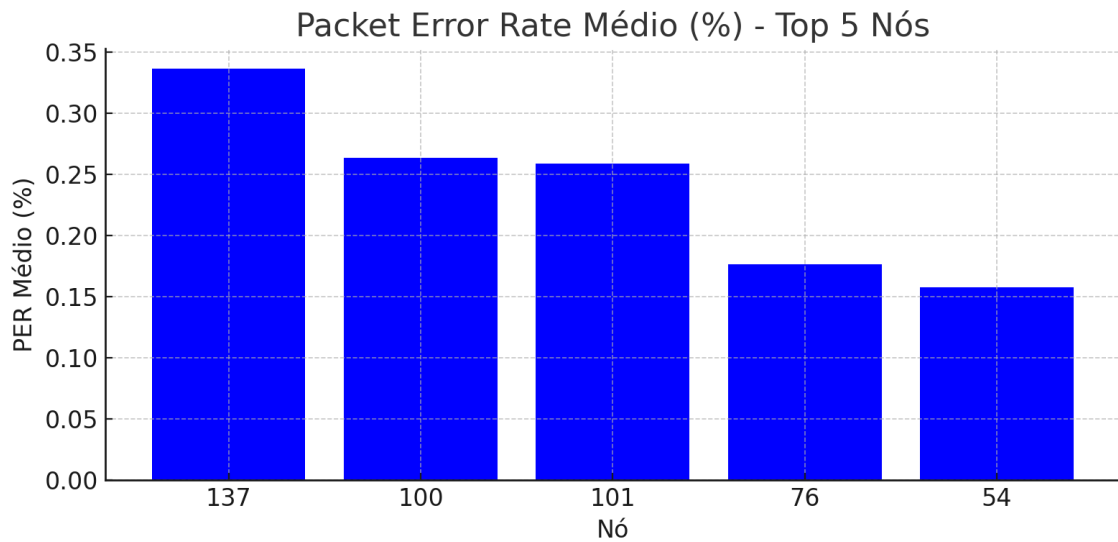
FIGURA 22 : TROUGHPUT CENÁRIO 1



FONTE: O Autor (2025)

A FIGURA 23 exibe os cinco nós com maior PER médio. Mesmo que todos os valores estejam abaixo de 0,5%, esses nós apresentaram valores acima da média geral do cenário.

FIGURA 23 : PER CENÁRIO 1



FONTE: O autor (2025)

#### 4.2 CENÁRIO 2 – 61 NÓS SIMULTÂNEOS

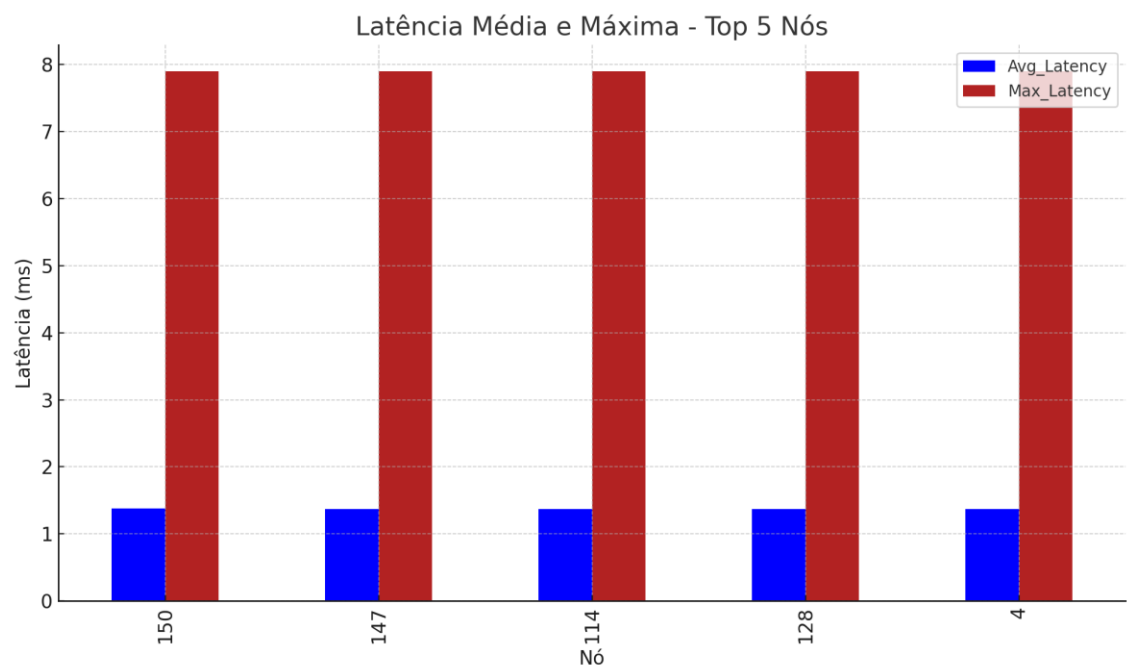
TABELA 2 : RESULTADOS CENÁRIO 2

|                   | Max.  | Med.  | Min.   |
|-------------------|-------|-------|--------|
| Latência (ms)     | 7,897 | 1,362 | 0,208  |
| Jitter Médio (ms) | 2,226 | 0,717 | 0,0002 |
| Throughput (Mbps) | 1,560 | 1,299 | 1,091  |
| PER Médio (%)     | 1,442 | 0,527 | 0      |

FONTE: O autor (2025)

A latência máxima nesse cenário foi consideravelmente maior que a média, sugerindo ocorrência pontual de congestionamento. A FIGURA 24 mostra que os nós com maior latência apresentaram valores próximos entre si.

