



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Uma Proposta de Solução para Mobilidade Urbana em Cidades Inteligentes Utilizando Fog Computing e Internet das Coisas

Samuel James de Lima Barroso

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientadora

Profa. Dra. Aletéia Patrícia Favacho de Araújo

Brasília
2023

Dedicatória

Eu dedico e me motivo a realizar este trabalho de graduação por meu presente e futuro como Engenheiro de Computação, Empresário, Líder, Inventor e Desenvolvedor de Projetos. Dedico também à minha namorada, que me motivou e me apoiou em diversos momentos na realização deste trabalho e durante toda a minha jornada na Universidade. Dedico também à minha família, que me deu todas as condições que precisava pra chegar até a Universidade e sempre me apoiou em todos os meus projetos, me provendo uma base sólida de valores que formam meu caráter. Dedico também a todos os professores, mestres, mentores e líderes que fizeram parte da minha jornada como estudante, principalmente à orientadora deste trabalho, Profa. Dra. Aletéia, que motivou e deu todas as oportunidades necessárias para a conclusão deste projeto. Dedico finalmente a Deus, sempre justo, que me dá as oportunidades, desafios, inteligência, força de vontade, esperança, motivação, saúde, e tudo que me é necessário para viver.

Agradecimentos

Agradeço, primeiramente, a todas as pessoas que me apoiaram durante minha formação acadêmica, desde minha infância até formação de ensino superior, incluindo minha família e todos os professores que me incentivaram nessa jornada. Agradeço, também, a todas as instituições de ensino que fizeram parte da minha formação acadêmica, de modo especial à Universidade de Brasília, pela disponibilização dos recursos necessários para realização da minha formação em Engenharia de Computação. Além disso, deixo registrado meu agradecimento a todas as instituições tecnológicas e grupos de desenvolvimento que disponibilizam sistemas de softwares em código aberto, o que serve de base para diversos projetos, como fonte de estudo e aprendizado. Agradeço também a todas as pessoas que confiaram na minha capacidade e me deram oportunidades de trabalhar na área de tecnologia e desenvolver projetos reais, mesmo durante os anos de faculdade, o que foi muito importante para manter a motivação necessária para continuar estudando, pois hoje tenho certeza de que o que foi estudado possui aplicação real e impactante na vida de diversas pessoas. Agradeço, finalmente, a Deus, por todas as bênçãos e livramentos concedidos na minha vida, todo os dons e capacidades que tenho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Este trabalho apresenta um estudo teórico e proposta de aplicação prática de conceitos relacionados a *Fog Computing* e Internet das Coisas dentro do contexto da mobilidade urbana em Cidades Inteligentes. Para isso, foi realizada uma prova de conceito e simulação de um sistema denominado Modo Colônia - um sistema de controle de tráfego para veículos autônomos em Cidades Inteligentes que funciona por meio do processamento de informações de trânsito em tempo real e. Assim, o caso de uso proposto necessita de uma infraestrutura de processamento de dados em baixa latência para seu funcionamento correto, ou seja, uma infraestrutura baseada em *Fog Computing*. Para validação da solução proposta, foram coletados dados comparativos entre o modo proposto e o modelo convencional, denominados, neste trabalho, Modo Colônia e Modo Legado, respectivamente. Com isso, foram obtidos resultados promissores no quesito de aplicabilidade do Modo Colônia, bem como suas otimizações, podendo praticamente dobrar a velocidade média de tráfego de veículos nas regiões adjacentes a um cruzamento de trânsito e, ainda, eliminar por completo o tempo de espera em sinais vermelhos, melhorando a fluidez do trânsito e diminuindo o desperdício energético do sistema de transporte e, conseqüentemente, melhorando a experiência do usuário.

Palavras-chave: Computação em Névoa, Internet das Coisas (IoT), Computação em Nuvem, Sistemas de Controle de Tráfego, Modo Colônia para Tráfego Urbano.

Abstract

This work provides a theoretical research and proposes a practical application related to Fog Computing and Internet of Things in the context of urban mobility in Smart Cities by proving the concept of a system denominated Colony Mode - a traffic control system for autonomous vehicles powered by the processing of real-time traffic data that demands low latency and fast response time for its correct operation. Thus, a Fog Computing infrastructure can be used. Furthermore, in order to validate the proposed solution, comparative data between the proposed and conventional model were collected. These models are denominated, in this work, by Colony Mode and Legacy Mode, respectively. At this point, the analysis of the collected data has shown promising results regarding the applicability of the proposed Colony Mode. Its optimizations could virtually double the average speed of traffic in the surroundings of a traffic cross and also completely eliminate the need to stop at a red light, leading to better fluidity of traffic, less waste of energy, and a better user experience on urban traffic systems.

