



Instituto Superior de Engenharia

Politécnico de Coimbra

DEPARTAMENTO DE INFORMÁTICA E SISTEMAS

Bases de Dados em Grafo na Comunicação entre Veículos Autónomos

Relatório de Estágio para a obtenção do grau de Mestre em
Engenharia Informática

Especialização em Análise Inteligente de Dados

Autor

Daniel de Oliveira Almeida

Orientador

Doutor João Pedro Costa

Supervisores na empresa Ubiwhere

Engenheiro Hélio Simeão / Engenheiro Pedro Teixeira



INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, Novembro 2023

RESUMO

A evolução dos veículos autónomos e a sua integração na infraestrutura rodoviária têm desencadeado uma série de desafios e oportunidades. A partilha de dados entre veículos é uma parte fulcral no que toca aos veículos autónomos, não só para garantir a segurança rodoviária mas também para otimizar as rotas. A enorme quantidade de veículos a circular nas vias coloca vários desafios incluindo a conectividade entre eles e a gestão e comunicação do subconjunto de dados que sejam relevantes para determinados veículos.

O presente relatório apresenta o trabalho realizado ao longo do estágio na Ubiwhere, cujo objetivo era a aplicação de bases de dados em grafo para a gestão e propagação de mensagens entre veículos autónomos aquando da ocorrência de determinados eventos. Neste contexto, surgem vários desafios, nomeadamente a determinação das características e classificação dos nós intervenientes (incluindo veículos e infraestrutura), o modelo de organização desses dados, a forma como vão ser organizados os nós e definidas as ligações entre eles e o modelo de propagação dos dados pertinentes para um subconjunto dos nós.

Neste trabalho são propostos cinco modelos de construção do grafo de propagação de dados, é desenvolvido um protótipo que implementa três desses modelos e realizada uma avaliação experimental.

Para um conjunto de métricas definidas, por exemplo tempo de notificação e tempo de registo no grafo, são identificadas as vantagens e desvantagens de cada modelo e identificados os contextos e requisitos (por exemplo, tempo máximo de notificação) em que é benéfica a sua utilização.

ABSTRACT

The evolution of autonomous vehicles and their integration into the road infrastructure has triggered a series of challenges and opportunities. Sharing data between vehicles is a key part of autonomous vehicles, not only to ensure road safety but also to optimize routes. The sheer number of vehicles on the roads poses several challenges, including connectivity between them and managing and communicating the subset of data is relevant to certain vehicles.

This report presents the work carried out during the internship at Ubiwhere, the aim of which was to apply graph databases to manage and propagate messages between autonomous vehicles when certain events occur. In this context, a number of challenges arise, including determining the characteristics and classification of the nodes involved (including vehicles and infrastructure), the model for organizing this data, how the nodes will be organized and the links between them defined, and the model for propagating the relevant data to a subset of the nodes.

This work proposes five models for constructing the data propagation graph, develops a prototype that implements three of these models, followed by an experimental evaluation.

For a set of defined metrics, such as notification time and time to register in the graph, the advantages and disadvantages of each model are identified and the contexts and requirements (e.g., maximum notification time) in which their use is beneficial are identified.

ÍNDICE

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	2
1.3	Estrutura do relatório.....	2
2	Revisão da Literatura.....	5
2.1	Bases de Dados em Grafo.....	5
2.1.1	Principais vantagens.....	6
2.1.2	Principais desafios e limitações.....	6
2.1.3	Grafo completo e grafo hierárquico	7
2.2	Modelos de comunicação V2V.....	10
2.3	Aplicação de grafos em contextos de redes IoT, veicular e <i>edge</i>	11
2.4	Ferramentas de simulação de tráfego.....	12
3	Utilização de grafos no contexto v2v.....	15
3.1	Gestão centralizada, distribuída ou parcialmente distribuída do grafo.....	15
3.2	Modelo de organização dos dados	17
3.3	Propagação de dados de um evento.....	18
3.4	Envio direto, envio por retransmissão e replicação	19
4	Proposta de Organização e Propagação de Dados de um Evento.....	23
4.1	Classificação dos nós.....	23
4.2	Modelos de grafo de propagação de dados de um evento	25
4.2.1	Ligações no grafo de propagação de dados	29
4.2.2	Atualização do grafo de propagação dos dados	31
4.3	Protótipo implementado	31
4.3.1	Processo de envio de mensagem.....	32
4.3.2	Minimização do envio de mensagens repetidas.....	33
5	Avaliação Experimental	35
5.1	Ambiente de Avaliação	35
5.2	Setup Experimental.....	36
5.3	Importação do mapa de estradas e definição das rotas	37
5.4	Simulações e métricas recolhidas.....	37

5.5	Impacto do envio simultâneo na taxa de insucesso	40
5.6	Impacto do número de mensagens a enviar na utilização de memória	42
5.7	Resultados das simulações.....	43
5.7.1	Avaliação do Tempo de Notificação (TN)	43
5.7.2	Avaliação do Tempo de Registo (TR)	45
5.7.3	Avaliação do Tempo de Armazenamento (TA)	46
5.7.4	Síntese dos resultados	47
6	Conclusões e Trabalho Futuro.....	53
6.1	Trabalho Futuro	53
	Referências bibliográficas	55

ÍNDICE DE FIGURAS

Figura 2.1 – Exemplo de um grafo	5
Figura 2.2 - Grafo completo	8
Figura 2.3 - Grafo hierárquico	9
Figura 3.1 - Grafo centralizado.....	15
Figura 3.2 - Grafo distribuído	16
Figura 3.3 - Modelo hierárquico	17
Figura 3.4 - Veículos elegíveis para o grafo global e para o grafo de propagação de dados de um evento	19
Figura 3.5 - Envio direto	19
Figura 3.6 - Envio por retransmissão.....	20
Figura 3.7 - Grafo de retransmissão sem replicação	21
Figura 3.8 - Grafo de retransmissão com uma réplica.....	21
Figura 4.1 – Definição da relevância (pesos) de cada característica de um nó	25
Figura 4.2 - Organização dos nós no Modelo 1	26
Figura 4.3 - Organização dos nós no Modelo 2	27
Figura 4.4 - Organização dos nós no Modelo 3 (com 2 réplicas – Modelo 32)	28
Figura 4.5 - Método de ligação de um novo nó ao grafo no Modelo 2.....	30
Figura 4.6 - Método de ligação de um novo nó ao grafo no Modelo 3.....	30
Figura 4.7 - Mecanismo de registo dos nós.....	32
Figura 4.8 – Processo de envio da mensagem e do acknowledge no envio por retransmissão.....	33
Figura 5.1 - Rua extraída do OpenStreetMap	37
Figura 5.2 - Taxa de sucesso das mensagens para diferentes números de envios simultâneos no nó com pior pontuação	41
Figura 5.3 - Taxa de sucesso das mensagens para diferentes números de envios simultâneos no nó com melhor pontuação.....	41
Figura 5.4 - Variação da memória utilizada para diferentes números de threads simultâneas	42
Figura 5.5 - TN por modelo e por setup.....	43
Figura 5.6 – Setup 4 a) Num. máximo e médio de nós por caminho por modelo; b) TN por modelo.....	44
Figura 5.7 - TR por modelo e setup.....	45

Figura 5.8 - Número total de mensagens armazenadas e número total de mensagens trocadas por modelo para o Setup 4	46
Figura 5.9 – Relação entre TN, TR e TA	49
Figura 5.10 - Relação entre TN, TA e TR.....	50

ÍNDICE DE TABELAS

Tabela 4.1 - Propriedades de cada nó	24
Tabela 5.1 - Características dos recursos utilizados.....	35
Tabela 5.2 - Número de veículos considerados em cada setup	38
Tabela 5.3 – Simulações consideradas	39
Tabela 5.4 - Resumo dos resultados do tempo de notificação (TN)	47
Tabela 5.5 - Resumo dos resultados do tempo de registo (TR)	48

LISTA DE SIGLAS E ACRÓNIMOS

CPU	Central Processing Unit
DSRC	Dedicated Short Range Communication
GPS	Global Positioning System
IDM	Intelligent Driver Model
IoT	Internet of Things
IP	Internet Protocol
LRU	Last Recently Used
LTE	Long-Term Evolution
MANETSs	Mobile Ad Hoc Networks
P2P	Peer to Peer
RSUs	Road-Side Units
SUMO	Simulation of Urban Mobility
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VANETs	Vehicular Ad Hoc Networks
VLC	Visible Light Communication

1 INTRODUÇÃO

Este relatório apresenta o trabalho efetuado na empresa Ubiwhere, no âmbito da unidade curricular de Estágio do Mestrado em Engenharia Informática, edição 2022/2023, ramo de Análise Inteligente de Dados do Instituto Superior de Engenharia de Coimbra.

1.1 Enquadramento

A evolução dos veículos autónomos e a sua integração na infraestrutura rodoviária têm desencadeado uma série de desafios e oportunidades. Nos últimos anos, têm sido desenvolvidas várias aplicações para evitar acidentes, colisões, ou engarrafamentos de trânsito, utilizando modelos de comunicação *Vehicle to Vehicle* (V2V) e *Vehicle to Infrastructure* (V2I). Esta comunicação interveicular é particularmente relevante no caso dos veículos autónomos, onde é necessária a partilha e recolha de muitos dados, não só para o aumento da segurança rodoviária mas também para uma melhor qualidade da experiência [1].

As redes veiculares, classe derivada das redes móveis, têm como principal objetivo estabelecer comunicação para o fluxo de mensagens entre veículos, habitualmente através da tecnologia de rede sem fios. Esta comunicação perfila-se como um aspeto crucial na operação segura e eficaz de veículos autónomos. A troca eficiente de diversos tipos de informações é relevante não só para garantir a segurança rodoviária mas também para otimizar o desempenho do sistema como um todo, por exemplo para evitar rotas que estejam congestionadas, e melhorar a eficiência das deslocações.

Os veículos autónomos, munidos de vários sensores e câmaras, produzem elevadas quantidades de dados a cada segundo. Estes dados, conjuntamente com dados provenientes de outras fontes, por exemplo semáforos, são utilizados no processo de condução autónoma do veículo. Alguns dos dados recolhidos por um veículo, tais como informação rodoviária recolhida ao longo do percurso, podem também ser pertinentes para outros veículos, sendo relevante a sua partilha. A troca de dados entre os vários veículos é fundamental, fornecendo informações como condições de trânsito, condições climáticas ou outras situações que ocorram nas estradas.

O objetivo deste projeto é explorar a utilização da tecnologia de bases de dados em grafo para, através da localização e proximidade entre veículos, guardar e partilhar informação relevante entre os veículos e modelando cada veículo, ou outro elemento relevante (semáforo, por exemplo), como um nó, e as ligações entre eles como arestas dum grafo. A utilização de grafos oferece a flexibilidade necessária para modelar e reconfigurar a estrutura de comunicação entre os veículos, criando uma

estrutura dinâmica que permite a criação de novos grafos específicos para cada evento, como acidentes ou condições de trânsito excepcionais, que requeira a comunicação entre os veículos. O objetivo é a criação de uma rede de comunicação adaptável e responsiva, projetada para otimizar a eficiência na troca das informações e no seu armazenamento, selecionando os melhores nós para esse efeito.

Os resultados obtidos neste trabalho oferecem orientações para o desenvolvimento futuro de sistemas de comunicação entre veículos autónomos, com o objetivo de otimizar a resposta a eventos críticos e, até, melhorar a segurança nas estradas.

1.2 Objetivos

Os objetivos propostos para este estágio foram os seguintes:

- Propor modelo de dados – organização e estruturação dos dados, das características relevantes de cada nó;
- Propor modelo de registo, classificação e organização do nós – com base nas características e localização dos nós;
- Propor modelos de propagação dos dados – com base na classificação dos nós e na forma como é feita a partilha dos dados entre um subconjunto dos nós interessados na sua receção;
- Implementar os modelo propostos num protótipo;
- Integrar o protótipo com um simulador de tráfego;
- Avaliar os modelos propostos, identificar as suas vantagens e desvantagens e os contextos em que são mais benéficos.

1.3 Estrutura do relatório

Este documento está organizado nos seguintes capítulos:

- Capítulo 1: introdução ao trabalho desenvolvido, apresentando o seu enquadramento, os objetivos propostos e a estrutura do documento;
- Capítulo 2: revisão da literatura, apresentando as bases de dados em grafo, modelos de comunicação entre veículos e contextos de aplicação de grafos;
- Capítulo 3: discussão de desafios emergentes na aplicação e utilização de grafos para a comunicação entre veículos;
- Capítulo 4: apresentação do protótipo para a organização e propagação de dados de um evento;

Bases de dados em grafo na comunicação entre veículos autónomos

- Capítulo 5: avaliação experimental do protótipo implementado, apresentando o ambiente de avaliação, o *setup* experimental e os resultados obtidos;
- Capítulo 6: apresentação da conclusão e trabalho a realizar futuramente sobre o trabalho desenvolvido.

2 REVISÃO DA LITERATURA

Neste capítulo, serão abordados alguns dos principais temas subjacentes ao trabalho descrito neste documento, nomeadamente modelos de comunicação V2V existentes e a aplicação de grafos em vários contextos – IoT, veicular e *edge*.

2.1 Bases de Dados em Grafo

As bases de dados em grafo são um tipo de base de dados NoSQL. São do tipo não relacional e fornecem uma solução eficaz para o armazenamento de informações em cenários onde os dados estejam fortemente interligados [2]. Este tipo de bases de dados tem vindo a ganhar relevância com o crescimento das aplicações ricas em relações (redes sociais, por exemplo) e devido às suas aplicações em áreas como a Web Semântica [3]. Pese embora a predominância das bases de dados relacionais no mercado, a sua adaptabilidade é fraca no processamento de dados conectados [2].

Um grafo é composto por vértices (habitualmente designados nós), que representam as entidades num determinado domínio - pessoas, locais, objetos, etc - e arestas, que conectam os vértices e representam as ligações entre eles, sendo fundamentais para a construção e organização das informações na estrutura da base de dados [2][4]. Tanto os nós como os vértices podem conter uma variedade de informações (propriedades) cuja finalidade é descrever as suas características. Para além disso, podem também ser etiquetados com etiquetas que representam as suas diferentes funções no domínio da aplicação[5]. Cada vértice mantém referências aos seus vértices adjacentes, dando origem a uma espécie de índice para os vértices na vizinhança. A Figura 2.1 apresenta um exemplo de um grafo, onde estão representados três vértices (*Person*, *Person* e *Car*) e cinco arestas (*Loves*, *Loves*, *Lives With*, *Owns* e *Drives*). Informações como *name* ou *born* nos vértices e *since* na aresta são propriedades.

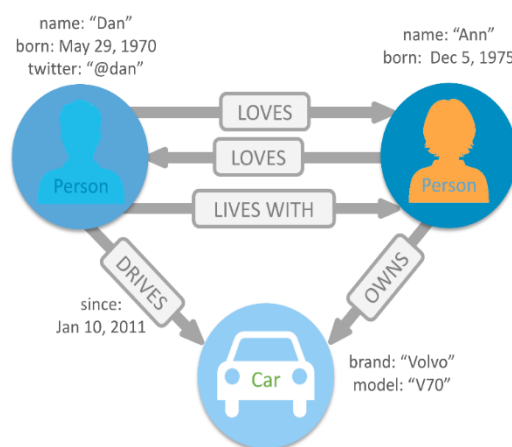


Figura 2.1 – Exemplo de um grafo

Fonte: Neo4j (<https://neo4j.com/developer/graph-database/>)

2.1.1 Principais vantagens

As bases de dados em grafo estão a tornar-se cada vez mais populares à medida que o volume de dados aumenta, bem como a conectividade entre eles. Não só armazenam eficazmente as relações entre os dados, como são também flexíveis na adição de novos tipos de relações ou na adaptação de um modelo de dados a novos requisitos comerciais [2].