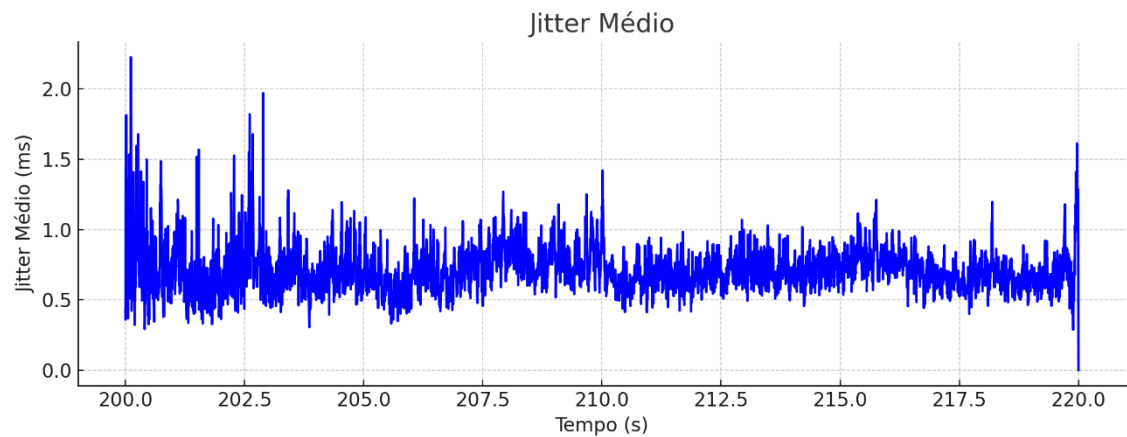
FIGURA 24 : LATÊNCIA CENÁRIO 2



FONTE: O autor (2025)

O jitter, conforme apresentado na FIGURA 25, atingiu seu valor máximo logo no início do intervalo analisado, refletindo instabilidade pontual.

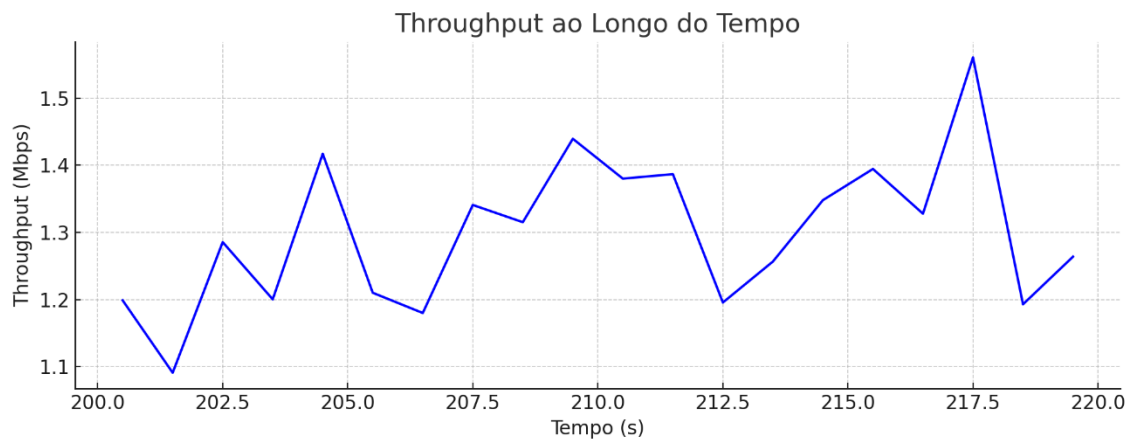
FIGURA 25 : JITTER CENÁRIO 2



FONTE: O autor (2025)

Conforme mostra a FIGURA 26, o valor mínimo do throughput coincide com o pico de jitter, indicando possível congestionamento do canal naquele instante.

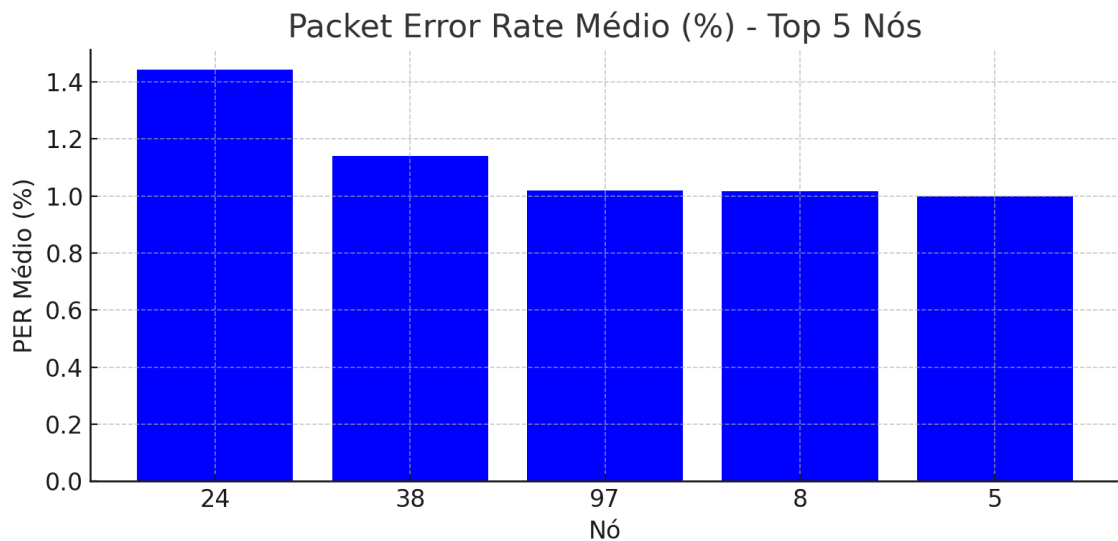
FIGURA 26 : TROUGHPUT CENÁRIO 2



FONTE: O autor (2025)

A FIGURA 27 revela que o PER médio aumentou em comparação ao cenário anterior, com nós superando 1% de perda.