Keywords: *Fog Computing, Internet of Things (IoT), Cloud Computing, Traffic Control Systems, Colony Mode for Urban Traffic.*

Sumário

1	Introdução	1
1.1	Objetivos	2
1.2	Organização deste Trabalho	2
2	Referencial Teórico	4
2.1	<i>Cloud Computing</i>	4
2.1.1	Computação como Serviço	5
2.1.2	Modelos de Implantação	6
2.2	<i>Fog Computing</i>	6
2.2.1	Nó Fog	7
2.2.2	Características Essenciais	8
2.2.3	Arquitetura em Três Camadas	9
2.3	Internet das Coisas (IoT)	10
2.4	Outras Tecnologias Relacionadas	11
3	Casos de Uso e Desafios Tecnológicos	14
3.1	Casos de Uso de <i>Fog Computing</i>	14
3.2	Desafios para Implantação de <i>Fog Computing</i>	17
3.3	Contradições de Latência nas Rotas da Internet	19
4	Proposta e Implementação	22
4.1	Visão Geral	23
4.2	Estrutura da Aplicação Desenvolvida	24
4.3	Ambiente de Simulação	25
4.3.1	Ciclos de Gerenciamento e Constantes de Simulação	25
4.3.2	Entidades Simuladas	28
4.4	Funcionamento do Modo Colônia	29
4.4.1	Operação dos Veículos Autônomos	31
4.5	Sistema de Controle de Tráfego	32
4.5.1	Autorização de Travessia e Segurança do Cruzamento	33

4.5.2	Algoritmos para Controle de Velocidade e Prioridade de Travessia . . .	34
5	Análise e Discussão dos Resultados	37
5.1	Análise da Velocidade de Tráfego	37
5.1.1	Análise da Velocidade Individual	39
5.1.2	Análise das Médias das Velocidades	41
5.1.3	Análise dos Deslocamentos Individuais	41
5.2	Análise de Utilização de Recursos	42
5.3	Considerações Finais das Análises	44
6	Considerações Finais	46
6.1	Síntese dos Resultados Obtidos	46
6.2	Limitações e Trabalhos Futuros	47
	Referências	50

Lista de Figuras

2.1	<i>Fog Computing</i> dando suporte a um ecossistema baseado em Nuvem e dispositivos finais inteligentes, adaptado de [1].	7
2.2	Arquitetura em três camadas de <i>Fog Computing</i> , adaptado de [2].	9
2.3	Interfaces entre <i>Fog</i> e IoT, adaptado de [3].	11
3.1	Ilustração, feita pelo autor, para exemplificar o cenário disposto na Situação A.	20
3.2	Ilustração, feita pelo autor, para exemplificar o cenário disposto na Situação B.	21
4.1	Trecho de código utilizado para iniciar as <i>Threads</i> de ciclo de gerenciamento.	26
4.2	Trecho de código onde são feitas as definições de importantes constantes da simulação.	26
4.3	Ilustração da quarta cor utilizada no semáforo para indicar o Modo Colônia, a cor azul.	30
4.4	<i>Screenshot</i> da simulação do cruzamento operando em Modo Colônia.	31
4.5	<i>Screenshot</i> da simulação do cruzamento operando em Modo Legado.	31
4.6	Momento de simulação onde foi priorizada a passagem de veículos em sentidos opostos durante operação em Modo Colônia.	35
5.1	Configuração do ambiente de testes e coleta de dados.	38
5.2	Dados de velocidade para Modo Legado.	39
5.3	Dados de velocidade para Modo Colônia.	40

Lista de Tabelas

5.1	Dados comparativos entre deslocamentos do Modo Legado versus Modo Colônia.	42
5.2	Estimativa de utilização de recursos de rede para funcionamento do Modo Colônia.	43

Lista de Abreviaturas e Siglas

BGP *Border Gateway Protocol.*

BLE *Bluetooth Low Energy.*

CCENT *Cisco Certified Entry Networking Technician.*

CDN *Content Delivery Network.*

EPC *Electronic Product Code.*

FPS *Frames por Segundo.*

IA *Inteligência Artificial.*

IaaS *Infrastructure as a Service.*

IoT *Internet of Things.*

IoV *Internet of Vehicles.*

IP *Internet Protocol.*

ITS *Intelligent Transportation Systems.*

IXP *Internet Exchange Point.*

LTE *Long Term Evolution.*

NFC *Near Filed Communication.*

NIST *National Institute of Standards and Technology.*

PaaS *Platform as a Service.*

PTT *Ponto de Troca de Tráfego.*

PTT-SP Ponto de Troca de Tráfego de São Paulo.