Uma das principais vantagens deste tipo de bases de dados é a melhor otimização na obtenção de dados comparativamente às bases de dados relacionais, uma vez que aproveita os dados de proximidade de uma ou mais raízes (nós principais) da base de dados. São também bastante intuitivos, devido à forma de representação dos dados e suportam o armazenamento de dados na ordem de *petabytes*. São adequados para dados conectados entre si, o que geralmente se enquadra em casos do mundo real e muito ágeis no desenvolvimento, uma vez que podem ser facilmente adaptados ao longo do tempo. Para além disso, comparativamente com bases de dados relacionais, as bases de dados em grafo possuem um alto desempenho em termos de consulta de pesquisas muito profundas. Devido ao seu modelo estrutural – grafo – são facilmente expansíveis e não é necessário declarar o tipo de dados para os nós ou arestas, ao contrário do que acontece no modelo relacional [2], [3].

Em poucas palavras, as vantagens podem resumir-se em essencialmente três palavras: desempenho, flexibilidade e agilidade. Desempenho porque, ao contrário das bases de dados tradicionais, onde as consultas tem a tornar-se mais lentas à medida que o número e a profundidade de relações aumenta, este mantém-se constante mesmo com o aumento dos dados nas bases de dados em grafo. Flexibilidade porque a estrutura e o esquema de um modelo de grafo ajustam-se à medida que as aplicações e as indústrias sofrem alterações. Ou seja, é possível adicionar à estrutura de grafo existente sem colocar em risco a funcionalidade atual. Por fim, agilidade. O desenvolvimento com bases de dados em grafo alinha-se com as atuais práticas de desenvolvimento ágeis e orientadas para testes, permitindo que a base de dados evolua em sintonia com o resto da aplicação e seja capaz de responder a quaisquer mudanças nos requisitos comerciais [2].

2.1.2 Principais desafios e limitações

O aumento de procura e utilização das bases de dados em grafo criou uma série de novos e interessantes problemas e desafios. Em primeiro lugar, as formas mais eficientes de computar as respostas às consultas ao grafo, mais concretamente padrões do grafo, consultas de caminho e combinações entre ambos [6]. Para além disso, a grande maioria das bases de dados em grafo comerciais não suportam linguagem declarativa [5]. Outra das principais limitações é o facto deste tipo de bases de dados não serem particularmente eficaz em todas as operações desejadas, nomeadamente na representação de dados dos modelos relacionais [3]. Também a escalabilidade pode ser comprometida, especialmente em casos de grafos com

muitos nós e arestas. Também a integração de dados de diferentes fontes também é dificultada, uma vez que as bases de dados em grafo são altamente dependentes da qualidade e consistência dos dados de entrada.

Assim, este tipo de bases de dados não substituem as bases de dados relacionais, mas apresentam-se como uma solução eficiente quando se trata de grandes volumes de dados que contêm muitos dados relacionados [2].

2.1.3 Grafo completo e grafo hierárquico

No contexto veicular, os vértices do grafo representariam os veículos e as arestas representariam, por exemplo, a distância ou o custo de transmissão entre dois veículos. A comunicação entre dois ou mais veículos poderá ser efetuada se existir uma ligação direta entre cada nó (grafo completo) ou indiretamente através de nós intermédios (grafo hierárquico). O grafo completo conteria todos os veículos e todos os relacionamentos entre eles.

É fundamental compreender as diferenças entre estes tipos de grafos e respetivas vantagens e desvantagens, para que possa ser feita uma escolha ponderada mediante os requisitos em questão.

O grafo completo é um método que considera a ligação de cada nó da base de dados a todos os restantes, sendo uma abordagem atrativa em vários casos, uma vez que permite que todas as informações estejam acessíveis a partir de qualquer ponto da base de dados .

Esta abordagem é conhecida como grafo completo, em que o grafo K_n , sendo n o número de nós do grafo, terá

$$\frac{n(n - 1)}{2}$$

arestas. No exemplo apresentado na Figura 2.2, o grafo com 9 nós teria um total de 36 ligações.

No entanto, esta abordagem tem algumas desvantagens, nomeadamente:

- Custo computacional – para grandes grafos, exige muitos recursos computacionais na sua construção e processamento, originando tempos de resposta mais lentos e problemas de escalabilidade.
- Espaço de armazenamento – grande espaço de armazenamento para guardar todas as ligações.

- Dificuldade de interpretação – a interpretação do grafo pode tornar-se visualmente difícil em grafos muito grandes, dada a elevada conectividade. Dificulta a identificação de padrões e relações entre os dados.

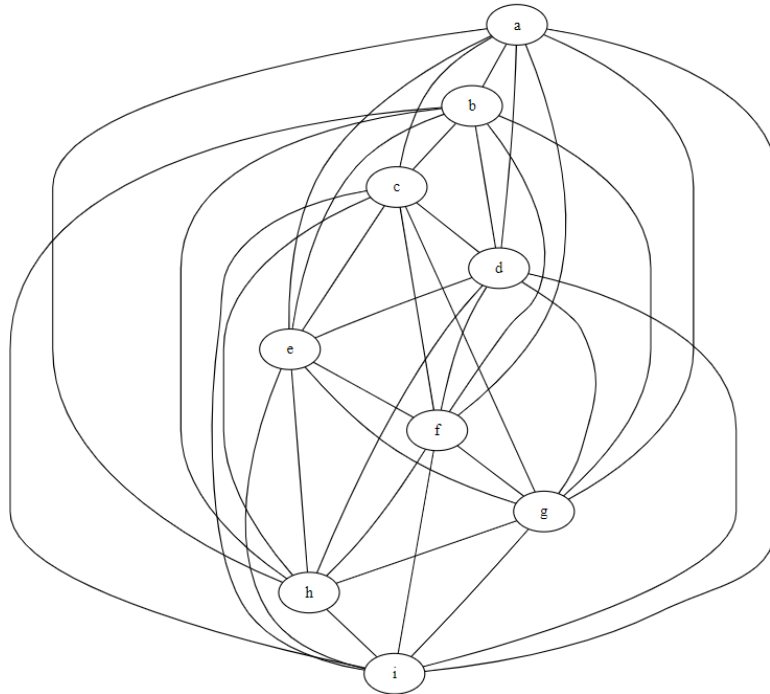


Figura 2.2 - Grafo completo

No grafo hierárquico é utilizada uma abordagem hierárquica com o objetivo de reduzir o número de ligações do grafo e estruturar o grafo de forma mais organizada e facilitar o processo de atualização.

Nesta solução, alguns dos nós são usados como nós de concentração/encaminhamento, doravante denominados como nós líderes, de modo a minimizar o número de ligações/conexões entre os nós.

A eleição do nó líder poderá ser feita seguindo vários critérios:

- Seleção aleatória – o nó líder é aleatoriamente selecionado como líder, de entre todos os nós. A facilidade de implementação é a sua principal vantagem, no entanto o nó escolhido pode ser um nó com menor capacidade de processamento em relação a outros nós do grafo;
- Seleção baseada em métricas – o nó líder é selecionado com base nas suas métricas / características, por exemplo a seleção do nó com maior capacidade de processamento. É mais complexo em relação à seleção aleatória, uma vez que é necessária a existência de um processo de registo e recolha das métricas que caracterizam o nó. Poderá ser considerada a

possibilidade de, caso seja adicionado um novo nó com melhor qualificação que o líder atual, proceder-se à reformulação da hierarquia e consequentemente todas as ligações, assumindo o novo nó como líder, de modo a garantir que o nó líder é melhor que os restantes;

- Seleção por ordem de chegada – o nó líder é o primeiro nó a ser adicionado ao grafo. Não garante que o nó selecionado seja o melhor, mas por outro lado é de mais fácil implementação uma vez que apenas existe necessidade de redefinição do líder caso o nó líder seja removido.

Todos os nós não líderes são doravante denominados de nós comuns. Cada nó comum cria ligação apenas para os nós líderes, e os nós líderes criam ligação entre si.

Em relação à abordagem anterior, ao invés de se estabelecer uma ligação direta entre cada par de nós, poder-se-á estabelecer uma organização hierárquica com vários níveis, em que cada nível representa um subconjunto de nós, agregados por alguma característica (por exemplo a sua proximidade). Cada um destes níveis terá um nó líder (nó agregador) que será responsável por garantir a ligação aos nós do seu nível. Assim, o encaminhamento será efetuado indiretamente através do nó líder.

Considerando o mesmo número de nós (9) e 3 níveis diferentes, são criadas apenas 9 ligações, apresentando uma redução de aproximadamente 75%, como exemplificado na Figura 2.3.

Esta estrutura facilita a consulta e a análise dos dados, uma vez que todos os nós não líderes ficam acessíveis através do seu líder, bem como a reorganização do grafo uma vez que o número de arestas é menor em relação ao grafo completo.

Como desvantagem, o nó líder pode acabar por se tornar um ponto crítico do grafo, concentrando nele muitas informações e exigindo por isso mais recursos computacionais.

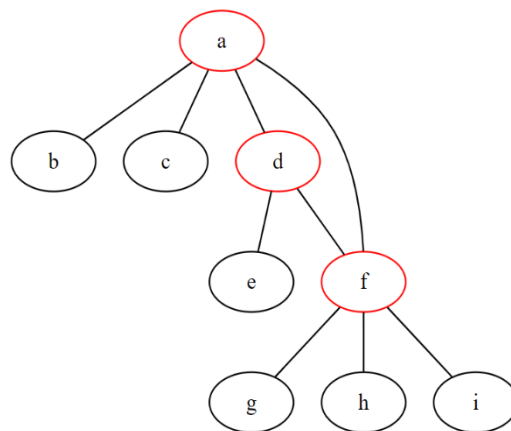


Figura 2.3 - Grafo hierárquico

2.2 Modelos de comunicação V2V

Ao longo dos últimos anos, têm sido feitos vários estudos para desenvolver os recursos dos modelos de comunicação V2V (*Vehicle to Vehicle*), com o objetivo de permitir uma comunicação mais robusta e segura entre os veículos.

Existem diversos estudos – nomeadamente [7], [8], [9], [10] e [11] – que abordam as VANETs (*Vehicular Ad Hoc Networks*), uma forma de MANETs (*Mobile Ad Hoc Networks*) em que os veículos na estrada formam os nós da rede. Estas redes têm como objetivo permitir a comunicação entre veículos e dispositivos, tornando possível a troca de informações em tempo real, nomeadamente alertas de segurança, dados de navegação ou informações de trânsito. Existem várias técnicas de disseminação de dados nestas redes, como V2I (*Vehicle to Infrastructure*), V2V ou P2P (*Peer to Peer*), conforme apresentado e discutido em [12]. Tirando partido destas técnicas, os autores [7] propõem um método de partilha de ficheiros P2P, utilizando o protocolo TC-MAC, cujo objetivo é melhorar o tempo de download dos ficheiros entre veículos vizinhos. Também em [8] são tomadas por base as VANETs e abordados alguns problemas relacionados com a troca de informações entre veículos, nomeadamente a redundância da informação, o que de certa forma degrada a eficiência da rede. Os autores sugerem um protocolo de comunicação para a disseminação de mensagens de emergência, em que o veículo candidato a encaminhar uma mensagem é selecionado de acordo com a sua distância ao veículo fonte na direção de propagação da mensagem. Esta abordagem apresenta semelhanças com os modelos propostos neste trabalho, uma vez que, nos casos onde a mensagem não é enviada de forma direta entre o emissor e o recetor, é encaminhada através de um ou mais veículos, através de um caminho definido tendo em conta a distância entre os veículos. Dada a elevada mobilidade dos veículos e as mudanças nas condições de tráfego, a topologia das VANETs é altamente dinâmica – tópico abordado em [9], que discute desafios do processamento de dados veiculares e sugere tecnologias de otimização baseadas em grafo. Tendo em conta este desafio, [10] analisa a conectividade e o potencial de criação de comunidades neste tipo de redes. Além disso, é estudado o desempenho de um algoritmo descentralizado para a deteção de comunidades – proposto em [11]. São sugeridas como potenciais melhorias do algoritmo de deteção de comunidades a criação de comunidades hierárquicas, ou a atribuição de um veículo a uma comunidade tendo por base uma métrica de similaridade. Também no trabalho apresentado neste documento pode considerar-se o conceito de comunidade: conjunto de veículos que constituem o subgrafo relativo a cada ocorrência. No entanto, neste caso a comunidade é definida tendo por base a distância entre a ocorrência e os veículos, enquanto em [10] o objetivo é o agrupamento de nós tendo por base propriedades semelhantes de acordo com alguma métrica pré-definida.

Tendo em consideração a elevada frequência de desconexão de redes veiculares, os autores [13] propõem uma solução em que o veículo em movimento descarrega determinado pacote de dados até que um novo veículo se mova na sua

vizinhança e encaminhe o pacote. São propostos vários protocolos de entrega de dados para encaminhar o pacote pelo melhor caminho e com o menor atraso possível na sua entrega. Também em [14] é abordada a troca de dados através de veículos, utilizando as trajetórias da rede de estradas previstas para calcular probabilidades de entrega dos pacotes. O objetivo é melhorar métricas como a taxa de entrega, sobrecarga de comunicação e o atraso na entrega.

A cooperação de várias tecnologias de comunicação e informação nas redes veiculares provoca uma grande heterogeneidade nas VANETs. Inicialmente, dependiam principalmente do DSRC (*Dedicated Short Range Communication*) para estabelecer comunicações veicular. No entanto, a escassez de largura de banda desta norma levou os investigadores a adotar outras ideias, como Wi-Fi, 5G, Bluetooth ou LTE. Wi-Fi é a mais difundida, devido ao seu custo aceitável, taxa de dados mais elevada e flexibilidade. Já o LTE (*Long-Term Evolution*) pode fornecer uma taxa de dados elevada, latência minimizada e uma grande região de cobertura. No entanto, as VANETs baseadas em LTE restringem a comunicação a V2I. O Bluetooth é conhecido como uma das melhores tecnologias utilizadas nas comunicações de curto alcance, sendo utilizado principalmente em comunicações intraveículo, como serviços de infoentretenimento ou serviços de navegação. O VLC (*Visible Light Communication*), inventado em 2012, proporciona comunicações óticas sem fios de comprimento de onda curto. Uma das vantagens da sua utilização é que a interferência de outros sinais eletromagnéticos é quase nula e é menos vulnerável a ataques de segurança [15]. O foco deste trabalho não se prende com uma tecnologia de comunicação específica, e considera que é possível a comunicação entre nós através da camada protocolar IP.

2.3 Aplicação de grafos em contextos de redes IoT, veicular e edge

No contexto veicular, existem já algumas abordagens da modelação de uma rede de veículos na forma de grafo. Em [16], é modelada uma rede onde cada veículo é representado como um nó do grafo, e cada comunicação V2V como uma aresta. Também em [17] a estrutura de dados tem por base um grafo acíclico, que armazena as transações entre os nós da rede – são representados como nós os veículos e RSUs (*Road-Side Units*). Na rede proposta por [16], cada veículo tem a sua própria tabela de roteamento sobre as informações GPS dos veículos que o rodeiam. Quando necessita de saber as informações de tráfego, consulta os veículos vizinhos na tabela de roteamento.

Também no contexto da IoT (*Internet of Things*), onde um dos principais obstáculos é lidar com o volume crescente de dados e com várias fontes e tipos de dados, a gestão dos dados pode ser feita tirando proveito da utilização de grafos – conforme discutido em [18], onde é feita uma revisão e são estudadas três abordagens diferentes: bases de dados SQL, bases de dados NoSQL e bases de dados em grafo. A aplicação de grafos neste contexto é uma forma eficaz e adequada para

a gestão dos dados porque são capazes de determinar a comunicação de uma forma simples entre os vários nós constituintes da rede IoT. Este estudo conclui que as bases de dados em grafo, quando comparadas com as bases de dados SQL, são capazes de otimizar a escalabilidade e a flexibilidade ao lidar com diferentes tipos de dados em simultâneo. Por outro lado, as bases de dados SQL conseguem uma melhor estabilidade e maior segurança. Em [19], são associadas redes neuronais com grafos, considerando como vértices os dados de sensores e como arestas as relações entre vértices. Neste estudo, o objetivo passa por modelar as relações entre os dados dos sensores e simular o funcionamento do equipamento IoT. As redes neuronais são utilizadas para aprender as relações entre os dados. No estudo [20], os autores propõem um *middleware* IoT multicamada baseado num grafo de conhecimento, cujo objetivo é a gestão uniforme das comunicações de todos os dispositivos.