FIGURA 27 : PER CENÁRIO 2



FONTE: O autor (2025)

#### 4.3 CENÁRIO 3 – 116 NÓS SIMULTÂNEOS

TABELA 3 : RESULTADOS CENÁRIO 3

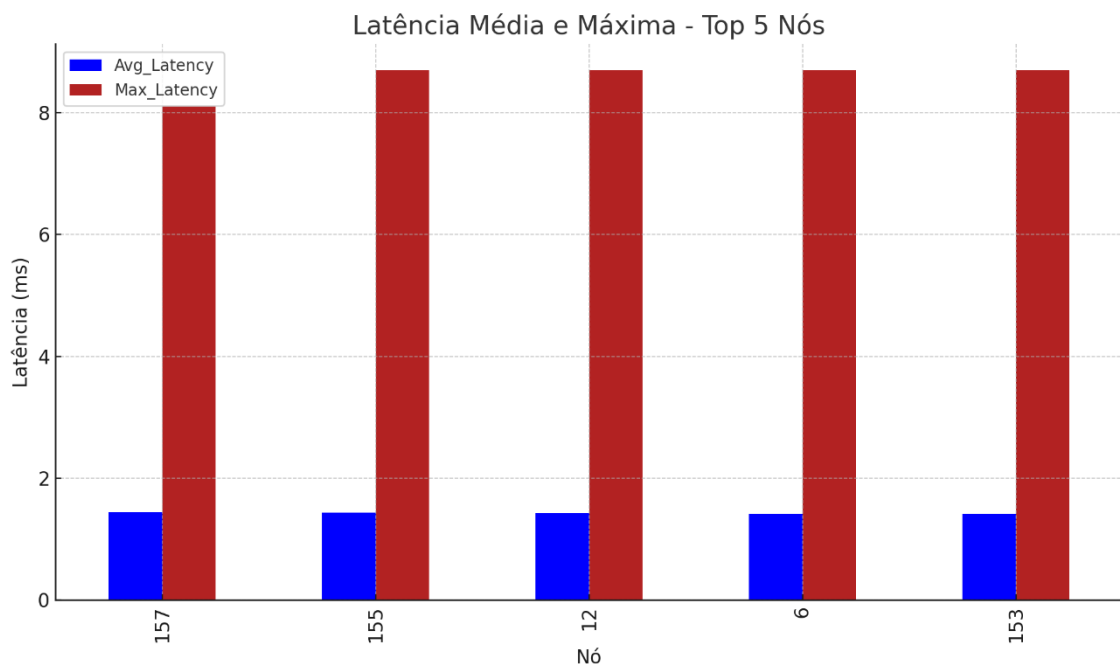
|                   | Max.  | Med.  | Min.    |
|-------------------|-------|-------|---------|
| Latência (ms)     | 8,696 | 1,405 | 0,208   |
| Jitter Médio (ms) | 2,153 | 0,827 | 0,023   |
| Throughput (Mbps) | 1,886 | 1,573 | 1,416   |
| PER Médio (%)     | 2,009 | 0,768 | 0,28351 |

FONTE: O autor (2025)

Com 116 nós, observou-se um leve aumento na latência média e no jitter, ainda que dentro de uma faixa próxima à do cenário anterior. A variação mais expressiva foi no PER médio, que passou a superar 0,75%.

A FIGURA 28 apresenta os nós com maior latência. Nota-se que, mesmo com o crescimento na quantidade de nós, os valores ainda se mantêm em patamares controlados, sem picos fora do esperado, quando comparados com o cenário anterior.

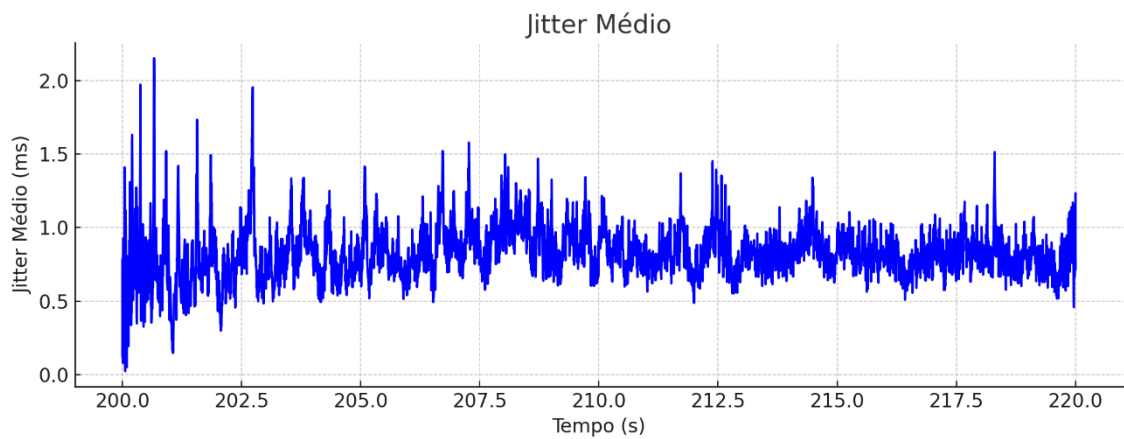
FIGURA 28 : LATÊNCIA CENÁRIO 3



FONTE: O autor (2025)

A FIGURA 29 mostra a evolução do jitter médio ao longo do tempo. A oscilação permanece estável, com pequenas variações em certos momentos da simulação.

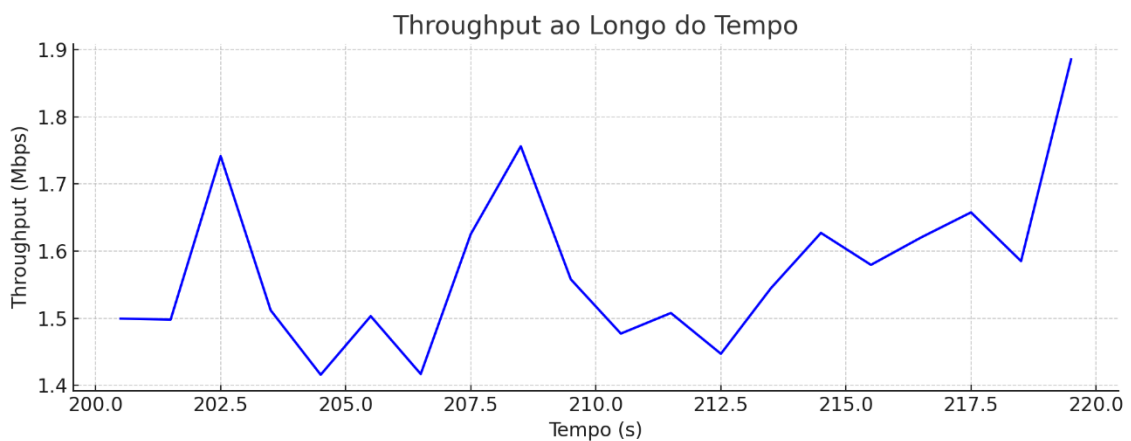
FIGURA 29 : JITTER CENÁRIO 3



FONTE: O autor (2025)

A FIGURA 30 revela que, apesar do aumento do tráfego, o throughput manteve um padrão contínuo, sem quedas abruptas. Isso demonstra que a rede ainda comporta bem o volume de mensagens trocadas.