RFID *Radio Frequency Identification.*

SaaS *Software as a Service.*

SCT Sistema de Controle de Tráfego.

Wi-Fi *Wireless Fidelity.*

WSN *Wireless Sensor Networks.*

Capítulo 1

Introdução

Ao longo dos anos, os paradigmas computacionais estiveram em constante evolução, evoluindo desde computação distribuída, paralela, em *cluster*, *peer-to-peer*, *grid* até chegar *Cloud Computing* [2]. Nesse contexto, *Cloud Computing* é um modelo computacional que possibilita acesso ubíquo, de forma conveniente e sob demanda, a um conjunto de recursos computacionais configuráveis (tais como redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados e liberados com mínimo de esforço de gerenciamento ou interação com o provedor de serviços [4].

Nesse contexto, para a implantação das tecnologias de *Cloud Computing*, foram construídos grandes centros de processamento centralizado, também conhecidos como *data-centers*, que geralmente estão dispostos em cidades distantes aos usuários finais, o que pode gerar problemas de alta latência[5]. Nesse contexto, com objetivo de se compor um paradigma computacional que pudesse lidar melhor com aplicações que dependem de baixa latência, surgem as ideias de *Fog Computing*, que consiste em uma tecnologia auxiliar a *Cloud Computing* que disponibiliza infraestrutura de processamento levando em conta a localização geográfica do usuário final [1].

Dessa forma, *Fog Computing*, ou névoa computacional, é uma infraestrutura que se encontra mais próxima aos usuário e dispositivos finais do que as Nuvens, uma vez que dispõe de recursos geograficamente posicionados entre as Nuvens e os dispositivos finais [1]. Assim, uma infraestrutura *Fog* pode atuar como um intermediador entre dispositivo final e Nuvem, agregando em qualidade de serviço, disponibilidade e redução de custos. Em outros termos, a névoa computacional vem como uma extensão da Nuvem e, segundo Gill et al. [6], nunca poderá substituí-la, uma vez que ainda precisamos dela para lidar com problemas de dados grandes e complexos. Devido a essas características, *Fog Computing* se apresenta como uma solução de extrema afinidade para processamento de dados de dispositivos da Internet das Coisas [3].

A Internet das Coisas é, de forma literal, a conexão de diversas *Coisas* na Internet. O

termo *Coisas*, nesse contexto, segundo Madakam et. al [7] se refere a qualquer objeto ou pessoa que pode ser distinguível no mundo real - inclusive objetos do dia-a-dia que *ainda* não estão conectados na rede, tais como roupas, móveis, equipamentos, monumentos, materiais de construção, plantas, eletrodomésticos, até animais e pessoas. A implantação dessa tecnologia representa um aumento considerável na quantidade de dados que devem ser processados pelas infraestruturas de nuvem, além de geralmente exigirem baixa latência para seu funcionamento, portanto, tornando-se fonte de diversos casos de uso para *Fog Computing*[3].

Diante do exposto, o contexto escolhido para pesquisa e desenvolvimento deste projeto foi o ambiente de *Fog Computing*, por ser um tema em constante crescimento e com diversas pesquisas realizadas nos últimos anos. Além disso, esse trabalho propõe um caso de uso prático das tecnologias *Fog Computing* e Internet das Coisas, por meio do desenvolvimento de um sistema de controle de tráfego em tempo real para veículos autônomos no contexto de uma Cidade Inteligente. Esse sistema foi denominado, pelo autor, Modo Colônia.

1.1 Objetivos

O objetivo geral deste trabalho é trazer uma aplicação prática de *Fog Computing*, tema tão citado na literatura, mas que, ainda, não foi implantado em massa pelos grandes provedores de *Cloud Computing*. Para cumprir o objetivo geral, faz-se necessário atingir os seguintes objetivos específicos:

- Pesquisa e levantamento teórico do estado da arte, relacionando os temas escolhidos, incluindo casos de usos existentes;
- Aplicação prática de um caso de uso conjunto das tecnologias *Fog Computing* e Internet das Coisas;
- Desenvolvimento de um sistema em Software que possa ser utilizado para coletar dados práticos a respeito da ideia de caso de uso;
- Coleta e análise de dados referentes à aplicação desenvolvida, a fim de demonstrar quão efetivo seria utilizar a proposta apresentada na prática, além de pontuar seus possíveis benefícios.