A aplicação de grafos tem sido também usada para melhorar a cibersegurança no contexto de *edge computing*, técnica onde o processamento dos dados é efetuado nas extremidades da rede, ou seja, mais próximo dos sensores e outros equipamentos geradores de dados, em vez de num controlador central. A sua principal vantagem é a redução da latência e dos requisitos de largura de banda. Em [21], é proposto um mecanismo baseado em *edge computing* que deteta e atenua automaticamente a falsificação de endereços e intrusões na rede. O método tem por base a computação de grafos para identificar a localização de um atacante ou intruso. O grafo é criado através da topologia da rede, em que as arestas representam as ligações e os nós representam os vértices. Partindo destes dois componentes, é obtido o estado da rede em diferentes intervalos de tempo. A aplicação da técnica de diferença de grafos é capaz de identificar a localização de um atacante ou intruso, bem como um utilizador legítimo.

2.4 Ferramentas de simulação de tráfego

Os avanços significativos nos sistemas de transporte inteligentes contribuíram para um maior desenvolvimento da modelação do tráfego, incluindo modelos de previsão e simulação utilizados para prever o comportamento do tráfego em estradas e redes urbanas [22].

No que toca a algoritmos de simulação, existem diversos, nomeadamente [22]:

- *Car following models* – têm como objetivo descrever a forma como os veículos simulados interagem com o veículo que os precede na mesma faixa de rodagem;
- *Lane changing models* – têm como objetivo simular o impacto dos veículos nas faixas adjacentes quando mudam de faixa. Este modelo avalia o espaço existente para determinar se é possível mudar de faixa;
- *Gap acceptance models* – têm como objetivo determinar as condições de tráfego nas vias adjacentes antes de um veículo aceder ao espaço

disponível. Estes modelos são utilizados para estimar a quantidade de espaço e tempo necessários para atravessar um cruzamento, entrar numa rotunda ou muda de faixa.

Quanto às ferramentas de simulação mais utilizadas, destacam-se [22]:

- The Verkehr In Städten – Simulationsmodell (VISSIM) – ferramenta comercial de simulação de tráfego microscópico comumente utilizada para simular e avaliar o estado do tráfego e os sistemas de controlo dos transportes;
- Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks (AIMSUN) – ferramenta comercial capaz de simular situações reais de tráfego numa rede urbana com o objetivo de construir e validar estruturas de tráfego, redes de transportes públicos e novas infraestruturas de transporte;
- Multi-Agent Transport Simulation (MATSim) – ferramenta *open-source* cujo foco é o comportamento individual dos veículos e não no comportamento do tráfego em geral como os dois anteriores. Simulador baseado em agentes, que possuem uma lista que simula a rotina diária do tráfego;
- Simulation of Urban Mobility (SUMO) – ferramenta *open-source* capaz de simular o tráfego ao nível microscópico e simula veículos em movimento e acidentes. Concebido como um simulador de tráfego intermodal que inclui transportes públicos, redes rodoviárias de tráfego e peões.

3 UTILIZAÇÃO DE GRAFOS NO CONTEXTO V2V

A utilização de grafos no contexto V2V coloca alguns desafios, nomeadamente na forma de organização e gestão do grafo (centralizado, distribuído, etc...) e na forma como os nós são organizados e como são estabelecidas as respetivas relações. A organização, e o tipo de grafo, são particularmente relevantes em contextos com um elevado número de veículos, que coloca exigências ao sistema (servidor) de gestão da base de dados de grafos.

Neste capítulo, discutem-se os prós e contras de cada abordagem.

3.1 Gestão centralizada, distribuída ou parcialmente distribuída do grafo

Numa base de dados centralizada, todos os nós e arestas do grafo são armazenados num único servidor, ao contrário do que acontece na solução distribuída, onde o grafo é dividido em várias partes e cada uma delas é armazenada num servidor diferente.

A solução centralizada (Figura 3.1) apresenta como vantagem o facto de ser mais fácil de gerir e atualizar, uma vez que o grafo se encontra todo num único servidor. Para além disso, oferece maior consistência e integridade dos dados, já que dispensa mecanismos de sincronização e atualização entre os vários servidores que seriam necessários na solução distribuída. Como limitação, aponta-se a maior vulnerabilidade a falhas do sistema, uma vez que um único servidor é responsável pela totalidade da base de dados, bem como a possível existência de

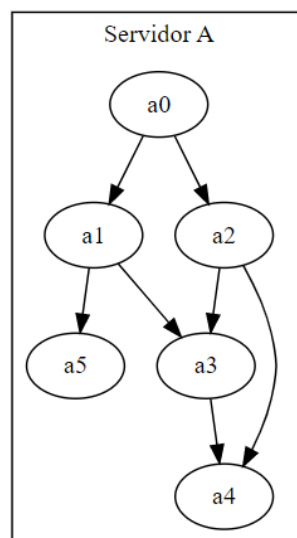


Figura 3.1 - Grafo centralizado

constrangimentos no tráfego, uma vez que todos os pedidos passarão pelo único servidor.

Numa solução distribuída o grafo é dividido em várias partições e armazenado em diferentes servidores (Figura 3.2). Esta solução escalável é mais resiliente, uma vez que a falha de um servidor não implica a perda total do grafo. Contudo, a sua gestão e manutenção é mais complexa, uma vez que requer a coordenação entre vários servidores, e a consistência e integridade dos dados pode ter menos garantias comparativamente com a solução centralizada. A latência em algumas operações pode ser superior à da solução centralizada, pois um pedido pode requerer o processamento em vários servidores.

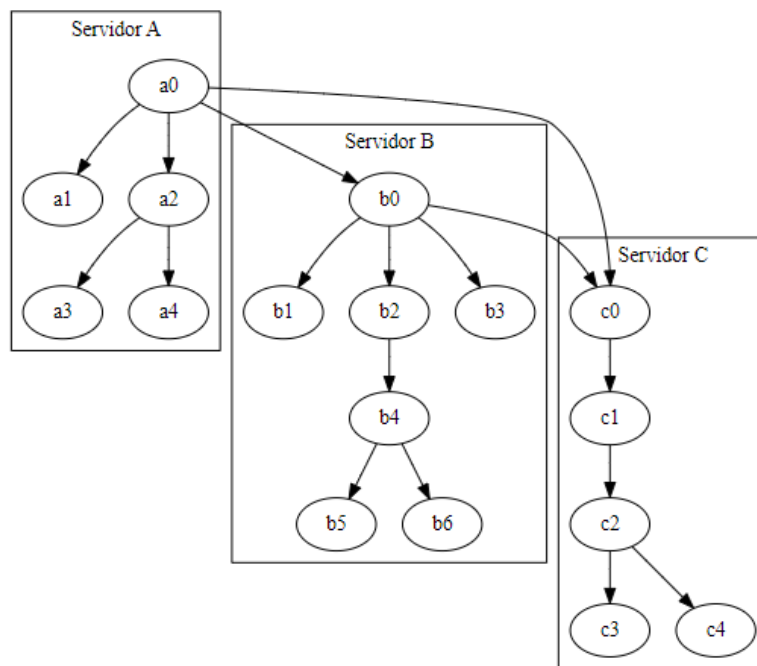


Figura 3.2 - Grafo distribuído

Para além das soluções centralizada e distribuída, pode ainda considerar-se uma terceira solução: a parcialmente distribuída – em que parte do grafo é armazenado e processado de forma distribuída e outra parte de forma centralizada. Nesta abordagem, algumas das partes (ou subgrafos) que resultam da divisão da totalidade do grafo podem ser distribuídas em diferentes servidores enquanto outras são armazenadas de forma centralizada. A principal vantagem desta solução é obter os benefícios de cada uma das abordagens anteriores: simplicidade e consistência dos dados da centralizada, e a escalabilidade da distribuída.

Embora as mais valias oferecidas pela solução mista, devido às limitações dos recursos disponíveis, neste trabalho foi considerada a solução centralizada, não sendo por isso analisadas questões relacionadas com a replicação e consistência de grafos entre servidores.

3.2 Modelo de organização dos dados

Num contexto veicular, devido ao elevado número de veículos que podem ser considerados, que podem variar com a entrada e saída de novos veículos (por exemplo, que anteriormente estavam desligados) e a necessidade de registar as respetivas ligações de conectividade entre eles, é vantajosa a opção pelo modelo de organização hierárquica em detrimento do grafo completo – sobretudo para minimizar os custos e tempos de computação, uma vez que desta forma, quando um novo nó se liga, apenas necessita de estabelecer ligação com o nó líder, e não com todos os nós que constituem o grafo.

A Figura 3.3 ilustra um exemplo de um modelo hierárquico onde estão representados os dois tipos de nós: os nós líderes (a vermelho) e os nós comuns (a preto). Um nó líder é o primeiro nó de cada rede diferente registada no grafo. Os restantes nós, comuns, são todos os outros nós de uma rede que são registados no grafo depois do nó líder. Considera-se como uma rede um conjunto de dispositivos conectados virtualmente numa rede sem fio, que geralmente operam sob um conjunto de políticas e configurações de rede comuns. Na Figura 3.3 estão representados três nós líderes de três redes diferentes. Os nós líderes estabelecem ligação com todos os restantes líderes, enquanto os nós comuns estabelecem ligação com o respetivo líder.

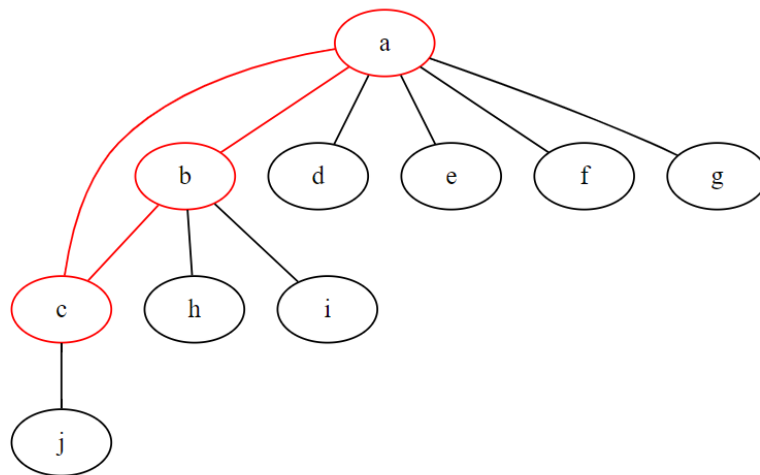


Figura 3.3 - Modelo hierárquico

A informação relativa a esta organização de ligação entre nós, que no contexto veicular representam veículos, poderá ser utilizada no processo de comunicação de informações entre veículos. Há informação que pode ser pertinente para todos os veículos, mas algumas informações, ou eventos, podem ser pertinentes apenas para um subconjunto de veículos, por exemplo, que se encontram próximo, ou numa zona. Neste caso, ao invés de se utilizar o grafo com a globalidade dos veículos (grafo global), um subconjunto de veículos pode ser guardado num grafo local (ou grafo do evento), derivado do grafo global, de menor dimensão e consequentemente com

menores tempos de acesso, pois o número de nós influencia o tempo de consulta do grafo.

3.3 Propagação de dados de um evento

Quando ocorre um evento e é necessário propagar essa informação pelos veículos próximos, ou utiliza-se o grafo global ou cria-se um grafo para cada evento.

A utilização do grafo global para a propagação de mensagens dos vários eventos apresenta vantagem na facilidade de implementação e na economia de recursos, uma vez que é mais simples manter apenas um grafo e alocar recursos para apenas um grafo. No entanto, seria complexo na medida em que as ligações em cada grafo de propagação de dados de um evento podem ser diferentes das ligações existentes no grafo global. Ou seja, no grafo global, os nós encontram-se organizados segundo uma estrutura hierárquica onde existe um líder (o primeiro nó da rede a entrar no grafo) e os nós comuns (os restantes nós da rede) – apresentada na secção 3.2. Já a organização dos nós, no grafo de propagação, deve constituir o caminho que a mensagem vai seguir até ao nó de destino, ou seja, pode não seguir a organização hierárquica da rede. As ligações poderão ser efetuadas considerando, por exemplo, a proximidade entre os veículos. No capítulo 4 são apresentados alguns modelos de ligação. Para além disso, um mesmo veículo pode constar em dois grafos de propagação de dados de um evento diferentes (dois eventos distintos), levando a que as suas ligações possam ser para diferentes veículos. A utilização do grafo global para representar todas estas ligações poderia levar à perda de organização e noção dos nós pertencentes a cada evento e dos nós que já foram notificados para cada evento.

Já a criação de um novo grafo para cada evento tem como vantagem o facto de apenas os veículos que vão receber informação desse evento o constituírem, não existindo nesse grafo veículos que não possuam qualquer interesse na receção de mensagens relacionadas com esse evento. No entanto, esta abordagem é mais complexa na medida em que é necessária a criação e gestão de vários grafos e, caso existam muitos eventos ou muitos veículos, pode levar à criação de um grande *overhead*.

Devido, sobretudo, à necessidade de criar novas ligações no grafo de propagação de dados de um evento, em relação ao grafo global, optou-se pela segunda opção: criação de um grafo de propagação para cada evento. Assim, sempre que ocorre um evento, é criado um grafo inicial, que será composto pelos veículos presentes no raio definido previamente (relativamente ao local do evento). Este grafo é mantido até que o evento termine, sendo eliminado depois disso. Durante o período de atividade, pode sofrer várias alterações, nomeadamente no que diz respeito aos nós que o constituem, e às relações entre eles.

A Figura 3.4 apresenta, de forma esquematizada, a lógica de criação de um grafo para cada evento. A azul, está representado o grafo global, que deve manter registo de todos os veículos, sendo cada veículo representado por um nó com suas

características. A verde, o grafo de propagação de dados de um evento, neste caso representando um acidente. Compõem este grafo os veículos que se encontrem num raio previamente definido, sendo um grafo preparado para lidar com entrada e saída de veículos ao longo do período temporal em que está ativo, bem como alterações nas respetivas ligações entre os nós.

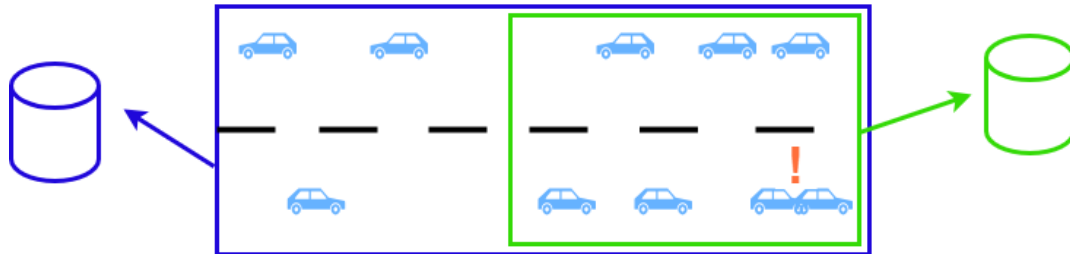


Figura 3.4 - Veículos elegíveis para o grafo global e para o grafo de propagação de dados de um evento

3.4 Envio direto, envio por retransmissão e replicação

O envio das mensagens pode ser realizado de duas formas distintas: envio direto e envio por retransmissão.

No envio direto, a mensagem é enviada diretamente do remetente, ou seja, o veículo que armazena a informação, para o destinatário, o veículo que pretende receber a mensagem, conforme representado na Figura 3.5. Ou seja, a mensagem é transferida ao longo de uma única aresta que conecta os dois nós. O remetente carrega o conteúdo armazenado no ficheiro correspondente ao evento e transfere para o destinatário como mensagem.