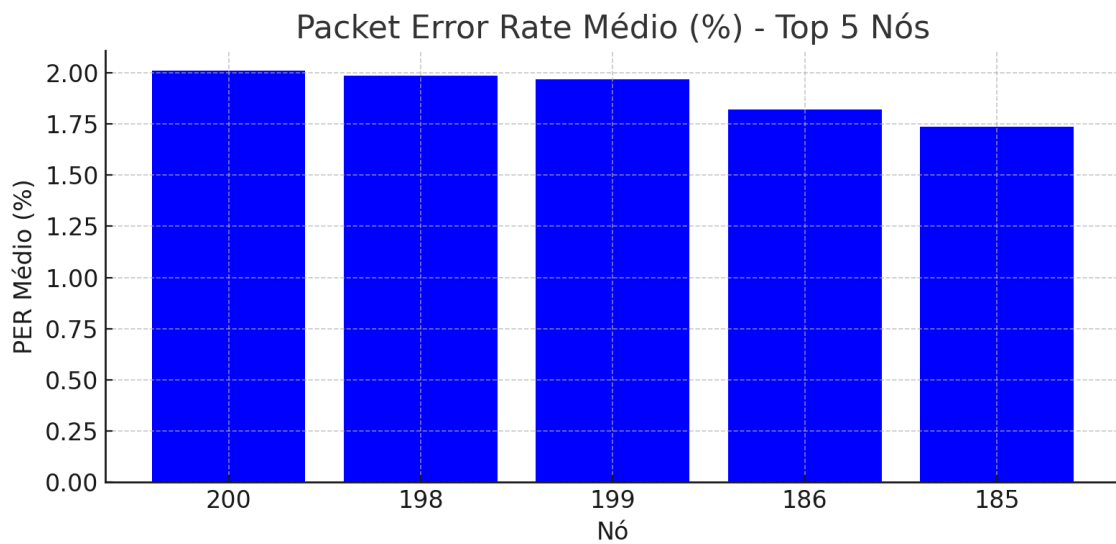
FIGURA 30 : TROUGHPUT CENÁRIO 3



FONTE: O autor (2025)

Por fim, a FIGURA 31 exibe os cinco nós com maior PER médio. Os valores ultrapassam 2% em alguns casos, refletindo o início de uma sobrecarga mais perceptível na comunicação.

FIGURA 31 : PER CENÁRIO 3



FONTE: O autor (2025)

#### 4.4 CENÁRIO 4 – 161 NÓS SIMULTÂNEOS

TABELA 4 : RESULTADOS CENÁRIO 4

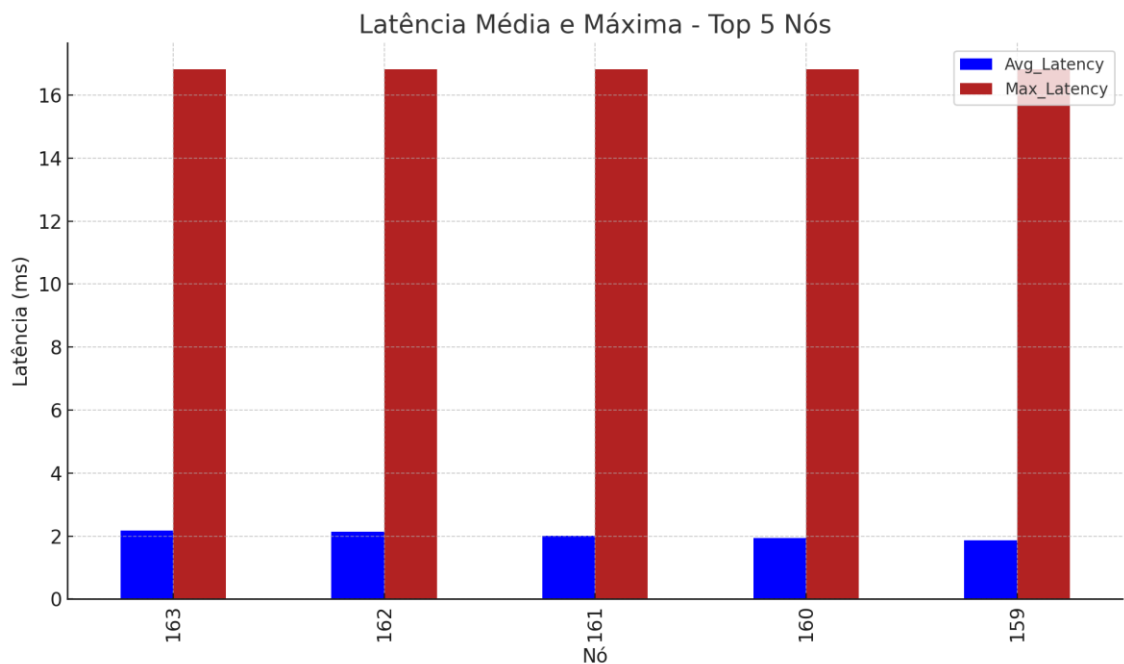
|                   | Max.   | Med.  | Min.    |
|-------------------|--------|-------|---------|
| Latência (ms)     | 16,824 | 1,563 | 0,208   |
| Jitter Médio (ms) | 2,453  | 0,928 | 0,071   |
| Throughput (Mbps) | 2,144  | 1,870 | 1,505   |
| PER Médio (%)     | 2,379  | 1,360 | 0,75069 |

FONTE: O autor (2025)

Neste cenário, com a maior quantidade de nós, a latência máxima dobrou em relação ao cenário anterior, evidenciando pontos de congestionamento no canal. O jitter médio também aumentou, e o PER passou a atingir valores acima de 1% com frequência.

A FIGURA 32 mostra os cinco nós com maior latência. O destaque vai para a forte assimetria entre os valores médios e máximos, indicando momentos isolados de atraso elevado, provavelmente associados à contenção no acesso ao canal.