1.2 Organização deste Trabalho

No Capítulo 2, para preparar o leitor com a base teórica necessária, é apresentado um estudo do estado da arte sobre os temas *Cloud Computing*, *Fog Computing* e Internet das

Coisas, na forma de um referencial teórico que inclui definições, características principais e tecnologias relacionadas.

Em seguida, no Capítulo 3, com o objetivo de contextualizar melhor a proposta deste projeto, são apresentados diversos casos de uso relacionados a *Fog Computing*, desafios para implementação do conceito na prática e, por fim, uma discussão a respeito da contradição existente na latência das rotas da Internet.

No Capítulo 4 é apresentado, de fato, a proposta de implementação prática deste trabalho, incluindo detalhes da aplicação desenvolvida. Para validar a proposta apresentada, o Capítulo 5 traz a análise e discussão dos resultados obtidos com a implantação das ideias apresentadas. Por fim, o Capítulo 6 apresenta as considerações finais, em formato de conclusão e resumo de resultados atingidos, além de trazer um apanhado de possibilidades para trabalhos futuros dentro do tema.

Capítulo 2

Referencial Teórico

Este capítulo apresenta definições e conceitos básicos necessários para o entendimento da proposta deste trabalho que é a aplicação prática em *Fog Computing*. Primeiramente será apresentado um breve resumo sobre parte do ecossistema de *Cloud Computing*, cuja plataforma *Fog Computing* está inserida. Em seguida, um breve apanhado bibliográfico a respeito da Internet das Coisas (do Inglês, *Internet of Things* (IoT)) e de seus dispositivos. Finalmente, será apresentado também algumas tecnologias relacionadas a esses temas.

2.1 *Cloud Computing*

Existem diversas formas possíveis de se definir e discorrer a respeito de *Cloud Computing*, *Fog Computing* e suas características, porém, pode-se destacar a padronização internacional feita pelo *National Institute of Standards and Technology* (NIST), uma vez que é a mais utilizada na literatura [2]. Assim sendo, este capítulo foca em apresentar os conceitos de forma objetiva e sucinta ao leitor, lançando mão das definições de NIST.

Computação em Nuvem (do Inglês, *Cloud Computing*), de acordo com NIST [4], é um modelo computacional que possibilita acesso ubíquo, de forma conveniente e sob demanda, a um conjunto de recursos computacionais configuráveis (tais como redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados e liberados com mínimo de esforço de gerenciamento ou interação com o provedor de serviços. Este modelo de *Cloud Computing* é composto de cinco características essenciais [4]:

- Acesso sob-demanda e *self-service*: um usuário pode unilateralmente provisionar recursos computacionais, tais como tempo de servidor e armazenamento em rede, sob-demanda e de forma automática, sem que seja necessário interação humana com cada provedor de serviços;

- Acesso amplo a rede: os recursos da Nuvem estão disponíveis na rede e podem ser acessados por diferentes tipos de dispositivo de usuário, seja por computadores, *smarthphones*, *tablets*, celulares, etc;
- Compartilhamento de recursos: os recursos do provedor de *Cloud Computing* são compartilhados e agrupados para servir múltiplos clientes utilizando a arquitetura de diversos locatários, viabilizando melhor aproveitamento desses recursos e diminuição da ociosidade de recursos ligados;
- Elasticidade: recursos podem ser provisionados de forma elástica e liberados para escalar rapidamente, seja alocando ou desalocando recursos, de acordo com a demanda, podendo acontecer de forma automática, manual ou programada;
- Serviço medido: os sistemas em nuvem controlam e medem automaticamente a utilização dos recursos disponibilizados, de acordo com o tipo de serviço. Dessa forma, a tarifação geralmente é feita no modelo *pay per use*, isto é, pague conforme o que foi utilizado.

2.1.1 Computação como Serviço

Ademais, *Cloud Computing* está majoritariamente atrelada ao conceito de Computação como Serviço (do Inglês, *Computing as a Service*), isto é, uma prestação de serviço que disponibiliza recursos computacionais na Internet em troca de recursos financeiros. Atualmente existem diversos modelos de serviço existentes, mas NIST [4] enfatiza os três principais:

- *Software as a Service* (SaaS): se trata da disponibilização de uma aplicação de *Software* que utiliza recursos da Nuvem de um provedor de serviços. Neste caso, o cliente não gerencia ou controla a infraestrutura utilizada, nem mesmo detalhes específicos da aplicação, com exceção da possibilidade de se configurar alguns parâmetros pré-determinados pelo provedor;
- *Platform as a Service* (PaaS): se trata da disponibilização de uma plataforma para o usuário executar aplicações próprias ou adquiridas que foram criadas utilizando linguagens de programação, bibliotecas, serviços e ferramentas compatíveis com o provedor de serviços. O usuário não gerencia nem controla a infraestrutura de Nuvem, incluindo rede, servidores, sistemas operacionais ou armazenamento, mas tem controle sobre as aplicação e possivelmente configurações sobre o ambiente de hospedagem da aplicação;
- *Infrastructure as a Service* (IaaS): se trata da disponibilização da capacidade de provisionar processamento, armazenamento, redes e outros recursos de computação

fundamentais, sendo que o usuário pode executar softwares diversos, o que inclui sistemas operacionais e aplicações. O usuário não gerencia a infraestrutura de nuvem, porém tem controle sobre sistemas operacionais, armazenamento e aplicações, possivelmente tem acesso limitado a alguns componentes de rede selecionados, (por exemplo, *firewalls* de *host*).

2.1.2 Modelos de Implantação

É muito importante entender, também, os modelos de implantação existentes para *Cloud Computing*. Modelos de implantação dizem respeito ao público-alvo de uma infraestrutura de Nuvem, isto é, quem pode contratar e usufruir de seus recursos. Ainda segundo NIST [4], os principais modelos de implantação são:

- Nuvem Privada: a infraestrutura de Nuvem é provisionada exclusivamente para ser utilizada por uma única organização;
- Nuvem Comunitária: a infraestrutura de Nuvem é provisionada exclusivamente para uma comunidade específica de usuários de organizações que tem interesses em comum (por exemplo, missão, requisitos de segurança, política e considerações de *compliance*);
- Nuvem Pública: a infraestrutura de Nuvem é provisionada para o público geral;
- Nuvem Híbrida: se trata de uma infraestrutura de Nuvem mista de duas ou mais infraestruturas de Nuvem (privada, comunitária ou pública) que são conectadas por meio de tecnologias que permitem portabilidade de dados e aplicações.

2.2 *Fog Computing*

Fazendo jus ao nome, a Névoa Computacional (do Inglês, *Fog Computing*) é uma infraestrutura que se encontra mais próxima aos usuários e aos dispositivos finais do que as Nuvens, uma vez que dispõe de recursos geograficamente posicionados entre as Nuvens e os dispositivos finais [1]. Para tal, *Fog Computing* flexibiliza algumas características da Nuvem, como será visto a seguir, e prioriza características relacionadas à dispersão geográfica dos recursos computacionais, tais como baixa latência e largura de banda [2]. Dessa forma, uma infraestrutura *Fog* pode atuar como um intermediador entre dispositivo final e Nuvem, agregando em qualidade de serviço, disponibilidade e redução de custos. Em outros termos, a névoa computacional vem como uma extensão da Nuvem e, segundo Gill et al. [6], nunca poderá substituí-la, uma vez que ainda precisamos dela para lidar com problemas de dados grandes e complexos.

NIST[1] define *Fog Computing* como um modelo em camadas que permite acesso ubíquo e contínuo a uma infraestrutura de recursos computacionais escaláveis. Esse modelo facilita a disponibilização de aplicações e serviços distribuídos sensíveis a latência, consiste de *Fog Nodes* (físicos ou virtuais), dispostos entre dispositivos finais inteligentes e serviços centralizados (Nuvem). Dessa forma, a camada *Fog Computing* minimiza o tempo entre requisição e resposta das aplicações e provê para os dispositivos finais recursos computacionais locais e, quando necessário, conectividade com os recursos da Nuvem.