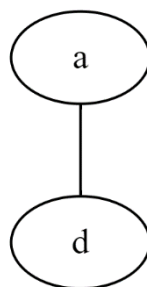


Figura 3.5 - Envio direto

Já no envio por retransmissão, a mensagem é enviada através de um, ou mais, nós intermediários até alcançar o destinatário – Figura 3.6. Estes nós intermediários funcionam como pontos de passagem, sendo responsáveis por encaminhar a

mensagem para o nó seguinte até que seja atingido o nó final. O remetente (nó “a”) transmite a mensagem para o nó seguinte (nó “b”). Este nó seguinte é responsável por passar a informação ao próximo nó (nó “c”) e assim sucessivamente, até chegar ao nó de destino (nó “d”).

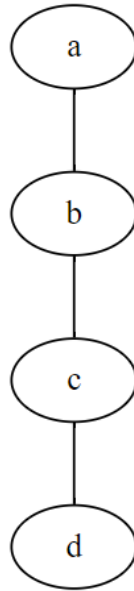


Figura 3.6 - Envio por retransmissão

Como vantagens do envio direto, em relação ao envio por retransmissão, apontam-se:

- Menor latência, uma vez que não existem atrasos causados pelas retransmissões;
- Menor sobrecarga na rede, já que a mensagem é trocada diretamente entre dois nós, não necessitando da passagem por nós intermediários;
- Simplicidade do processo, exigindo apenas o processo de envio de um nó para outro, não sendo necessário desenvolver o processo de retransmissão.

Já o envio por retransmissão apresenta algumas vantagens, como:

- Tolerância a falhas, no caso de falhar uma ligação intermédia. Neste caso, pode ser estabelecida outra ligação intermédia que permita alcançar o mesmo destino;
- Flexibilidade no que toca à exploração de vários caminhos com vista a otimizar o processo de retransmissão.

Relativamente à replicação, assume-se como uma vantagem em vários aspetos, nomeadamente:

- Tolerância a falhas, na medida em que em caso de falha na ligação de um dos veículos que armazena a mensagem, outro veículo pode ficar encarregue dessa tarefa;
- Menor latência, uma vez que tendo a mensagem armazenada em vários veículos aumenta a probabilidade de um deles estar mais próximo do nó de destino, reduzindo o tempo de envio e receção;
- Redução do consumo de energia, já que pode levar a uma melhor distribuição do consumo de energia dos veículos, uma vez que vários veículos são responsáveis pelos envios e não apenas um.

Nos casos onde é utilizada retransmissão, a replicação pode ser útil também na redução da profundidade dos grafos de transmissão, isto é, na redução do número de nós necessários para a passagem da mensagem até ao nó final – conforme representado na Figura 3.7 e na Figura 3.8. Na Figura 3.7, onde apenas o nó *a* armazena a mensagem, para que a mensagem chegue ao nó *f* é necessária a utilização de dois nós intermédios. Já na Figura 3.8, com uma réplica, para atingir o nó *f* apenas é necessária a utilização de um nó intermédio, sendo o destinatário o nó *d* e não o *a*.

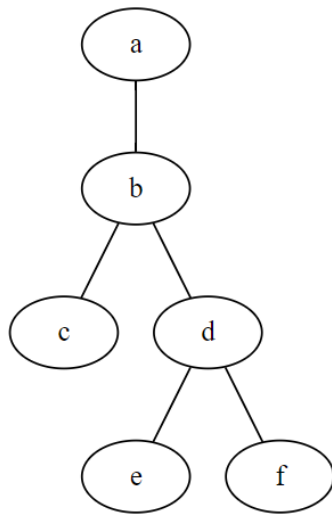


Figura 3.7 - Grafo de retransmissão sem replicação

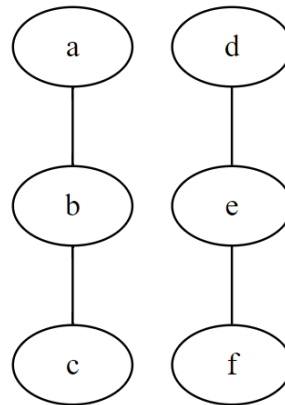


Figura 3.8 - Grafo de retransmissão com uma réplica

4 PROPOSTA DE ORGANIZAÇÃO E PROPAGAÇÃO DE DADOS DE UM EVENTO

Para o processo de comunicação de um evento ou ocorrência, por exemplo um acidente, quando a informação é pertinente para apenas um subconjunto de nós, propõe-se uma abordagem flexível em que a comunicação propagada é proveniente do local da ocorrência.

Desta forma, aquando de uma ocorrência, é construído um grafo de propagação da informação, mais pequeno, centrado na proximidade da ocorrência, contendo os nós interessados na receção dos dados dessa ocorrência.

De uma forma geral, a proposta passa pela construção e gestão de vários grafos, de acordo com uma ocorrência ou evento. O grafo global, utilizando o contexto centralizado, armazena todos os nós e respetivas métricas e características, e está em constante atualização, de modo a manter os dados que caracterizam cada nó sempre atualizados. Para além deste grafo, é criado um novo grafo para cada evento ou ocorrência, grafo propagação de dados do evento, que é constituído pelos nós com interesse na receção da mensagem. O critério de interesse, que leva à inclusão no grafo, pode ser baseado em diversas características, por exemplo a proximidade ao local do evento ou veículos que estejam em movimento em direção ao local do evento. Neste trabalho, foi considerado o critério de proximidade ao local de ocorrência do evento, ou seja, todos os veículos (do grafo global) presentes numa área de alcance previamente definida, em relação à localização do evento, serão adicionado a este grafo. A saída ou entrada de veículos do raio de alcance levará à atualização do grafo de propagação de dados do evento. Estes grafos são responsáveis pela gestão da partilha da informação, bem como pela gestão dos nós que já receberam a mensagem.

Neste capítulo, é apresentada a proposta mais detalhadamente.

4.1 Classificação dos nós

Num sistema heterogéneo com vários nós, existirão nós com características diferentes, nomeadamente no que concerne à capacidade de processamento e/ou armazenamento. O objetivo é que, quando existir necessidade de selecionar um ou mais nós (por exemplo para armazenamento de dados), possam ser selecionados aqueles que reúnem melhores características – uma vez que serão mais solicitados para a partilha da informação. Num contexto veicular, os nós de infraestrutura provavelmente terão capacidade de processamento e de comunicação superior à da maioria dos veículos.

No modelo proposto, para cada nó presente no grafo são registadas, como propriedades, um conjunto de características e métricas que o descrevem e o

qualificam. A Tabela 4.1 apresenta as características e métricas armazenadas em cada nó.

Tabela 4.1 - Propriedades de cada nó

Nome	Descrição
id	Identificador único do nó
node_type	Tipo de nó (veículo, infraestrutura, etc)
cpu_logical_cores	Número de núcleos físicos de CPU
cpu_physical_cores	Número de núcleos lógicos de CPU
cpu_type	Tipo de CPU
disk_free	Quantidade de memória livre (Mb)
disk_used	Quantidade de memória utilizada (Mb)
hostname	Hostname
ip_address	Endereço IP
lat	Latitude
lon	Longitude
node_score	Pontuação do nó
percent_disk_used	Porcentagem de memória utilizada
physical_memory	Memória física total (Mb)
timestamp	Registro de data e hora da última atualização

Através das métricas e características de cada nó é possível definir uma função que permite qualificar o nó, utilizando uma métrica *node_score*. Este valor é recalculado e atualizado à medida que as restantes características são também atualizadas. Ou seja, desde que um nó é criado até que seja removido, as suas métricas dinâmicas (aquelas cujo valor tem tendência a variar) mantêm-se em atualização com determinada frequência. O nó será tanto melhor quanto maior for esse valor. Para o cálculo dessa pontuação, podem ser utilizadas todas ou parte das características apresentadas anteriormente, dependendo das características que sejam consideradas mais relevantes para o sistema em questão.

Conforme exemplificado na Figura 4.1, deve ser atribuído, a cada métrica, um valor entre 0 e 1 (a importância da métrica é tanto maior quanto mais próximo de 1 for o valor atribuído). Depois disto, os valores atribuídos são normalizados e recalculados para que a soma de todos os valores definidos totalize 1. De seguida, é calculada a pontuação do nó, segundo a seguinte fórmula:

$$Ns = \sum_{i=1}^n M_i \times W_i \quad (4.1)$$

Na equação 4.1, M representa uma métrica do nó e W o peso atribuído a essa métrica. As métricas a considerar no cálculo da pontuação de um nó, e os respetivos pesos, podem ser definidos num ficheiro de configuração, ilustrado na Figura 4.1. Estes pesos serão tomados em consideração quando um nó é adicionado ao grafo.

```
"metrics_weight":[
  {
    "metric": "cpu_logical_cores",
    "weight": 0.4
  },
  {
    "metric": "cpu_physical_cores",
    "weight": 0.4
  },
  {
    "metric": "physical_memory",
    "weight": 0.5
  },
  {
    "metric": "memory_total",
    "weight": 0.7
  },
],
```

Figura 4.1 – Definição da relevância (pesos) de cada característica de um nó

4.2 Modelos de grafo de propagação de dados de um evento

Quando ocorre um determinado evento, e existe necessidade de propagar mensagens de aviso ou alerta, é necessário identificar quem vai enviar essa informação, a forma como vai ser feita essa partilha e os interessados na sua receção.

Para isso é criado, para cada evento, um grafo de propagação dos dados, que represente e defina a forma como a informação vai ser propagada e de que fonte vai partir. Estes grafos permanecem ativos enquanto perdurar o evento a que dizem respeito, sendo removidos quando este terminar.

Consideram-se nós a adicionar ao grafo todos aqueles que cumpram um determinado critério de interesse (por exemplo, estar num raio de X km ou a uma distância máxima de Y metros do local do evento). Desta forma, os nós presentes no grupo de interesse podem não coincidir com os nós definidos nos grupos hierárquicos anteriormente apresentados.

A estrutura do grafo pode tomar diversas formas, considerando diferentes modos de propagação da mensagem na rede, nomeadamente:

- Modelo 1 – apenas um nó (o melhor dos nós envolvidos na colisão) é responsável pelo registo e o envio da mensagem diretamente para cada um dos outros nós. O envio é feito de forma direta (nó de origem para o nó de destino). Envio de um nó para todos os restantes ($1 \rightarrow N$), como ilustrado na Figura 4.2.

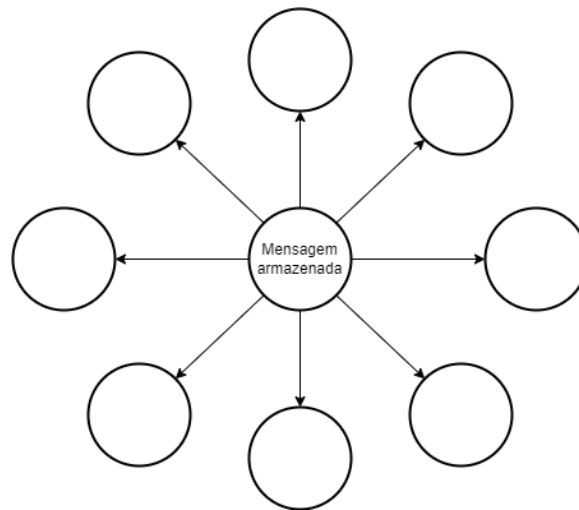


Figura 4.2 - Organização dos nós no Modelo 1

- Modelo 2 – a mensagem do evento é registada por um nó (o melhor dos nós envolvidos na colisão). Este nó é responsável pelo envio da mensagem para todos os restantes interessados, mas o envio poderá ser feito por retransmissão, tomando em consideração o caminho definido pelas ligações criadas segundo a lógica apresentada na secção 4.4.1 – como ilustrado na Figura 4.3. Envio de um nó para todos os restantes ($1 \rightarrow N$), mas que pode conter retransmissão, isto é, a mensagem poderá ser encaminhada por nós intermédios entre o emissor e o recetor. O emissor, para enviar a mensagem para cada nó de destino, vai instruindo os nós intermédios que se encontram no percurso para que retransmitam a mensagem até ao recetor.

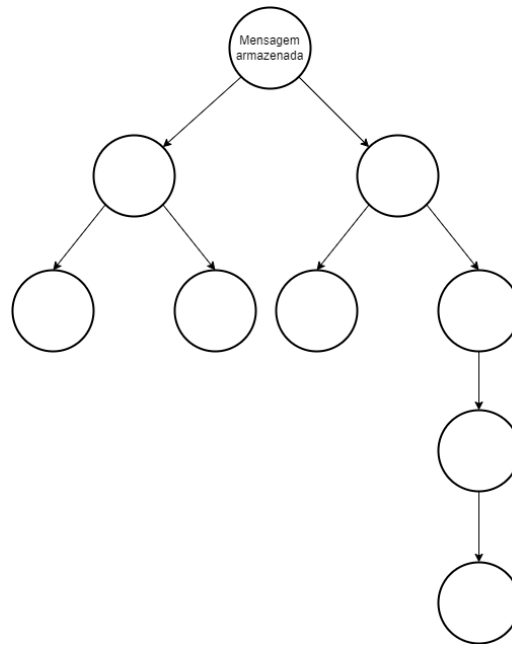


Figura 4.3 - Organização dos nós no Modelo 2

- Modelo 3 – a mensagem do evento é registada em vários nós (o melhor dos nós envolvidos na colisão + N melhores nós). Ou seja, antes de ser construído o grafo de propagação de dados (exemplificado na Figura 4.4), o melhor dos nós envolvidos na colisão, que armazena a informação, transfere-a para as restantes N réplicas. Estes nós, que ficam também com a informação armazenada, adquirem uma nova propriedade que os distingue dos restantes, ficando também com a capacidade de autonomamente enviar a mensagem para os restantes. Posteriormente, são construídos subgrafos de propagação dos dados, com base no número de nós que têm a informação armazenada e que, por isso, são potenciais emissores para os restantes. O envio da mensagem é feito pelo nó com a mensagem armazenada ao qual cada nó está ligado ou ao qual cada nó tem caminho. O envio é efetuado por vários nós ($M \rightarrow N$). À semelhança do Modelo 2, pode existir retransmissão ou não. Este modelo, ao contrário dos anteriores, é mais tolerante a falhas, uma vez que, caso um dos nós que armazena a informação falhe, a informação encontra-se armazenada noutros nós. Este modelo foi implementado e testado com três variantes: duas réplicas, quatro réplicas e oito réplicas, designados daqui em diante por Modelo 32, Modelo 34 e Modelo 38, respetivamente.

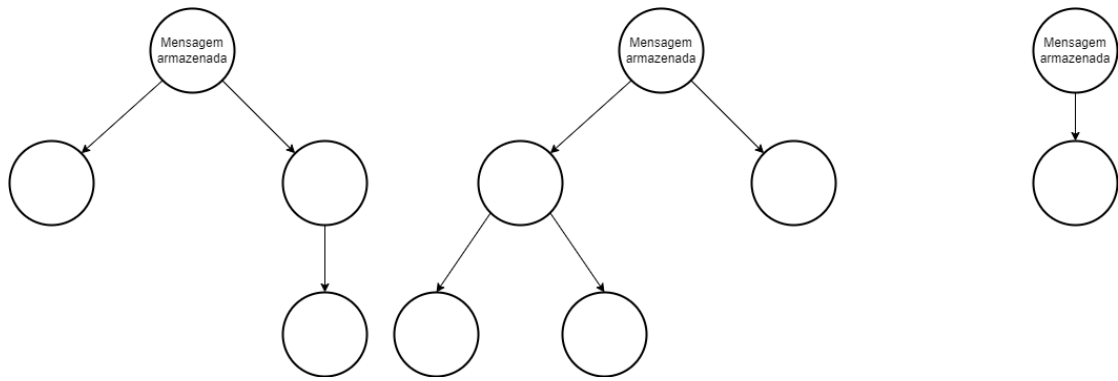


Figura 4.4 - Organização dos nós no Modelo 3 (com 2 réplicas – Modelo 32)

Para além dos três modelos (e as variantes) descritos anteriormente, foram ainda considerados outros modelos potencialmente interessantes, mas que, devido às limitações temporais, não foram implementados.