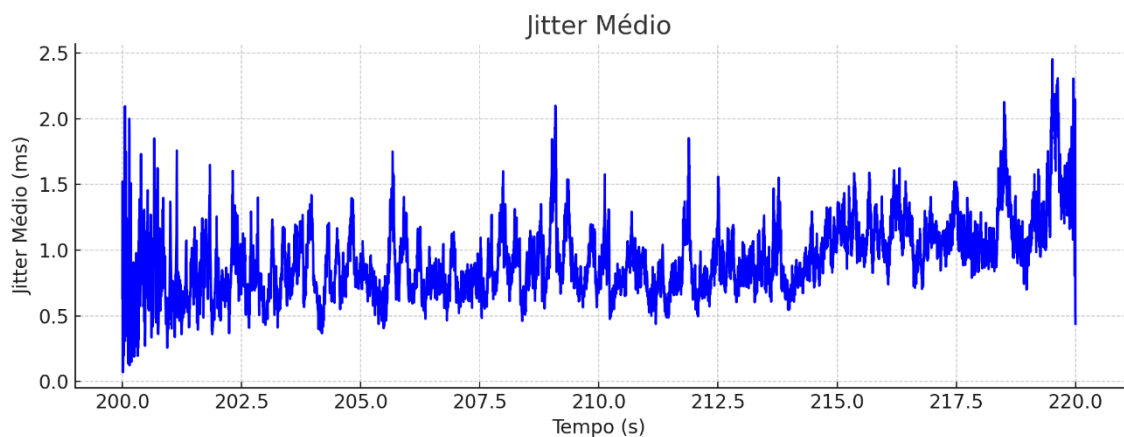
FIGURA 32 : LATÊNCIA CENÁRIO 4



FONTE: O autor (2025)

A FIGURA 33 apresenta o jitter ao longo do tempo. As flutuações aumentam, principalmente em trechos com maior troca de mensagens, sugerindo instabilidade crescente conforme mais nós competem pelo meio.

FIGURA 33 : JITTER CENÁRIO 4

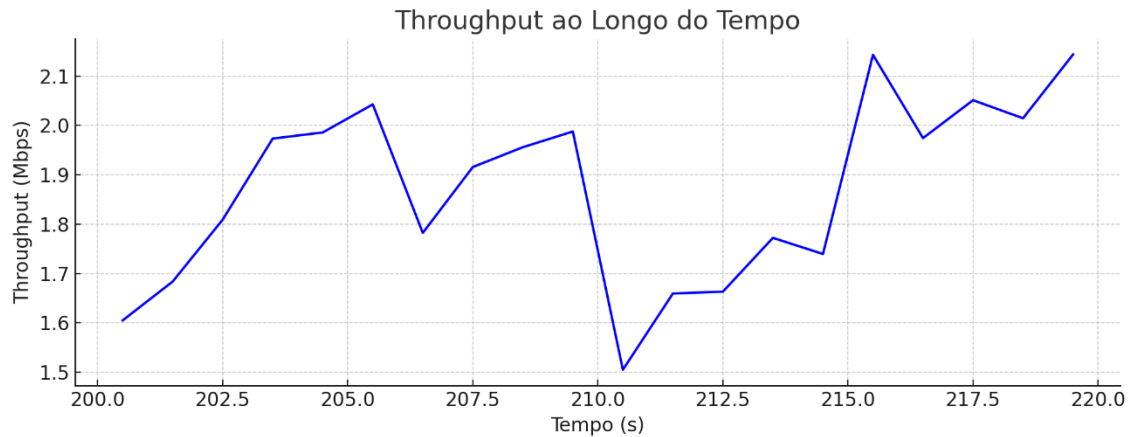


FONTE: O autor (2025)

A FIGURA 34 destaca um vale no throughput durante o período de pico de jitter, sugerindo que múltiplos nós tentavam transmitir simultaneamente, mas poucos

conseguiram acesso ao canal. Esse comportamento é típico de redes sobrecarregadas.

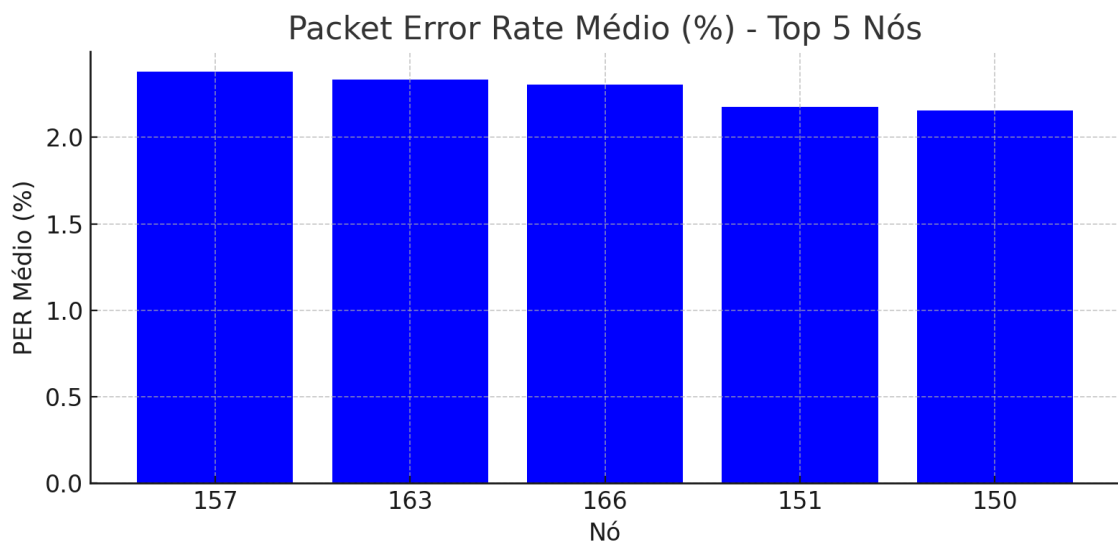
FIGURA 34 : TROUGHPUT CENÁRIO 4



FONTE: O autor (2025)

Na FIGURA 35, observa-se que os nós com maior PER médio ultrapassam 2%, com tendência de crescimento conforme a densidade da rede aumenta. Isso confirma que o aumento da carga afeta diretamente a confiabilidade da entrega de pacotes.