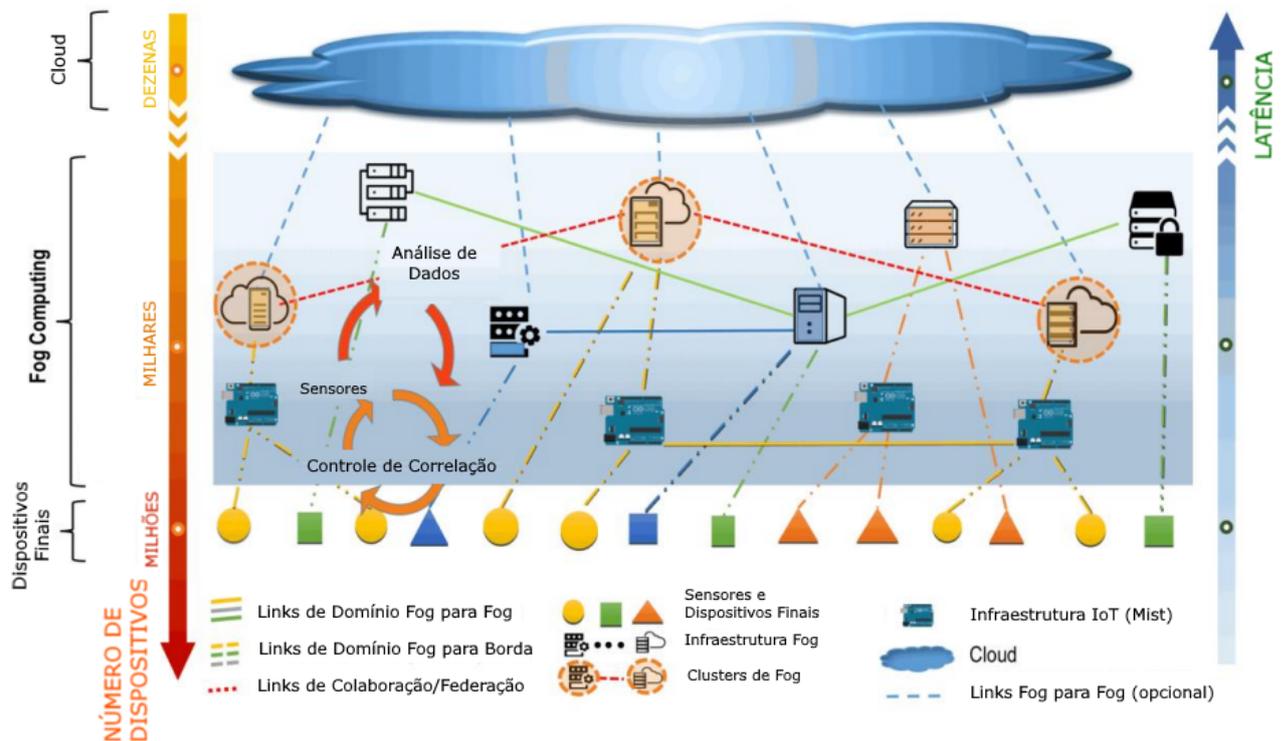


Figura 2.1: *Fog Computing* dando suporte a um ecossistema baseado em Nuvem e dispositivos finais inteligentes, adaptado de [1].

2.2.1 Nó Fog

Ainda de acordo com NIST[1], um Nó Fog (do Inglês, **Fog Node**) é o principal componente da arquitetura *Fog*. *Fog Nodes* podem ser componentes físicos (por exemplo, roteadores, *switches*, servidores, etc) ou virtuais (por exemplo, máquinas virtuais, *switches* virtualizados, etc) que estão fortemente acoplados aos dispositivos finais ou às redes de acesso, aos quais proveem recursos computacionais. *Fog Nodes* são sensíveis ao contexto e dão suporte a sistemas de gerenciamento e comunicação de dados comum. Podem ser

organizados em *clusters*, seja verticalmente (para suportar isolamento), horizontalmente (para suportar federação), ou relativamente à distância de latência entre *Fog Nodes* e os dispositivos finais inteligentes. Por fim, um *Fog Node* deve dar suporte a pelo menos um dos seguintes atributos:

- Autonomia: *Fog Nodes* podem operar de forma independente, tomando decisões locais, a nível de nó ou de *Cluster*;
- Heterogeneidade: *Fog Nodes* estão dispostos de formas diversas e podem ser provisionados em uma variedade diversa de ambientes, isto é, utilizando de recursos computacionais (processadores, memória, rede) de capacidades distintas operando em rede;
- Agrupamento hierárquico: *Fog Nodes* dão suporte a estruturas hierárquicas, com diferentes camadas provendo subconjuntos de funcionalidades e serviços, mas trabalhando juntas e de forma íntegra;
- Gerenciabilidade: *Fog Nodes* são gerenciados e orquestrados por sistemas complexos que podem exercer a maioria das operações de rotina de forma automática;
- Programabilidade: *Fog Nodes* são inerentemente programáveis em múltiplos níveis, por múltiplos *stakeholders* - tais como operadoras de rede, provedores de equipamentos, profissionais de área ou usuários finais.