- Modelo 4 – a mensagem do evento é armazenada em todos os nós incluídos no grafo de propagação, no momento do evento. Posteriormente, a mensagem é enviada para os novos nós interessados (que entram no grafo nos instantes seguintes à ocorrência do evento) através do nó mais próximo de cada um. Este modelo, em princípio, propiciaria a existência de mais envios diretos e de menos por retransmissão, tornando o envio potencialmente mais rápido. No entanto, no momento de construção do grafo de propagação dos dados, teria custos maiores no que toca a tempo de armazenamento da mensagem nos nós.
- Modelo 5 – exclui todos os nós que não possuam condições mínimas para armazenar a informação, mediante algumas características como por exemplo a ocupação da memória ou a percentagem de utilização de CPU. Os nós que reúnem condições armazenam a informação, e os restantes recebem-na do nó mais próximo que tenha a informação armazenada.

4.2.1 Ligações no grafo de propagação de dados

A construção do grafo de propagação de dados, correspondente a determinado evento, parte sempre do nó, ou de um dos nós, que tem a informação armazenada.

Para cada nó, excetuando no Modelo 1, em que é estabelecida uma ligação direta entre o nó que armazena a informação e cada um dos restantes nós, é necessário determinar:

- Modelo 2 – qual o melhor caminho para o nó que armazena a informação;
- Modelo 3 – qual o melhor caminho para o melhor dos nós que armazena a informação.

Para esta decisão, foi criada uma função para determinar a melhor ligação. Podem ser consideradas diversas métricas, como por exemplo a distância entre os nós, a latência ou a largura de banda. Neste caso, a implementação e os testes foram realizados considerando a distância entre os veículos. Ou seja, em cada um dos modelos, cada nó liga-se da seguinte forma:

- Modelo 2 – cada novo nó liga-se ao nó mais próximo de si. No entanto, caso a distância do caminho do nó mais próximo ao nó com a informação armazenada seja superior à distância direta do novo nó ao nó com informação armazenada, então o novo nó liga-se diretamente ao nó com informação armazenada. A distância de um novo nó A até ao nó com a informação armazenada, passando por nós intermédios, é dada por:

$$d(A, X_1) + \sum_{i=1}^{n-1} d(X_i, X_{i+1}) + d(X_n, E) \quad (4.2)$$

Na equação 4.2, $d(A, X_1)$ é a distância entre o novo nó A e o primeiro nó intermédio X_1 no melhor caminho X_1, X_2, \dots, X_n, E , em que $\sum_{i=1}^{n-1} d(X_i, X_{i+1})$ é a soma das distâncias entre os nós intermédios X_i consecutivos ao longo do caminho onde n representa o número de nós intermédios do caminho, e $d(X_n, E)$ é a distância entre o último nó intermédio X_n e o nó com a informação armazenada E . Caso a distância obtida pela fórmula anterior for superior à distância direta entre o novo nó A e o nó com a informação armazenada E , então o novo nó estabelece ligação diretamente com E – conforme exemplificado na Figura 4.5.

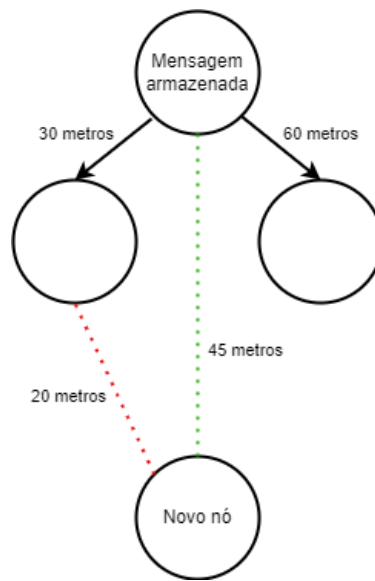


Figura 4.5 - Método de ligação de um novo nó ao grafo no Modelo 2

- Modelo 3 - cada novo nó liga-se ao nó mais próximo de si. No entanto, caso a distância do nó mais próximo ao nó com a informação armazenada à qual está ligado (direta ou indiretamente) seja superior à distância do novo nó ao nó mais próximo com informação armazenada, então o novo nó liga-se diretamente ao nó mais próximo com informação armazenada (Figura 4.6). Segue a mesma metodologia do modelo anterior, mas em vez de analisar apenas uma fonte de dados, toma em consideração os subgrafos de várias réplicas.

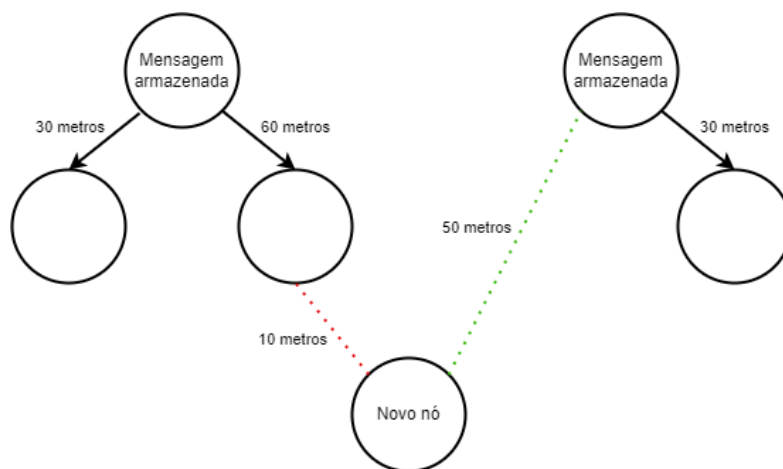


Figura 4.6 - Método de ligação de um novo nó ao grafo no Modelo 3

4.2.2 Atualização do grafo de propagação dos dados

Dada a elevada mobilidade dos elementos que constituem o sistema (veículos), o que pode levar a mudanças de localização e conseqüente entrada e saída da zona de interesse, é necessário considerar a reconstrução das ligações do grafo. Se um veículo que abandonou for um nó folha (nó final do grafo, não sendo “nó de passagem” no grafo para outro nó), então não há necessidade de reconstruir o grafo. No entanto, caso seja um nó que estabelece ligação com outros, é necessário refazer o grafo ou parte dele. Foram consideradas duas possibilidades:

- Reconstrução total – todo o grafo é reconstruído. Como desvantagem, esta abordagem leva a um maior custo computacional. No entanto, refaz todo o grafo e pode criar melhores caminhos do que os que tinham sido anteriormente criados porque os veículos podem ter-se movimentado.
- Reconstrução parcial – apenas as ligações afetadas pela saída do nó em questão são reconstruídas. Esta abordagem exige menores custos computacionais, no entanto pode fazer com que o grafo mantenha ligações desatualizadas – por exemplo no caso em que existe movimentação de veículos e as distâncias são alteradas.

Existe ainda a possibilidade de o veículo que sai da zona de interesse ser um dos que armazena a informação. Neste caso, ainda antes da reconstrução do grafo, é necessário transferir a informação desse veículo para o veículo com melhor pontuação do grafo e que ainda não contenha a informação armazenada.

Os veículos que saem da zona de interesse são removidos do grafo e, caso voltem a entrar, são novamente registados.

4.3 Protótipo implementado

No protótipo implementado foram utilizados dois servidores Neo4j: um que guarda o grafo global, e outro que guarda um grafo para cada evento. As várias funcionalidades do protótipo, obtenção dos dados dos nós, registo dos nós, interação com os servidores de grafos, comunicação entre nós e integração com o simulador de tráfego foram desenvolvidas em Python.

Foi desenvolvido um mecanismo de criação e registo de nós – cujo processo se encontra representado na Figura 4.7. Aquando da inicialização de um nó, é consultado um ficheiro de configuração, onde são definidas as características que serão registadas em cada nó, bem como o peso a atribuir a cada uma para o cálculo da pontuação do nó – descrito em 4.1. Estas características são obtidas de várias bibliotecas Python, nomeadamente *Osquery* (que permite obter múltiplas informações acerca de um dispositivo) e *shutil*. O programa Python inicializa o nó, adiciona-o ao grafo global com as respetivas características e estabelece ligação com os nós já existentes no grafo (caso não seja o primeiro) conforme apresentado na

secção 3.2. Periodicamente, as características de cada nó vão sendo atualizadas, bem como a respetiva pontuação.

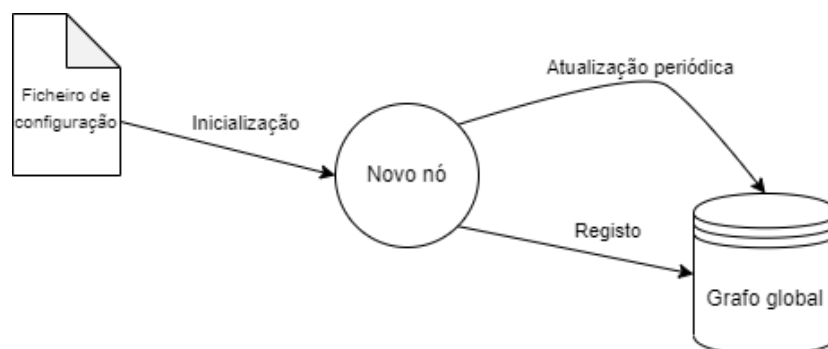


Figura 4.7 - Mecanismo de registo dos nós

Quando ocorre um evento, é criado um novo grafo de propagação de mensagens desse evento. A esse grafo são adicionados os nós do grafo global que cumpram o que está descrito na secção 4.2 – neste caso o critério considerado foi a distância em relação ao local do evento. Os nós adicionados a este novo grafo são novos nós, que contêm como propriedades apenas a *id*, o *ip*, o *hostname* e a pontuação do nó do grafo global a que correspondem. Depois de adicionados ao grafo de propagação de dados do evento, estes nós estabelecem ligação com os restantes dependendo do modelo considerado, de entre os modelos apresentados também na secção 4.2. O nó com melhor pontuação de entre os dois envolvidos no evento (por exemplo: uma colisão) é o responsável pelo envio da mensagem. Caso não seja emissor único, então esse nó é responsável conjuntamente com mais N réplicas. O processo de envio da mensagem apresenta-se de seguida, na secção 4.3.1, bem como o mecanismo de minimização de envio de mensagens repetidas, na secção 4.3.2.

4.3.1 Processo de envio de mensagem

O processo de envio das mensagens foi implementado através da utilização de *threads*. Quando um nó precisa de enviar uma mensagem para vários nós, este envio é efetuado através da execução de uma thread para comunicar com cada nó de destino. Assim, quando existe mais que um veículo interessado em receber uma mensagem, é enviada simultaneamente uma mensagem para cada veículo. Desta forma, não é necessário esperar pelo término do envio a decorrer para iniciar o envio seguinte. Para além disso, e para garantir a entrega da mensagem, foi implementado um mecanismo de resposta por parte do nó recetor, que envia uma confirmação de receção ao nó emissor.

Cada *thread* utiliza *sockets*, fazendo ligação do endereço IP de origem ao endereço IP do próximo nó (pode ser o nó de destino, no caso de envio direto, ou um nó intermediário, no caso do envio por retransmissão).

No caso do envio direto, o nó de origem apenas carrega a mensagem e envia-a para o nó de destino. Quando o nó de destino recebe a mensagem, envia um *acknowledge* (sinal utilizado na transmissão de dados) de volta para o nó de origem para assinalar a sua receção.

Já no envio por retransmissão, o nó de origem carrega a mensagem e envia-a para o próximo nó intermediário. Para além da própria notificação (a entregar ao nó de destino) esta mensagem possui uma lista de identificação dos nós seguintes, do caminho até ao nó de destino. Ou seja, o nó seguinte, ao receber a mensagem, lê a identificação do primeiro nó da lista, remove-o da lista, e envia a mensagem para esse nó, com a lista atualizada. Este processo é repetido iterativamente até que seja alcançado o nó de destino. Ao chegar ao nó de destino, este procede ao envio do *acknowledge*. À semelhança do que acontecia no envio direto, o nó de destino envia o *acknowledge* para o nó anterior, e este nó para o nó que o antecede, e assim sucessivamente, fazendo o percurso inverso ao envio da mensagem, até atingir o nó de origem da mensagem – conforme representado na Figura 4.8, onde o caminho a verde representa o envio da mensagem, o caminho a azul o envio do *acknowledge*, e os números as etapas do processo, por ordem crescente.

Este processo de receção do *acknowledge* é também útil para a verificação de funcionamento dos nós. Caso um nó que armazena a mensagem deixe de receber este sinal, é porque o seu funcionamento não está de acordo com aquilo que seria previsto. Assim, todos os nós dependentes do mesmo para a receção da informação (nos casos em que existe mais que um emissor) estabelecem ligação com outro nó que tenha a informação armazenada, não deixando por isso de receber a informação. Nos Modelos 1 e 2, onde existe um único nó que armazena a informação, a indisponibilidade desse nó faz com que todos os restantes deixem de receber a informação.

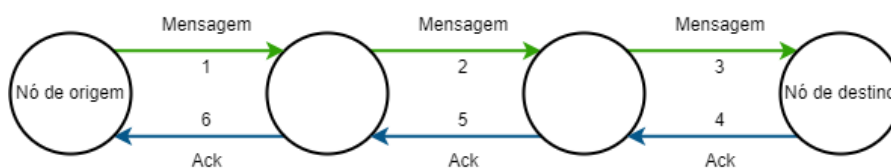


Figura 4.8 – Processo de envio da mensagem e do *acknowledge* no envio por retransmissão

4.3.2 Minimização do envio de mensagens repetidas

Em qualquer dos modelos de construção de grafo apresentados, apenas os nós com a mensagem armazenada – identificados por uma propriedade – são responsáveis pelo envio da mensagem. Esta propriedade permite organizar os nós de um grafo, entre os nós que atuam como responsáveis pelo envio da mensagem e os nós que atuam como retransmissores mas não como fonte de origem da mensagem.

Um nó, quando recebe uma mensagem, regista localmente o identificador da mensagem e *timestamp* da mensagem recebida. Desta forma, os nós elegíveis para a receção de uma mensagem são os nós do grafo que não possuam registo de receção dessa mensagem, representando os novos nós que foram recentemente adicionados ao grafo e/ou os nós que por qualquer motivo ainda não receberam a mensagem.

Existe um *timeout* t associado a cada envio. Ou seja, o nó de origem espera t segundos pela resposta, considerando que o envio falhou caso não receba resposta nesse intervalo de tempo. Neste caso, então esse nó será elegível para receber a mensagem no próximo envio, até que a mensagem seja recebida.

Quando a capacidade de guardar mensagens é limitada, é utilizado um algoritmo LRU para libertar espaço, onde as mensagens mais antigas são removidas. Adicionalmente poderá ser associado, a cada mensagem, um *deadline* temporal, que indica o intervalo temporal em que esta mensagem é relevante, sendo removida depois disso.

5 AVALIAÇÃO EXPERIMENTAL

Neste capítulo, é apresentado o ambiente de avaliação utilizado, bem como o *setup* experimental que foi construído para a realização dos testes. São também apresentadas as métricas de teste recolhidas e analisados os resultados obtidos.

5.1 Ambiente de Avaliação

Num ambiente real, cada veículo seria um elemento autónomo, com capacidade de processamento e transmissão, que seria registado como um nó no grafo. Na avaliação experimental, foram utilizadas quatro máquinas virtuais para simular vários veículos. Em cada simulação, é simulado o mesmo número de veículos em cada máquina virtual, por exemplo, numa simulação envolvendo dezasseis veículos, cada máquina virtual será responsável por simular quatro nós, que representam quatro veículos.

Para avaliação experimental, foram disponibilizados os recursos computacionais apresentados na Tabela 5.1.