FIGURA 35 : PER CENÁRIO 4



FONTE: O autor (2025)

## 4.5 COMPARAÇÃO ENTRE CENÁRIOS

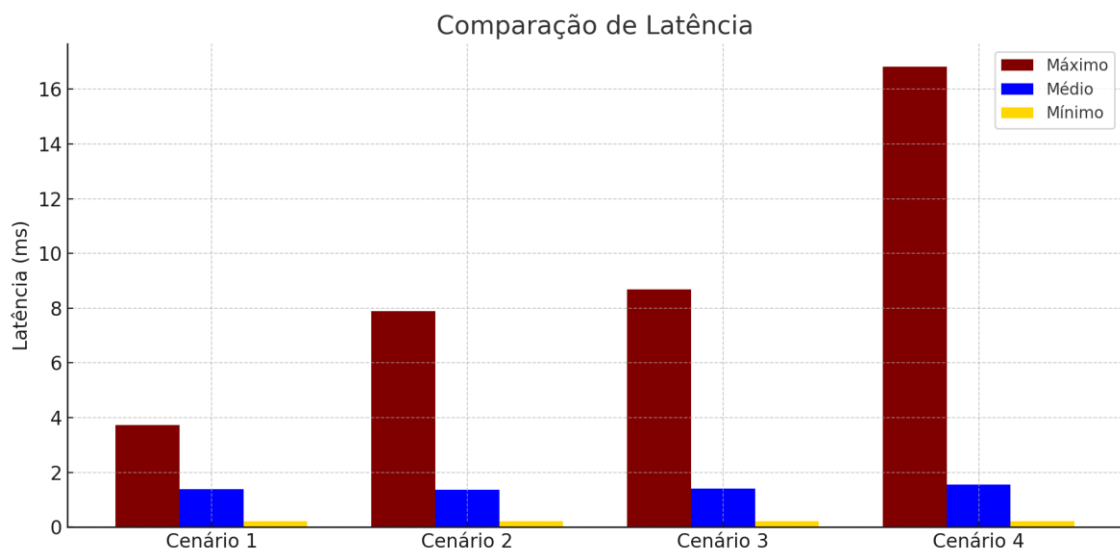
### 4.5.1 LATÊNCIA

TABELA 5 : COMPARAÇÃO LATÊNCIA

|           | Latência (ms) |       |       |
|-----------|---------------|-------|-------|
|           | Max.          | Med.  | Min.  |
| Cenário 1 | 3,736         | 1,384 | 0,208 |
| Cenário 2 | 7,897         | 1,362 | 0,208 |
| Cenário 3 | 8,696         | 1,405 | 0,208 |
| Cenário 4 | 16,824        | 1,563 | 0,208 |

FONTE: O autor (2025)

FIGURA 36 : COMPARAÇÃO DE LATÊNCIA



FONTE: O autor (2025)

A latência média apresentou crescimento quase linear com o aumento dos nós. O valor mínimo permaneceu constante, enquanto o valor máximo variou significativamente, como esperado devido ao aumento do número de transmissões e interferências. Ainda assim, todos os valores ficaram bem abaixo do limite de 300 ms considerado aceitável para redes ITS-G5, segundo a ETSI.

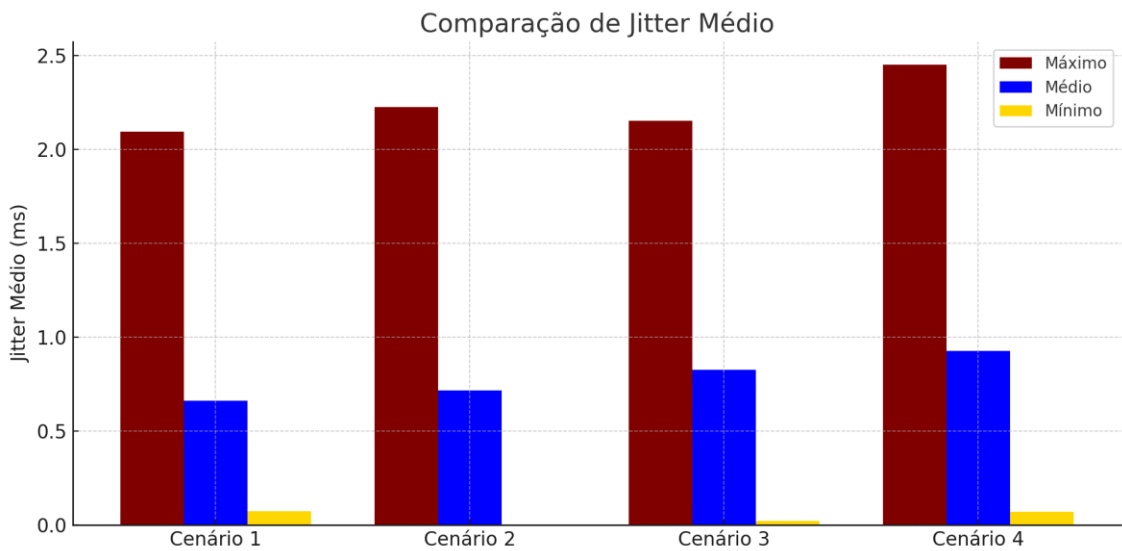
### 4.5.2 JITTER

TABELA 6 : COMPARAÇÃO JITTER

| Jitter Médio (ms) |       |       |        |
|-------------------|-------|-------|--------|
|                   | Max.  | Med.  | Min.   |
| Cenário 1         | 2,095 | 0,663 | 0,075  |
| Cenário 2         | 2,226 | 0,717 | 0,0002 |
| Cenário 3         | 2,153 | 0,827 | 0,023  |
| Cenário 4         | 2,453 | 0,928 | 0,071  |

FONTE: O autor (2025)

FIGURA 37 : COMPARAÇÃO DE JITTER



FONTE: O autor (2025)

Mesmo em cenários com alta densidade, o jitter manteve estabilidade relativa, indicando que a variação de atraso entre mensagens permaneceu sob controle.