2.2.2 Características Essenciais

NIST[1] define seis características essenciais para distinguir *Fog Computing* de outros paradigmas, são elas:

- Conhecimento de localização e baixa latência: *Fog Computing* oferece a latência mais baixa possível por conta do conhecimento dos *Fog Nodes* de sua própria localização lógica dentro do contexto do sistema como um todo e do conhecimento do custo em latência de se comunicar com outros nós. Como os *Fog Nodes* estão próximos aos dispositivos finais, a análise e resposta dos dados gerados por esses dispositivos é muito mais rápida quando comparada à Nuvem;
- Distribuição geográfica: em contraste com o processamento centralizado das Nuvens, os serviços e aplicações alvo de *Fog Computing* demandam recursos distribuídos geograficamente de forma ampla;
- Heterogeneidade: *Fog Computing* dá suporte à coleta e ao processamento de dados em diversos tipos de hardware, acessados por meio de diversas formas de comunicação em rede;

- Interoperabilidade e Federação: dar suporte a certos tipos de aplicação (tais como serviço de *streaming* em tempo real) exige a cooperação entre diferentes provedores de serviço. Assim sendo, os elementos de *Fog Computing* precisam trabalhar de forma conjunta, provendo serviços federados através de diferentes domínios. Além disso, segundo Bachiega et al.[2], *Fog Computing* depende da existência e da interoperabilidade com a Nuvem e, de acordo Gill et. al.[6] não pode substituí-la;
- Interações em tempo-real: aplicações de *Fog Computing* envolvem interações em tempo real ao invés de processamento em lote;
- Escalabilidade e agilidade: *Fog Computing* é adaptativo por natureza, seja em nível de *Cluster* ou de *Cluster* de *Clusters*, permitindo, por exemplo, computação elástica, agrupamento de recursos, mudança de cargas de dados e condições de rede variantes.

2.2.3 Arquitetura em Três Camadas

Existem diversas formas de se sub-dividir as camadas de *Fog Computing* e *Cloud Computing*. Para este trabalho, será considerada a classificação de 3 camadas, com base no artigo de Bachiega et al. (Bachiega, et al., 2022) [2], sendo elas: Camada *Cloud*, Camada *Fog*, Camada IoT/Usuário Final. Vide Figura 2.2.

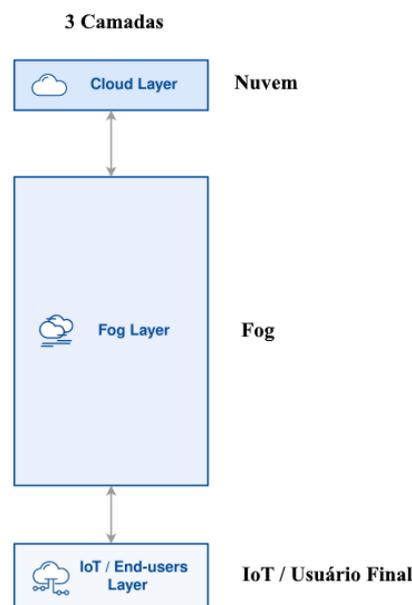


Figura 2.2: Arquitetura em três camadas de *Fog Computing*, adaptado de [2].

Com base na Figura 2.2, fica claro o posicionamento das camadas da arquitetura *Fog*. A Camada *Cloud* está na parte superior, e é responsável pelo processamento mais intenso

e das cargas que a *Fog* não é capaz de lidar, além de incluir possíveis *back-ups* de dados. A camada intermediária, entre usuário e nuvem, é de fato a Camada *Fog*. A última camada pode ser caracterizada ou por usuários finais ou por dispositivos IoT, a depender de qual módulo da aplicação está sendo referenciado. Para os módulos de interface de usuário, será nomeada Camada de Usuário. Para os módulos que lidam com dispositivos IoT, será nomeada Camada de IoT.

2.3 Internet das Coisas (IoT)

Com base nas características levantadas a respeito de *Fog Computing*, fica claro o forte alvo em aplicações que exigem conectividade baseada em posicionamento geográfico e baixa latência. Nesse mesmo contexto, é possível citar diversos exemplos de aplicações também relacionadas à Internet das Coisas, tais como sistemas de transporte inteligentes (carros autônomos)[8], aplicações de casas inteligentes [3], cabines de controle de tráfego nas rodovias [3], medidores inteligentes e controladores de malha de distribuição elétrica [3].

A Internet das Coisas é, de forma literal, a conexão de diversas *Coisas* na Internet. O termo *Coisas*, nesse contexto, segundo Madakam et. al [7] se refere a qualquer objeto ou pessoa que pode ser distinguível no mundo real - inclusive objetos do dia-a-dia que *ainda* não estão conectados na rede, tais como roupas, móveis, equipamentos, monumentos, materiais de construção, plantas, eletrodomésticos, até animais e pessoas.