Tabela 5.1 - Características dos recursos utilizados

	CPU(s)	RAM (GB)	Armazenamento (GB)	Função
VM1	1	2.1	40	Servidor dos grafos
VM2	2	2.1	40	Simulador de veículos (nós)
VM3	2	1	20	Simulador de veículos (nós)
VM4	2	1	15	Simulador de veículos (nós)
VM5	2	1	25	Simulador de veículos (nós)
PC	12	8	930	Simulador SUMO

Além das máquinas virtuais utilizadas para simular veículos, utilizou-se uma máquina virtual que executa o Neo4j, um servidor de bases de dados em grafos, e um PC que executa o simulador de tráfego SUMO.

5.2 Setup Experimental

De modo a avaliar a solução proposta num ambiente veicular, e analisar a forma como as mensagens são trocadas entre os veículos, surge a necessidade de fazer variar a localização de cada nó, que representa um veículo, e eventos que possam ocorrer, por exemplo acidentes. Para esse efeito, foi utilizado o simulador de tráfego SUMO – apresentado na revisão da literatura.

Dada a impossibilidade de cobrir todas as possibilidades de eventos ou ocorrências que possam afetar o normal funcionamento e circulação dos veículos, nestes testes serão considerados apenas acidentes rodoviários. Apesar de, na realidade, cada acidente poder envolver mais que dois veículos, ou até mesmo só um, os acidentes simulados envolvem apenas dois veículos (o *collider* e a *victim*). Assim, toda a lógica de propagação e troca de mensagens será desenvolvida tomando por base um acidente entre dois e apenas dois veículos.

Quanto à reação dos veículos envolvidos, este simulador permite optar por um de quatro comportamentos:

- *None* – A colisão mantém-se no ponto onde aconteceu até ao fim da simulação. Pode bloquear a passagem de outros veículos, parando o trânsito e fazendo com que não seja possível lidar com a entrada e saída de veículos no grupo de veículos interessados em receber a mensagem;
- *Remove* – No passo seguinte à colisão, os dois veículos envolvidos são removidos e os restantes veículos comportam-se normalmente;
- *Teleport* – Logo após a colisão, um dos veículos envolvidos é teletransportado para a aresta seguinte do trajeto, prosseguindo ambos o seu trajeto e não causando qualquer transtorno ao trânsito;
- *Warn* – No momento da colisão, é emitido um *warning*.

De entre os comportamentos disponíveis no simulador, foi utilizado o comportamento *Remove*. No entanto, como num contexto real este comportamento é irrealista pois os veículos não são imediatamente removidos e também porque assim não causa transtorno à circulação dos restantes, foi implementada uma solução em que a remoção dos veículos envolvidos no evento não é feita imediatamente, mas sim apenas alguns momentos depois.

A movimentação realizada pelos veículos na simulação é feita segundo o IDM (*Intelligent Driver Model*), um modelo de simulação de tráfego capaz de reproduzir comportamentos realistas de tráfego, nomeadamente mudanças de faixa, aceleração em curvas ou redução de velocidade em cruzamentos.

5.3 Importação do mapa de estradas e definição das rotas

O simulador utilizado, SUMO, permite a importação de ficheiros extraídos do *OpenStreetMap*. Assim, as simulações realizadas foram aplicadas a uma zona. Para isso, foi extraído e importado no simulador um ficheiro contendo uma zona da cidade de Coimbra, conforme apresentado na Figura 5.1.



Figura 5.1 - Rua extraída do OpenStreetMap

Depois de importado o ficheiro, procedeu-se à definição das rotas, isto é, os trajetos que os veículos presentes na simulação vão seguir. Foram definidas quatro rotas distintas, pelas quais foram distribuídos os veículos considerados em cada simulação. Aos veículos foram atribuídas várias características, incluindo a aceleração ou a velocidade máxima.

5.4 Simulações e métricas recolhidas

Com o objetivo de testar alguns dos modelos abordados na secção 4.2, foram definidos quatro setups com diferentes conjuntos de veículos.

Foi criado um ficheiro de configuração de cada simulação - onde é definido o mapa de estradas, os veículos e respetivas velocidades, entre outros parâmetros – garante que as ocorrências dessa simulação aconteçam exatamente no mesmo instante e entre os mesmos veículos em qualquer execução dessa mesma simulação.

O número total de veículos presente no grafo global pode diferir do número de veículos presentes no grafo de propagação de dados, nos casos em nem todos entram na zona de interesse de receção da mensagem, e conseqüentemente não recebem a mensagem. Esta diferença acontece também porque, mesmo no contexto real, existe necessariamente saída de veículos de uma determinada área para que outros possam entrar, ou seja, com o aumento do número de veículos total passa a

ser difícil colocá-los simultaneamente num mesmo espaço. O número de veículos considerados em cada setup apresenta-se na Tabela 5.2.

Tabela 5.2 - Número de veículos considerados em cada setup

Setup	Número de veículos no grafo global	Número de veículos no grafo de propagação do evento
1	16	16
2	32	31
3	64	46
4	100	64

Em cada teste realizado serão consideradas e recolhidas várias métricas, para análise e comparação, incluindo:

- Tempo de notificação (TN) – representa o tempo desde que é feito o envio da mensagem, desde o nó de origem, até que a mensagem é recebida, no nó de destino, incluindo o tempo das retransmissões efetuadas por nós intermédios;
- Tempo de registo (TR) – representa o tempo necessário para a adição de um nó ao grafo. Inclui também o tempo de análise da topologia do grafo para perceber que ligações (arestas) devem ser criadas entre o novo nó e os nós existentes no grafo de propagação do evento;
- Tempo de armazenamento (TA) – representa o tempo necessário para o armazenamento da mensagem, nos nós em que há armazenamento. Na simulação foi considerado que a mensagem é armazenada num ficheiro;

Para cada *setup*, foi definida uma simulação para cada modelo. Para o Modelo 3, foram definidas três simulações distintas com diferentes números de réplicas: duas, quatro e oito, que na análise dos resultados são genericamente apresentados por 32, 34 e 38, respetivamente. As simulações definidas na avaliação experimental apresentam-se na Tabela 5.3.

Tabela 5.3 – Simulações consideradas

Simulação	Modelo	Tipo de envio	Número de réplicas	Setup	Número de veículos no grafo de propagação do evento
1	1	Direto	--	1	16
2	2	Direto e retransmissão	--	1	16
3	3	Direto e retransmissão	2	1	16
4	3	Direto e retransmissão	4	1	16
5	3	Direto e retransmissão	8	1	16
6	1	Direto	--	2	31
7	2	Direto e retransmissão	--	2	31
8	3	Direto e retransmissão	2	2	31
9	3	Direto e retransmissão	4	2	31
10	3	Direto e retransmissão	8	2	31
11	1	Direto	--	3	46
12	2	Direto e retransmissão	--	3	46
13	3	Direto e retransmissão	2	3	46
14	3	Direto e retransmissão	4	3	46
15	3	Direto e retransmissão	8	3	46
16	1	Direto	--	4	64
17	2	Direto e retransmissão	--	4	64
18	3	Direto e retransmissão	2	4	64
19	3	Direto e retransmissão	4	4	64
20	3	Direto e retransmissão	8	4	64

5.5 Impacto do envio simultâneo na taxa de insucesso

Considerando o Modelo 1, onde um nó é responsável pelo envio direto da mensagem para todos os restantes, foi realizado um teste para determinar o número de pedidos de mensagens simultâneos que um nó consegue suportar. Isto é, qual a quantidade máxima de *threads* um único nó é capaz de lidar com sucesso, sem falhas. Este teste foi realizado ao nó com melhor pontuação e ao nó com a pior pontuação para o envio das mensagens, a fim de avaliar o impacto da seleção do nó na capacidade de enviar vários pedidos simultâneos.

Para esse efeito, foi testado um grafo onde um nó é responsável pelo envio simultâneo de 100, 200, 300, 400 e 500 mensagens, distribuídas pelos restantes nós. Cada um dos envios corresponde a uma *thread*, e o objetivo é avaliar quantos foram bem sucedidas e quantas falharam.

Na Figura 5.2 apresentam-se os resultados quando o nó responsável pelo envio das mensagens é aquele que tem pior pontuação. É possível perceber que começa a apresentar falhas a partir de 400 *threads* e a percentagem de *threads* falhadas aumenta para números superiores de mensagens simultâneas.

Os resultados apresentados na Figura 5.3 (respeitantes ao nó com melhor pontuação), mostram que o nó responsável pelo envio das mensagens teve maior capacidade para lidar com 400 envios simultâneos, contrariamente ao pior nó que para este valor já apresentava falhas. Já para 500 envios, começam a existir *threads* que falham, não sendo o nó capaz de lidar com tantas solicitações. Para 600 envios, a taxa de insucesso é ainda mais elevada, ultrapassando os 25%. Ainda assim, as taxas de insucesso são sempre mais baixas que as do pior nó.

Impacto do envio simultâneo na taxa de insucesso - Nó com pior pontuação

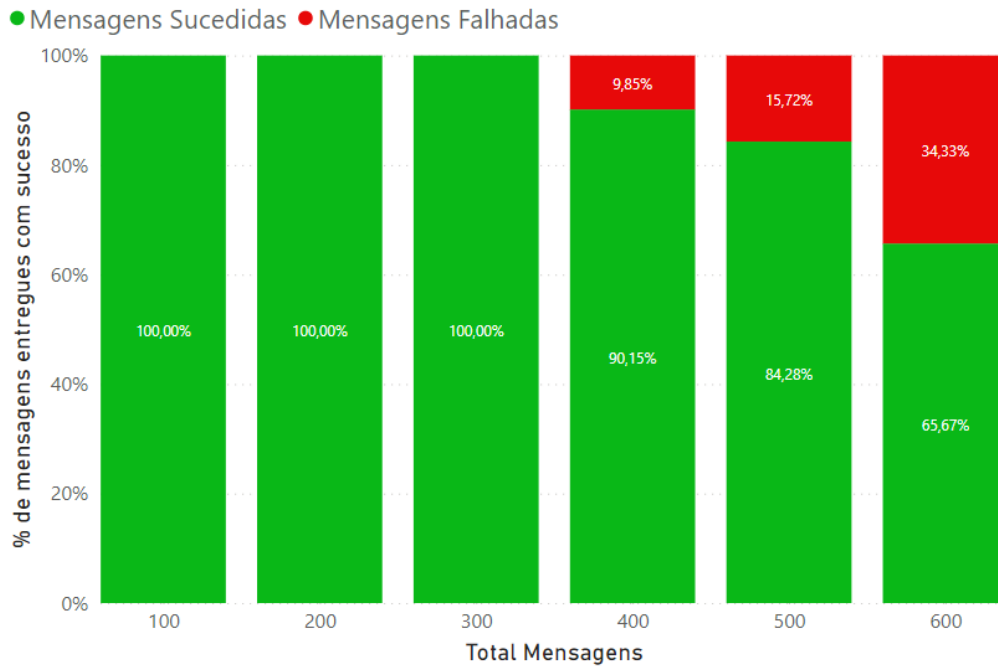


Figura 5.2 - Taxa de sucesso das mensagens para diferentes números de envios simultâneos no nó com pior pontuação

Impacto do envio simultâneo na taxa de insucesso - Nó com melhor pontuação

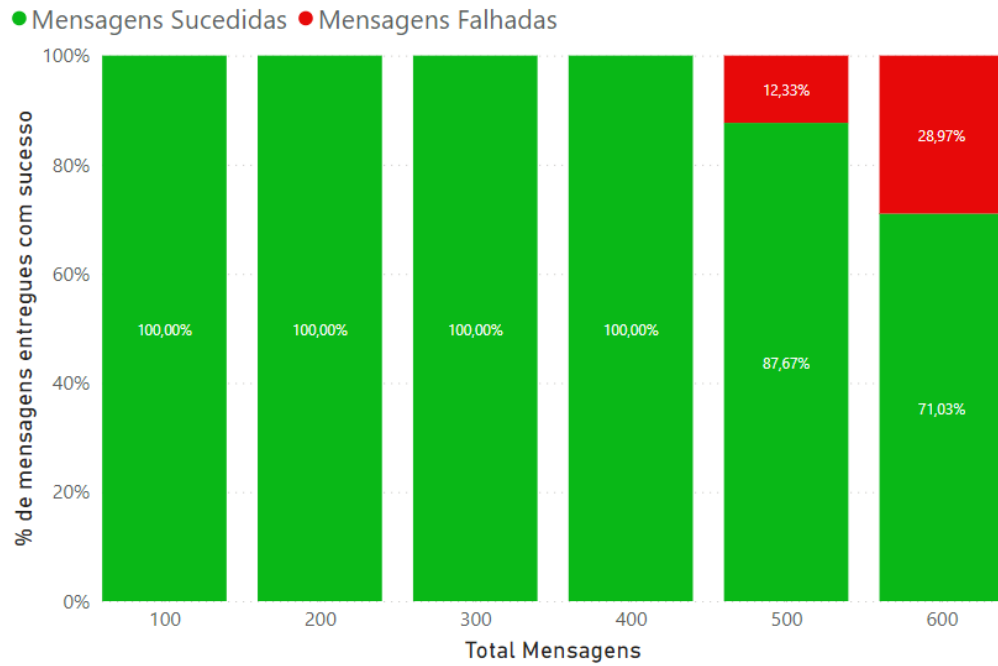


Figura 5.3 - Taxa de sucesso das mensagens para diferentes números de envios simultâneos no nó com melhor pontuação

O pico do gráfico em termos de memória, ou seja, o valor mais elevado, que corresponde ao momento em que o número de *threads* é também o mais elevado, é cerca de 20 Mb. Em média, o consumo de memória por cada *thread* é de aproximadamente 0,38 Mb. Conclui-se que não existe um grande impacto a nível de memória, e por isso podem ser utilizados equipamentos com poucos recursos a este nível.

5.7 Resultados das simulações

No total, foram executadas cem execuções das simulações: cinco para cada simulação definida, contemplando os cinco modelos existentes, e os quatro *setups* testados. Em cada simulação, existe apenas um evento – um acidente.

5.7.1 Avaliação do Tempo de Notificação (TN)

A Figura 5.5 apresenta a média do tempo de notificação (barras verticais) e o desvio padrão das várias execuções. Considerando os quatro *setups* descritos na secção 5.4, é possível perceber que este valor é mais elevado no Modelo 2 e apresenta os menores valores para o Modelo 1. Este resultado justifica-se pelo facto do Modelo 1 considerar o envio direto, enquanto no Modelo 2 a mensagem pode ter de passar por vários nós até chegar ao nó final, à semelhança do que acontece nos Modelos 3. No entanto, as simulações envolvendo o Modelo 3 contemplam replicação, o que significa que existem mais nós disponíveis para o envio da mensagem. Desta forma, conseguem construir-se caminhos de retransmissão da mensagem mais curtos, isto é, que exigem menos nós intermediários entre o nó transmissor e o recetor. Nas simulações consideradas, o TN diminui com o número de réplicas, particularmente em *setups* com um maior número de nós.

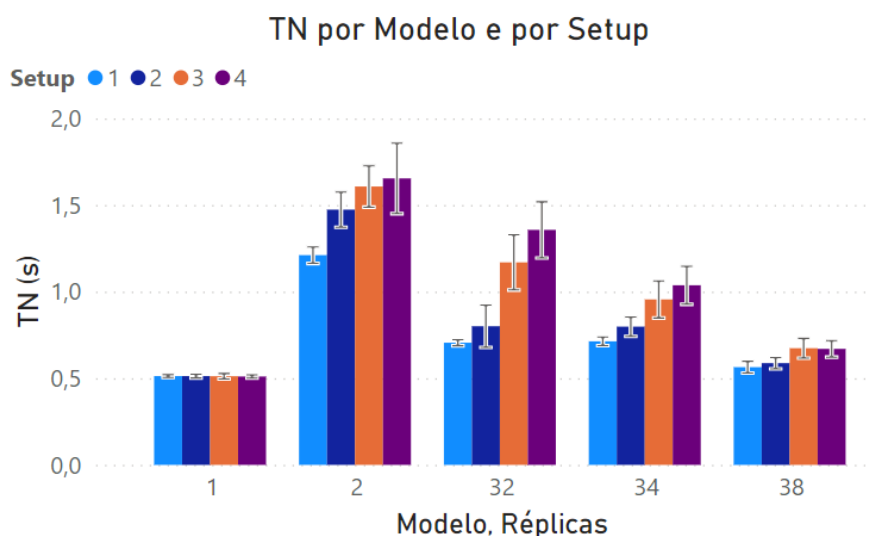


Figura 5.5 - TN por modelo e por setup

As barras de erro presentes em cada coluna da Figura 5.5 representam o desvio padrão dos valores obtidos. O desvio tem tendência a diminuir à medida que aumenta o número de réplicas, uma vez que os caminhos de retransmissão da mensagem tendem a ser menores e conseqüentemente os valores obtidos para TN são também mais próximos. Para além disso, o desvio padrão tem tendência a aumentar com o aumento do número de nós, porque com um maior número de nós a probabilidade de construção de caminhos maiores é também maior. Assim, os valores de TN têm tendência a dispersar mais da média quando o número de nós aumenta e quando existe um baixo número de réplicas.