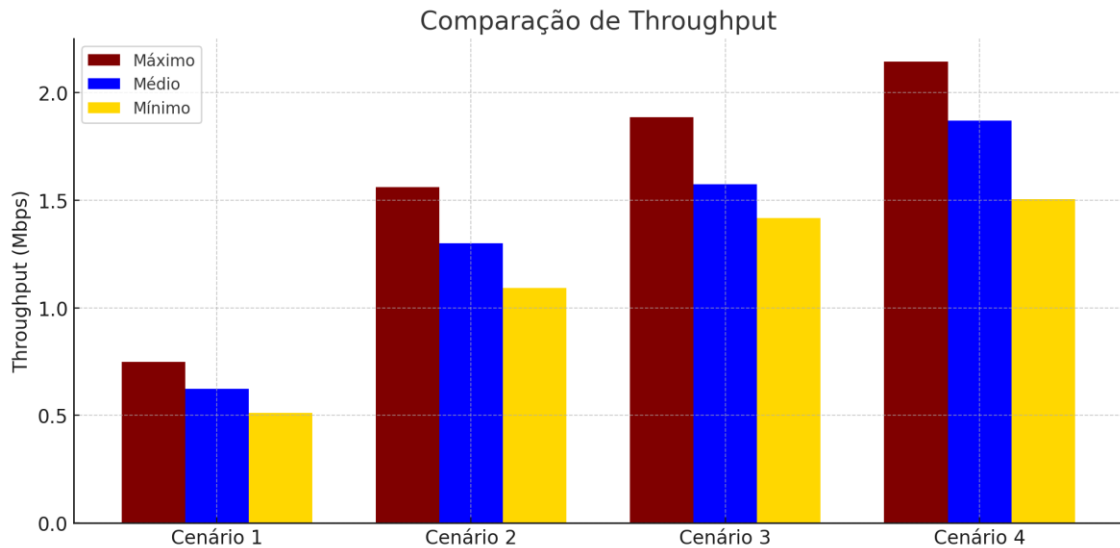
4.5.3 THROUGHPUT

TABELA 7 : COMPARAÇÃO TROUGHPUT

| Throughput (Mbps) |       |       |       |
|-------------------|-------|-------|-------|
|                   | Max.  | Med.  | Min.  |
| Cenário 1         | 0,748 | 0,624 | 0,511 |
| Cenário 2         | 1,560 | 1,299 | 1,091 |
| Cenário 3         | 1,886 | 1,573 | 1,416 |
| Cenário 4         | 2,144 | 1,870 | 1,505 |

FONTE: O autor (2025)

FIGURA 38 : COMPARAÇÃO DE TROUGHPUT



FONTE: O autor (2025)

O throughput aumentou proporcionalmente à quantidade de nós, conforme esperado. Nenhum dos cenários ultrapassou os limites máximos teóricos (BPSK a 3 Mbps e QPSK a 6 Mbps, conforme cabeçalhos ITS-G5 pré-estabelecidos), mas o cenário 4 se aproxima do limite da modulação BPSK.

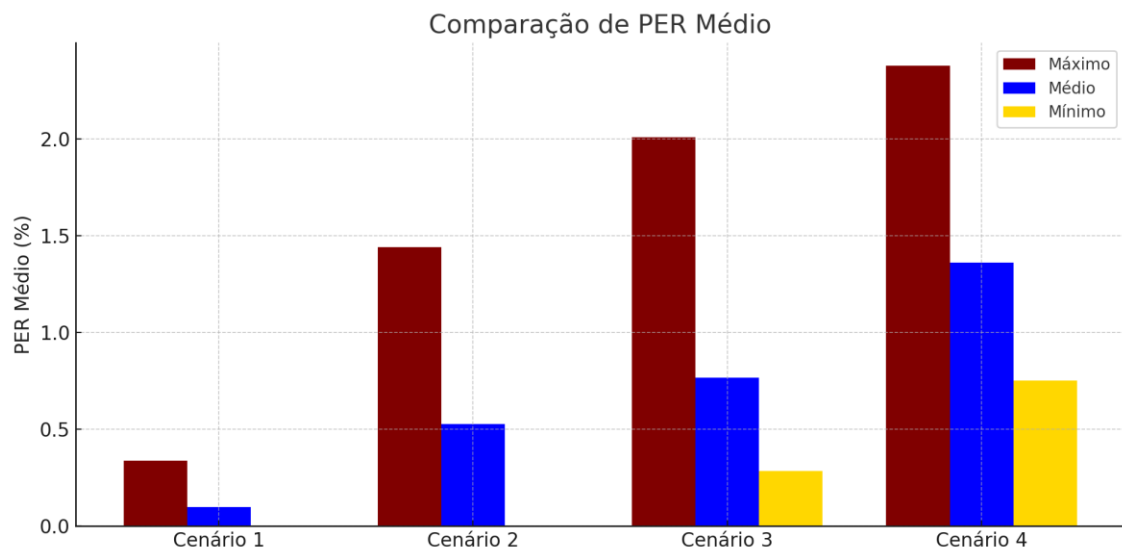
#### 4.5.4 PER

TABELA 8 : COMPARAÇÃO PER

|           | PER Médio (%) |       |         |
|-----------|---------------|-------|---------|
|           | Max.          | Med.  | Min.    |
| Cenário 1 | 0,337         | 0,098 | 0       |
| Cenário 2 | 1,442         | 0,527 | 0       |
| Cenário 3 | 2,009         | 0,768 | 0,28351 |
| Cenário 4 | 2,379         | 1,360 | 0,751   |

FONTE: O autor (2025)

FIGURA 39 : COMPARAÇÃO DE PER



FONTE: O autor (2025)

O PER apresentou comportamento crescente com o aumento da quantidade de nós. A partir do cenário 3, os valores mínimos também passaram a ser relevantes, refletindo aumento da perda de pacotes devido à contenção do canal. Ainda assim, todos os valores ficaram bem abaixo do limite de 10% sugerido pela ETSI para redes ITS-G5 sem interferência.

## 5 CONCLUSÃO

Este trabalho apresentou uma análise do impacto da transmissão de mensagens CPM no desempenho de redes V2X, utilizando simulações baseadas no padrão ETSI ITS-G5. Foram avaliados quatro cenários com diferentes densidades de nós simultâneos, permitindo a observação dos efeitos do aumento do tráfego sobre métricas críticas como latência, jitter, throughput e PER.

Os resultados mostraram que o aumento na quantidade de nós leva a um crescimento linear nas métricas de latência e PER, com a latência máxima chegando a ultrapassar 16 ms e o PER médio atingindo 1,36% no cenário mais carregado. Apesar disso, os valores permaneceram dentro dos limites aceitáveis definidos pelas normas, indicando que a rede consegue suportar o aumento de carga até certo ponto. Conclui-se que a utilização da CPM é viável em ambientes urbanos com densidade moderada a alta, desde que haja controle adequado do canal de comunicação.

Além de permitir o alcance dos objetivos propostos, este trabalho também proporcionou a aplicação prática de conhecimentos adquiridos ao longo da graduação, especialmente nas áreas de comunicação digital, redes de computadores e princípios de telecomunicações. O estudo exigiu a compreensão do funcionamento de sistemas de comunicação em camadas, com foco em redes veiculares, e a realização de simulações em um ambiente modular orientado a objetos. Esse processo contribuiu para o aprofundamento técnico e consolidou a capacidade analítica no contexto de sistemas cooperativos.

Durante o desenvolvimento, surgiram dificuldades relacionadas à aprendizagem de ferramentas específicas como OMNeT++, SUMO e o framework Artery, que ainda está em constante evolução. A falta de experiência prévia em programação orientada a objetos também representou um desafio, exigindo dedicação extra na implementação e compreensão da estrutura de simulação. Além disso, limitações de hardware dificultaram a simulação de cenários maiores, devido à alta demanda por processamento e ao grande volume de dados gerados. Apesar

desses entraves, os conhecimentos adquiridos ao longo do curso e da fomentação desse trabalho foram suficientes para viabilizar a realização de um estudo funcional.

Por fim, este trabalho reforçou a conexão entre a formação acadêmica e a prática profissional. A análise teórica e a experiência com simulações de redes V2X contribuíram para uma visão mais crítica e técnica da área de comunicações veiculares. Dessa forma, o encerramento da graduação com um estudo aprofundado neste tema representa um diferencial significativo para atuação no setor automotivo.

## 6 SUGESTÕES DE TRABALHOS FUTUROS

Como continuação deste estudo, sugere-se a realização de simulações que integrem múltiplos tipos de mensagens C-ITS, como a combinação de CPM com CAM e DENM, a fim de avaliar a competição por canal em cenários com múltiplos serviços ativos. Outra possibilidade é ampliar o ambiente simulado, tanto em área geográfica quanto em densidade de nós simultâneos, para observar os limites operacionais do sistema. Além disso, diferentes modelos de propagação poderiam ser testados e comparados com o Log-Normal Shadowing utilizado neste trabalho, possibilitando a análise do desempenho em ambientes com características físicas variadas. Por fim, propõe-se a avaliação da transmissão de mensagens CPM em redes V2X baseadas em LTE, permitindo um estudo comparativo entre arquiteturas ITS-G5 e soluções celulares de comunicação veicular.

## 7 BIBLIOGRAFIA

- [1] REDES DE COMPUTADORES. [S.I.]: Elsevier, 2003. ISBN 9788535211856. TANENBAUM, A.S.
- [2] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. Automotive Intelligent Transport Systems (ITS). Sophia Antipolis: ETSI, 2022.
- [3] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band; ETSI EN 302 663 V1.2.1. Sophia Antipolis, 2013
- [4] LOBO, Silas; NEUMEIER, Stefan, et al. InTAS - The Ingolstadt Traffic Scenario for SUMO. SUMO Conference Proceedings, p. 73–92, 2020.
- [5] BARBOSA DA SILVA., Leonardo et al. What Is the Right Bounding Box of a VRU Cluster in V2X Communication? How to Form a Good Shape? [S.I.]: SciTePress, 2024. P. 144–155. ISBN 978-989-758-703-0. DOI: 10.5220/0012699100003702. Cit. on p. 20
- [6] S. Lobo, L. B. Da Silva and C. Facchi, "To Cluster or not to Cluster: A VRU Clustering Based on V2X Communication," 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 2023, pp. 2218-2225
- [7] Alvarez Lopez, Pablo and Behrisch, Michael and Bieker-Walz, Laura and Erdmann, Jakob and Flötteröd, Yun-Pang and Hilbrich, Robert and Lücken, Leonhard and Rummel, Johannes and Wagner, Peter and Wießner, Evamarie (2018) Microscopic Traffic Simulation using SUMO. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2575-2582. IEEE. The 21st IEEE International Conference on Intelligent Transportation Systems, 2018-11-04 - 2018-11-07, Maui, USA. doi: 10.1109/ITSC.2018.8569938 <<https://doi.org/10.1109/ITSC.2018.8569938>>.
- [8] Christoph Sommer, Reinhard German and Falko Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," IEEE Transactions on Mobile Computing (TMC), vol. 10 (1), pp. 3-15, January 2011

- [9] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. ETSI EN 302 637-2 V1.3.2 Sophia Antipolis: ETSI, 2014.
- [10] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. Intelligent Transport System (ITS); Vehicular Communications; Basic Set of Applications; Collective Perception Service; Release 2. ETSI TS 103 324 V2.1.1, Sophia Antipolis: ETSI, 2023.
- [11] Špitálová, Zuzana. (2022). Vehicle-to-Everything Communication. Communications - Scientific letters of the University of Zilina. 25. 10.26552/com.C.2023.017.
- [12] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Part 1: General Framework*. ETSI TR 103 257-1 V1.1.1, Sophia Antipolis: ETSI, 2019.
- [13] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *Intelligent Transport Systems (ITS); GeoNetworking; Part 3: Transport Protocol (BTP); Protocol Specification*. ETSI EN 302 636-3 V1.2.1, Sophia Antipolis: ETSI, 2020.
- [14] FOROUZAN, Behrouz A. *Comunicação de Dados e Redes de Computadores*. 5. ed. São Paulo: McGraw-Hill, 2010.
- [15] TANENBAUM, Andrew S.; WETHERALL, David J. *Redes de Computadores*. 5. ed. São Paulo: Pearson, 2011.
- [16] STALLINGS, William. *Redes de Computadores e a Internet*. 6. ed. São Paulo: Pearson, 2014.