Ainda na Figura 5.5, verifica-se que o TN para o Modelo 1 não varia com o número de veículos, ou seja, o número de mensagens enviadas. Já para os restantes modelos, esta métrica é afetada pelo número de veículos, uma vez que um maior número de veículos levará à formação de caminhos de transmissão com mais nós e fará com que aumente o TN.

Para o *setup* com o maior número de nós (*setup* 4), a Figura 5.6 a) apresenta o número médio de nós e o maior número de nós para o envio de uma mensagem para o nó mais distante, e em b) o tempo de notificação.

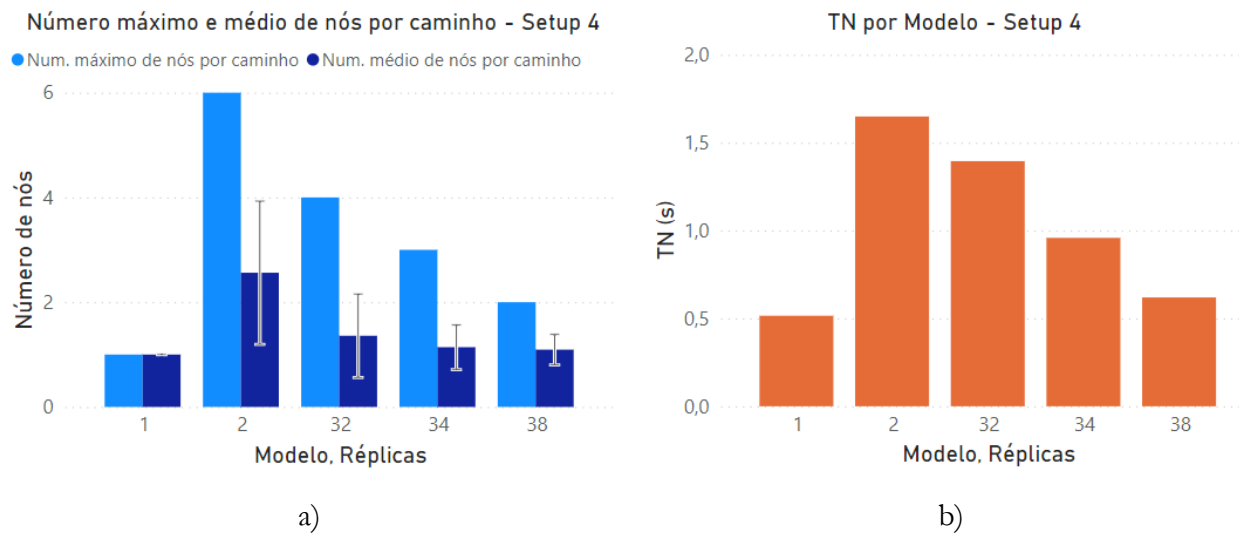


Figura 5.6 – Setup 4 a) Num. máximo e médio de nós por caminho por modelo; b) TN por modelo

É possível concluir que o Modelo 2, conforme expectável, leva à criação de maiores caminhos de retransmissão (6), e com a inserção de duas réplicas, existe redução para 4. A adição de mais réplicas leva, cada vez mais, à redução dos caminhos de retransmissão, sendo que com 8 réplicas o maior caminho já apresenta apenas duas retransmissões. É também visível o impacto do número de réplicas na diminuição do TN, bem como a variação do TN com o número de retransmissões (nós por caminho).

5.7.2 Avaliação do Tempo de Registo (TR)

A Figura 5.7 apresenta os valores do tempo de registo (TR), por modelo e *setup*.

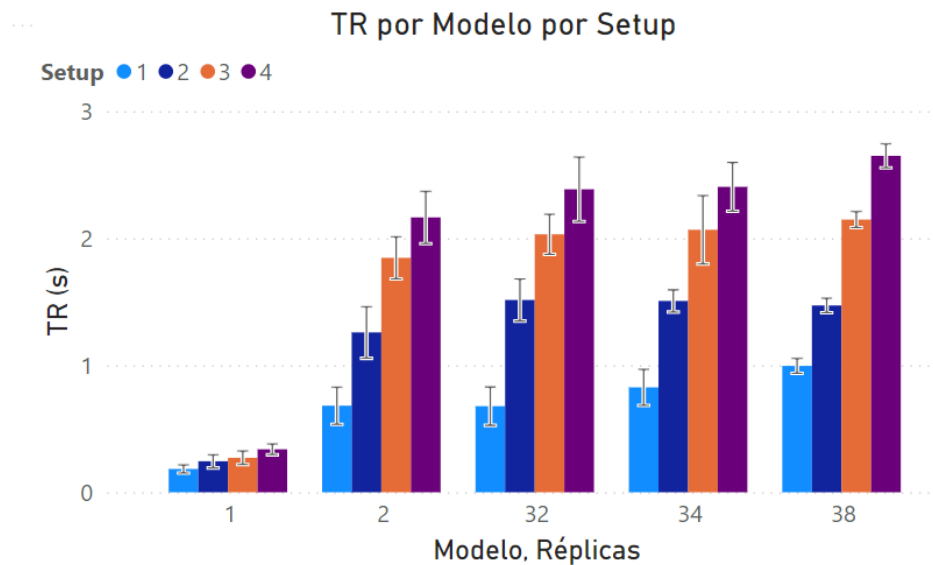


Figura 5.7 - TR por modelo e setup

Analisando esta métrica, verifica-se uma grande diferença entre os tempos de registo do Modelo 1 e dos demais. Isto porque, no Modelo 1, aquando do registo de um novo nó, apenas é necessário criar uma ligação ao nó que armazena a informação, não necessitando de nenhuma análise adicional ao grafo. Nos restantes, é necessário calcular as distâncias do nó aos restantes, de modo a perceber a que nó irá estabelecer uma ligação. No Modelo 2, há necessidade de analisar se a ligação deve ser estabelecida diretamente ao nó com a informação armazenada, ou a outro nó próximo que tenha ligação com o nó que armazena a informação. Já nos modelos com replicação, o aumento do TR prende-se com o facto de ser necessária ainda outra análise extra ao estado do grafo, nomeadamente qual dos nós que armazena a informação se encontra mais próximo em termos de distância, e se deve ser estabelecida ligação a esse nó ou a outro nó que embora não armazenando a informação, tenha ligação a um dos nós que a armazena.

Para além da análise entre modelos, pode também verificar-se um padrão entre *setups*, ou seja, com o aumento do número de nós. Esta métrica aumenta à medida que aumenta o número de nós no grafo, uma vez que para o registo de cada nó existe a necessidade de verificar distâncias para um maior número de nós para perceber qual o mais próximo.

Pese embora o custo associado à replicação, estes modelos têm a capacidade de reduzir o TN, conforme apresentado na secção 5.7.1. O aumento do número de nós com a informação armazenada (replicação) faz com que a informação fique mais

perto dos veículos com interesse, sendo por isso necessários caminhos mais curtos, muitas vezes até diretos, do emissor da informação até ao recetor.

5.7.3 Avaliação do Tempo de Armazenamento (TA)

O tempo de armazenamento (TA) é uma métrica cuja variação não está diretamente associada ao número de nós. O valor médio de TA, por cada armazenamento, mantém-se praticamente constante em todos os modelos e todos os *setups*. Nem o aumento do número de nós nem o aumento do número de réplicas leva ao aumento do valor por cada armazenamento.

Considerando o *setup* com o maior número de veículos (*setup* 4), a Figura 5.8 apresenta o total de mensagens armazenadas e o total de mensagens trocadas (contando com as retransmissões) por evento do emissor, para cada modelo.

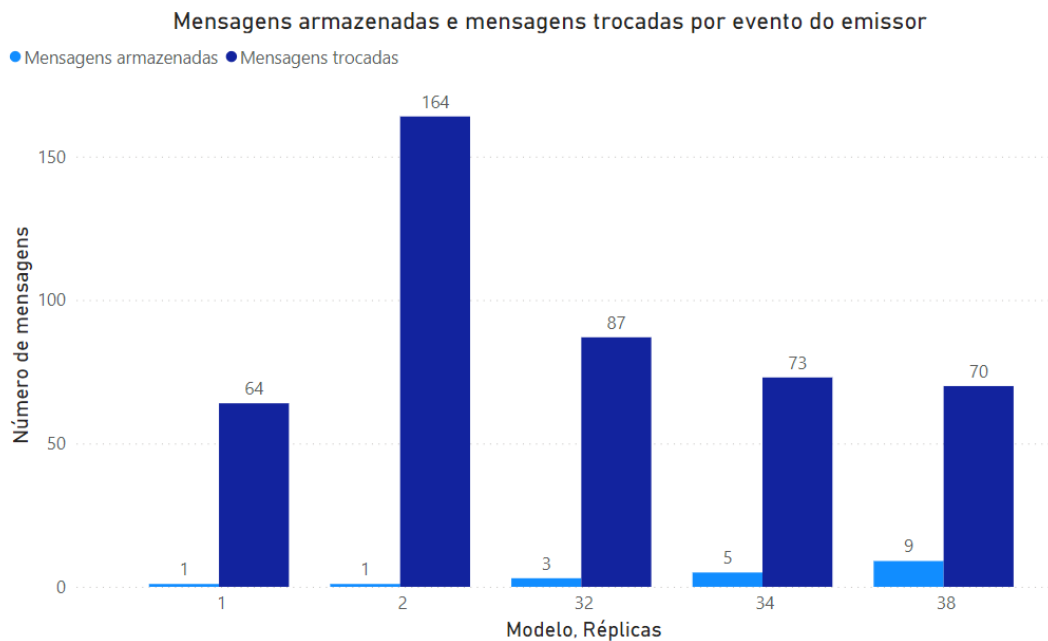


Figura 5.8 - Número total de mensagens armazenadas e número total de mensagens trocadas por modelo para o Setup 4

Conforme é possível visualizar, o aumento do número de réplicas, e consequentemente do número de mensagens armazenadas, leva à redução do número total de mensagens trocadas, uma vez que a mensagem está armazenada mais próxima de mais veículos o que leva à criação de caminhos de retransmissão menores. Ou seja, apesar de aumentar o TA total (tempo de armazenamento de todas as réplicas), o processo de transmissão da mensagem engloba menos nós por caminho, sendo criados caminhos mais curtos e que exigem a passagem por menos nós no processo de envio da mensagem. Optando por uma solução sem replicação e onde o envio não seja exclusivamente direto (Modelo 2), o TA total é mais reduzido mas o número total de mensagens trocadas é muito elevado. Tomando como base

o número de mensagens trocadas no Modelo 1 (que é igual ao número de veículos que receberam a mensagem), verifica-se que o Modelo 2 apresenta um aumento significativo – superior ao dobro – do número de mensagens. Este número vai sendo reduzido à medida que há replicação, atingindo no Modelo 3 com 8 réplicas um valor próximo do Modelo 1.

5.7.4 Síntese dos resultados

A Tabela 5.4 e a Tabela 5.5 apresentam, de forma resumida, os resultados de TN e TR, respetivamente, para cada setup e modelo testado, num esquema de cores onde o verde representa um melhor resultado, o amarelo um resultado razoável e vermelho um pior resultado.

Tabela 5.4 - Resumo dos resultados do tempo de notificação (TN)

Modelo	Número de réplicas	Tipo de envio	TN				Média Total por Modelo
			Setup 1	Setup 2	Setup 3	Setup 4	
1	0	Direto	0,51	0,51	0,51	0,51	0,51
2	0	Direto e retransmissão	1,21	1,49	1,61	1,65	1,49
3	2		0,71	0,8	1,17	1,35	1,01
3	4		0,72	0,8	0,97	1,04	0,88
3	8		0,56	0,59	0,65	0,67	0,62
Média Total por Setup			0,74	0,84	0,98	1,03	

O Modelo 1 é aquele que apresenta os melhores resultados para qualquer setup. No entanto, para além de ser necessário que o veículo que armazena a mensagem consiga estabelecer ligação com todos os restantes, este modelo acaba por ser limitativo no número de mensagens que é possível enviar simultaneamente, uma vez que relega essa responsabilidade a apenas um nó. Caso o número de mensagens a enviar seja elevado, o nó pode não ter capacidade para lidar com todos os pedidos em simultâneo – conforme demonstrado na secção 5.5. Para além disso, apenas um veículo armazena a mensagem, o que acaba por ser uma limitação na medida em que caso esse veículo falhe, nenhum outro é capaz de enviar as mensagens.

O Modelo 2, e conforme é possível verificar rapidamente pelo esquema de cores, é aquele que apresenta piores resultados no que toca ao TN. Para além de apresentar a mesma limitação já apontada ao Modelo 1 – apenas um veículo armazenar a mensagem – é o modelo que leva à construção de caminhos de

retransmissão mais longos, o que acaba por se refletir nos valores obtidos para TN, onde apresenta valores superiores a qualquer outro modelo.

As variantes do Modelo 3 testadas (duas, quatro e oito réplicas) têm a capacidade de reduzir o TN à medida que aumenta o número de réplicas. Inclusive, para oito réplicas, foi possível obter valores próximos aos do Modelo 1.

Analisando a média total por setup, é possível verificar o aumento de TN com o aumento do número de veículos. Já no que toca à média total por modelo, o Modelo 2 é o que apresenta maior TN e o Modelo 1 o menor. O aumento do número de réplicas leva à diminuição de TN.

Tabela 5.5 - Resumo dos resultados do tempo de registo (TR)

Modelo	Número de réplicas	Tipo de envio	TR				Média Total por Modelo
			Setup 1	Setup 2	Setup 3	Setup 4	
1	0	Direto	0,19	0,24	0,27	0,34	0,26
2	0	Direto e retransmissão	0,68	1,27	1,85	2,17	1,49
3	2		0,69	1,51	2,03	2,39	1,66
3	4		0,83	1,52	2,07	2,41	1,71
3	8		1,02	1,47	2,15	2,61	1,81
Média Total por Setup			0,68	1,2	1,67	1,98	

Apesar das limitações já apontada ao Modelo 1, é o modelo que apresenta melhores resultados também para o tempo de registo (TR).

Se, na tabela anterior, podia verificar-se uma diminuição do TN com o aumento do número de réplicas, a Tabela 5.5 mostra que, por outro lado, o aumento do número de réplicas leva também ao aumento do TR. Para além disso, as réplicas têm também outro custo associado: o aumento de TA total (tempo de todos os armazenamentos, incluindo réplicas).

Os modelos que apresentam piores resultados para TR são aqueles que contam com replicação, sendo os resultados tanto piores quanto maior o número de réplicas.

Analisando a média total por setup, é possível concluir que, tal como o TN, também o TR aumenta com o aumento do número de veículos. O aumento do número de réplicas leva à diminuição de TN, mas ao aumento de TR.

A Figura 5.9 apresenta o valor médio de TN e TR e TA total (tempo de armazenamento de todas as réplicas) para todos os *setups* apresentados, por cada modelo e número de réplicas testados. O TA total corresponde ao tamanho do ponto associado a cada modelo.

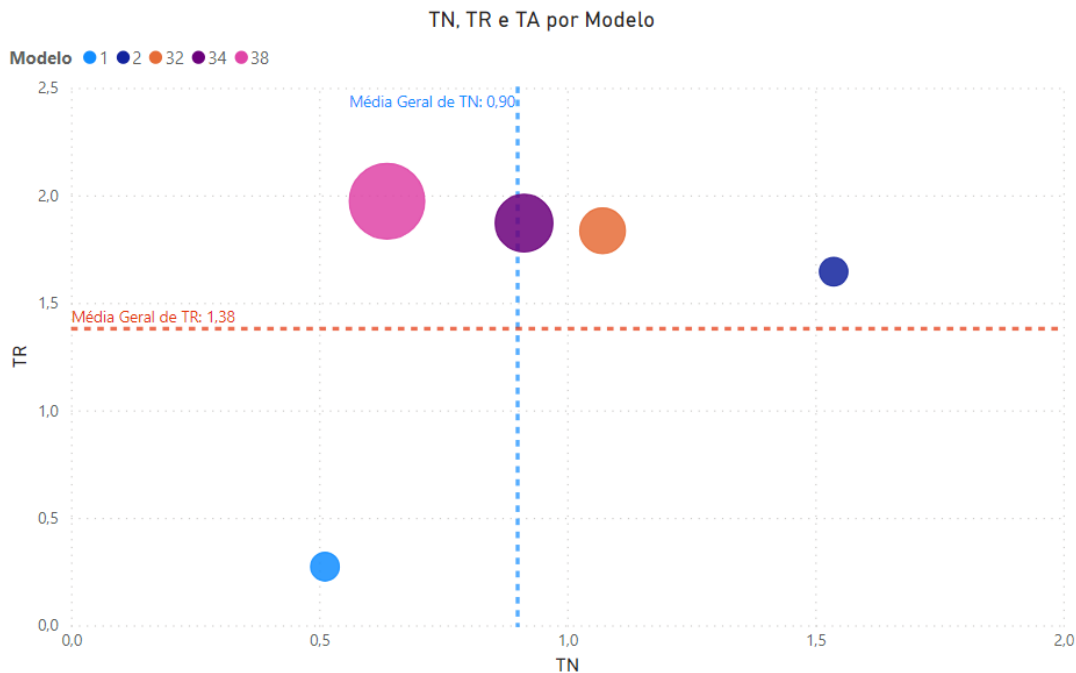


Figura 5.9 – Relação entre TN, TR e TA

Para contextos onde o número de mensagens simultâneas não seja muito elevado (depende das características do nó, conforme os resultados apresentados em 5.5), e onde exista comunicação do nó que armazena a mensagem diretamente com todos os restantes, o Modelo 1 será o mais favorável, já que apresenta o menor valor de TN, TR. Também o TA total é muito reduzido neste modelo, uma vez que existe armazenamento em apenas um nó. Num contexto com restrições associadas ao TN, por exemplo TN máximo de 0,6, apenas este modelo poderia ser utilizado.

O Modelo 2 poderia ser aplicado num contexto semelhante ao anterior, mas onde o nó que armazena a mensagem não conseguisse estabelecer ligação direta com todos os restantes. Assim, alguns dos envios teriam de ser feitos passando por nós intermediários, utilizando a retransmissão. Este modelo destaca-se dos demais pelo TN elevado que apresenta. Ainda assim, o TR é menor que o TR dos modelos com replicação. O TA total é praticamente igual ao do Modelo 1.

O Modelo 3 é o que apresenta maior TR comparativamente com o Modelo 1 e o 2, para qualquer número de réplicas, sendo que tem tendência a aumentar com

o número de réplicas. Também o TA total apresenta o mesmo comportamento. Pese embora o TR e o TA total superiores aos restantes modelos, é possível obter um TN bastante próximo do TN apresentado pelo Modelo 1, com oito réplicas. Para além disso, o Modelo 3 é o mais robusto para lidar com elevados números de mensagens simultâneas, uma vez que apresenta múltiplas fontes de dados levando à divisão da carga total por diferentes nós.

De forma a analisar mais pormenorizadamente os modelos que se encontram mais próximos em termos de TR, criou-se uma nova representação dos dados, onde o tamanho dos pontos representa o TR - Figura 5.10.

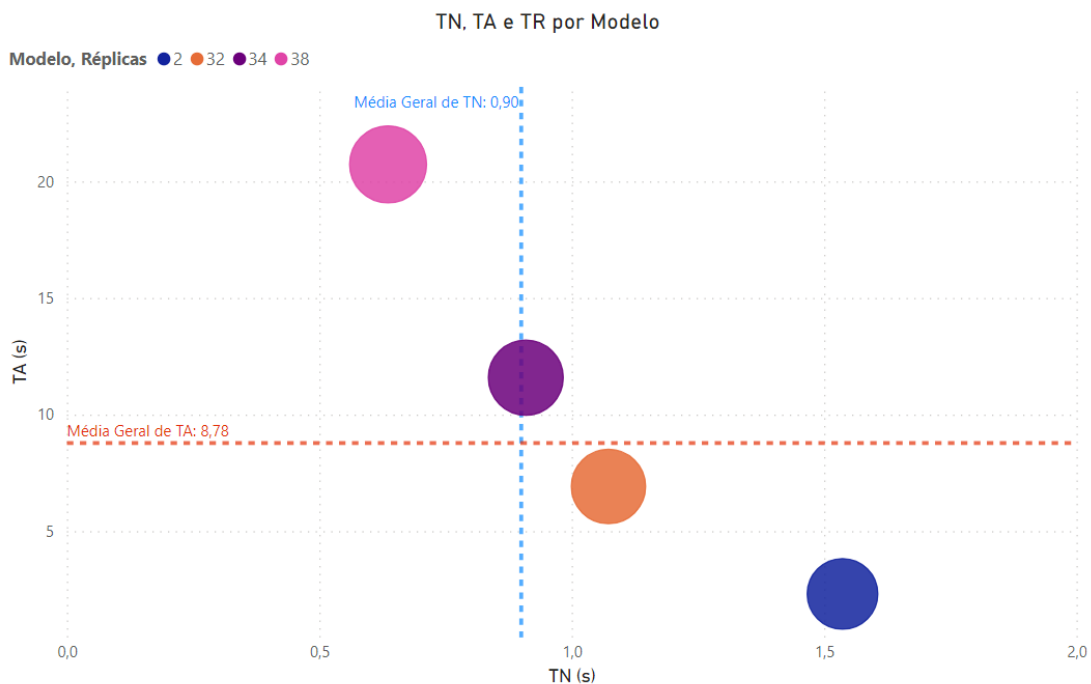


Figura 5.10 - Relação entre TN, TA e TR

Em relação à média geral de TA total, apenas o Modelo 2 e o Modelo 3 com 2 réplicas apresentam valores inferiores. Já no que toca ao TN, apenas o Modelo 3 com 8 réplicas se apresenta abaixo da média geral, estando o Modelo 3 com 4 réplicas no limite. O aumento do TR entre estes modelos não é muito significativo, quando comparado com o TA.

Quando as restrições a nível de TA forem reduzidas, por exemplo 8 segundos de tempo total de armazenamento, então deve ser utilizado o Modelo 3 com 2 réplicas ou então o Modelo 3. Um maior grau de replicação fará com que o TA total ultrapasse esse limite. A escolha destes modelos leva a tempos de TN superiores, mas onde o TR é inferior.

No caso de não existirem restrições a nível de TA, então a escolha do Modelo 3 com 4 ou 8 réplicas é uma solução capaz de diminuir em quase metade o TN apresentado, por exemplo, pelo Modelo 3 com 2 réplicas.

6 CONCLUSÕES E TRABALHO FUTURO

O trabalho desenvolvido durante o estágio envolveu um tópico relevante na área dos veículos autónomos: a forma como os nós são organização e o modelo de troca de mensagens entre eles. Neste contexto foram aplicadas bases de dados em grafo, tanto para organizar e armazenar a totalidade dos veículos, como para construir grafos de propagação de mensagens entre um subconjunto dos nós quando ocorrem incidentes rodoviários. O objetivo destes grafos de propagação é facilitar o processo de troca de mensagens de acordo com o modelo (a forma como o grafo é construído e organizado) mais vantajoso para o contexto em questão.

Os resultados mostram que a escolha do modelo tem impacto quer no tempo de notificação como no tempo de registo e disponibilidade dos dados. A comunicação direta (Modelo 1) é a mais vantajosa quando há capacidade computacional e de rede para comunicar diretamente com todos os nós envolvidos. Quando o número de nós é elevado e excede essa capacidade, é vantajoso utilizar o modelo que envolve retransmissão (Modelo 2), no entanto verifica-se um aumento do tempo de notificação devido ao número de nós envolvidos na retransmissão. Neste contexto, a replicação (Modelo 3) permite diminuir o número de nós envolvidos na retransmissão e conseqüentemente diminui o tempo de notificação.

Os objetivos do estágio foram alcançados com sucesso. Para isso contribuíram as reuniões semanais com os ambos os orientadores, onde era feita uma retrospectiva crítica ao trabalho realizado durante a semana e planificado o trabalho a desenvolver na semana seguinte. O estágio envolveu pesquisa e aprendizagem de novas ferramentas, nomeadamente bases de dados em grafo e o simulador, tecnologias com as quais não estava familiarizado.

6.1 Trabalho Futuro

Ao longo deste estágio surgiram várias outras hipóteses, algumas delas descritas ao longo deste documento mas que não foram implementadas nem testadas, que podem ser exploradas futuramente para aprimorar o estudo realizado. Sugeriria a implementação e avaliação experimental dos modelos propostos, que não foram implementados – modelos 4 e 5 descritos na secção 4.2 – de forma a compará-los com os restantes.

Seria também interessante realizar uma avaliação experimental em ambiente real, com máquinas capazes de simular realisticamente veículos, cuja localização pudesse ser efetivamente muito variável em curtos espaços de tempo. Desta forma, seria possível validar o protótipo e a aplicabilidade dos modelos propostas em cenários do mundo real.

O processo de classificação dos nós e de avaliação dos modelos de propagação de mensagens podem ser integrados num sistema que permita, de acordo com as

características do contexto de aplicação, sugerir o melhor modelo de propagação a aplicar e o grau de replicação que pode apresentar melhores resultados.

Por último, seria relevante a realização de um *benchmarking* de alguns motores de bases de dados em grafo de forma a determinar qual seria mais vantajoso para aplicar neste contexto e onde cada um dos grafos pode ser guardado, mediante as suas características.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] V. Hassija, V. Chamola, G. Han, J. J. P. C. Rodrigues, and M. Guizani, "DAGIoV: A Framework for Vehicle to Vehicle Communication Using Directed Acyclic Graph and Game Theory," *IEEE Trans Veh Technol*, vol. 69, no. 4, 2020, doi: 10.1109/TVT.2020.2968494.
- [2] D. Fernandes and J. Bernardino, "Graph databases comparison: Allegrograph, arangoDB, infinitegraph, Neo4J, and orientDB," in *DATA 2018 - Proceedings of the 7th International Conference on Data Science, Technology and Applications*, 2018. doi: 10.5220/0006910203730380.
- [3] J. Guia, V. G. Soares, and J. Bernardino, "Graph databases: Neo4j Analysis," in *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems*, 2017. doi: 10.5220/0006356003510356.
- [4] S. Patil, G. Vaswani, and A. Bhatia, "Graph Databases- An Overview," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, 2014.
- [5] J. Pokorný, "Graph databases: Their power and limitations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9339, 2015, doi: 10.1007/978-3-319-24369-6_5.
- [6] J. L. Reutter, "Current Challenges in Graph Databases (Invited Talk)," *23rd International Conference on Database Theory, {ICDT} 2020, March 30-April 2, 2020, Copenhagen, Denmark*, vol. 155, no. 3, 2020.
- [7] M. S. Almalag, S. Olariu, M. C. Weigle, and S. El-Tawab, "Peer-to-peer file sharing in VANETs using TC-MAC," in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2013*, 2013. doi: 10.1109/PerComW.2013.6529461.
- [8] Y. Bi, H. Zhao, and X. Shen, "A directional broadcast protocol for emergency message exchange in inter-vehicle communications," in *IEEE International Conference on Communications*, 2009. doi: 10.1109/ICC.2009.5198592.
- [9] Z. Zheng and A. K. Bashir, "Graph-Enabled Intelligent Vehicular Network Data Processing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, 2022, doi: 10.1109/TITS.2022.3158045.
- [10] A. Grzybek, G. Danoy, M. Seredynski, and P. Bouvry, "Evaluation of dynamic communities in large-scale vehicular networks," in *DIVANet 2013 - Proceedings of the 3rd ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, Co-located with ACM MSWiM 2013*, 2013. doi: 10.1145/2512921.2516962.
- [11] I. X. Y. Leung, P. Hui, P. Liò, and J. Crowcroft, "Towards real-time community detection in large networks," *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 79, no. 6, 2009, doi: 10.1103/PhysRevE.79.066107.

- [12] B. Sharan, M. Chhabra, and A. K. Sagar, "State-of-the-art: Data Dissemination Techniques in Vehicular Ad-hoc Networks," in *Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development, INDIACom 2022*, 2022. doi: 10.23919/INDIACom54597.2022.9763249.
- [13] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular Ad hoc networks," *IEEE Trans Veh Technol*, vol. 57, no. 3, 2008, doi: 10.1109/TVT.2007.901869.
- [14] R. S. De Sousa, A. Boukerche, and A. A. F. Loureiro, "Vehicle-Assisted Data Delivery Based on Trajectory Prediction," in *2022 IEEE Global Communications Conference, GLOBECOM 2022 - Proceedings*, 2022. doi: 10.1109/GLOBECOM48099.2022.10001329.
- [15] N. H. Hussein, C. T. Yaw, S. P. Koh, S. K. Tiong, and K. H. Chong, "A Comprehensive Survey on Vehicular Networking: Communications, Applications, Challenges, and Upcoming Research Directions," *IEEE Access*, vol. 10, 2022. doi: 10.1109/ACCESS.2022.3198656.
- [16] L. Yang and H. Li, "Vehicle-to-vehicle communication based on a peer-to-peer network with graph theory and consensus algorithm," *IET Intelligent Transport Systems*, vol. 13, no. 2, 2019, doi: 10.1049/iet-its.2018.5014.
- [17] V. Hassija, V. Chamola, G. Han, J. J. P. C. Rodrigues, and M. Guizani, "DAGIoV: A Framework for Vehicle to Vehicle Communication Using Directed Acyclic Graph and Game Theory," *IEEE Trans Veh Technol*, vol. 69, no. 4, 2020, doi: 10.1109/TVT.2020.2968494.
- [18] P. Azad, N. J. Navimipour, A. M. Rahmani, and A. Sharifi, "The role of structured and unstructured data managing mechanisms in the Internet of things," *Cluster Comput*, vol. 23, no. 2, 2020, doi: 10.1007/s10586-019-02986-2.
- [19] W. Zhang *et al.*, "Modeling IoT Equipment with Graph Neural Networks," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2902865.
- [20] C. Xie, B. Yu, Z. Zeng, Y. Yang, and Q. Liu, "Multilayer Internet-of-Things Middleware Based on Knowledge Graph," *IEEE Internet Things J*, vol. 8, no. 4, 2021, doi: 10.1109/JIOT.2020.3019707.
- [21] R. Amin, M. Hussain, M. Alhameed, S. M. Raza, F. Jeribi, and A. Tahir, "Edge-computing with graph computation: A novel mechanism to handle network intrusion and address spoofing in SDN," *Computers, Materials and Continua*, vol. 65, no. 3, pp. 1869–1890, 2020, doi: 10.32604/cmc.2020.011758.
- [22] T. Alghamdi, S. Mostafi, G. Abdelkader, and K. Elgazzar, "A Comparative Study on Traffic Modeling Techniques for Predicting and Simulating Traffic Behavior," *Future Internet*, vol. 14, no. 10, Oct. 2022, doi: 10.3390/fi14100294.



**Instituto Superior
de Engenharia**

Politécnico de Coimbra