Diferentemente da padronização internacional feita por NIST para *Fog* e *Cloud Computing*, não existe uma definição única e globalmente aceita para Internet das Coisas, mas o que todas têm em comum é a ideia de que a primeira versão da Internet conta com dados criados por pessoas, ao passo que a próxima versão (IoT) conta com dados gerados por coisas [7]. A melhor definição do termo, segundo segundo Madakam et. al [7], seria, em tradução livre:

“Uma rede aberta e compreensível de objetos inteligentes que têm a capacidade de se auto organizar, compartilhar informações, dados e recursos, reagindo e agindo de acordo com as situações e mudanças no ambiente”.

Finalmente, para definir a relação entre os dispositivos da Internet das Coisas e a infraestrutura de *Fog Computing*, existem três interfaces possíveis entre as camadas IoT, *Fog* e *Cloud*, definidas por Chiang et. al[3], são elas: Interfaces *Cloud-Fog*, Interfaces *Fog-Fog* e Interfaces *Fog-IoT*. Esta última representa a interface principal que será abordada neste trabalho. A visualização dessas interfaces pode ser vista na Figura 2.3.

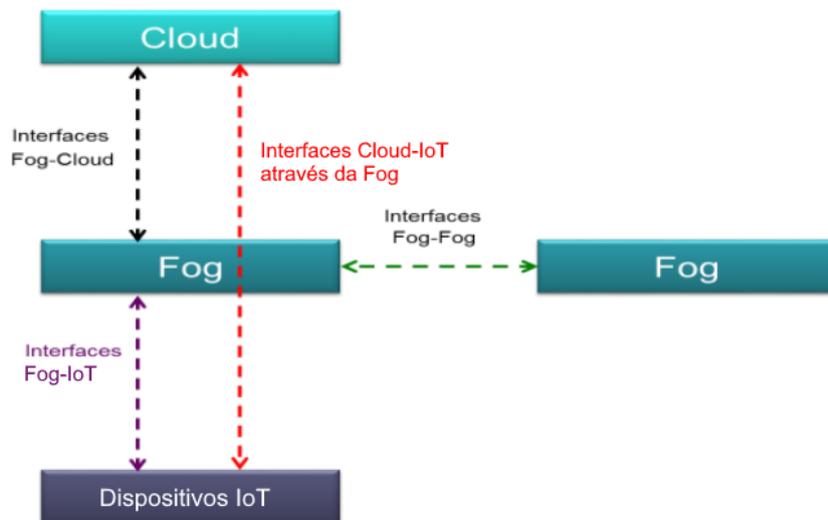


Figura 2.3: Interfaces entre *Fog* e IoT, adaptado de [3].

2.4 Outras Tecnologias Relacionadas

Existem diversas tecnologias importantes para dar suporte a este mundo das coisas conectadas na Internet, seja via *Fog*, ou diretamente na nuvem, ou até mesmo em sistemas *Peer-to-Peer*. Dentre essas e outras tecnologias que serão comentadas neste trabalho, convém destacar:

- *Radio Frequency Identification* (RFID): é um sistema que transmite a identidade de um objeto ou pessoa utilizando rádio frequências na forma de um número serial. Essa tecnologia tem um papel fundamental no mundo IoT, uma vez que resolve problemas de identificação de objetos a um custo acessível. Essa tecnologia é classificada em três categorias, de acordo com o tipo de alimentação utilizada, podendo ser ativa, passiva ou semi passiva [7];
- *Electronic Product Code* (EPC): é um código de 64 ou 98 bits gravado eletronicamente em uma *tag* RFID e tem como objetivo aprimorar o sistema de código de barras. Esse código pode armazenar informações a respeito do tipo de produto, número de série, especificações, informações do fabricante, etc. Ele tem quatro componentes, um componente de nomeação, um de componente de descoberta, um componente de informação e um componente de segurança [7];
- *Near Filed Communication* (NFC): é um conjunto de tecnologias sem fio em 13.56 MHz com alcance típico de 4cm. Essa tecnologia permite que usuários ao redor do mundo façam transações financeiras, troquem conteúdo digital e conectem dispositivos eletrônicos por meio de aproximação. O NFC permite a inicialização intuitiva

de redes sem fio, além de ser complementar ao *Bluetooth* e 802.11 nas suas capacidades de longo alcance em uma distância curta de até 10cm. Funciona também em ambientes sujos, não requer visada, é um método de conexão simples e fácil. Foi inicialmente desenvolvido pelas empresas Philips e Sony. A taxa de troca de dados, atualmente, chega a aproximadamente 424 Kbps [7];

- *Bluetooth*: é uma tecnologia sem fio de curto alcance barata e que elimina a necessidade de cabeamento proprietários entre dispositivos como PCs, *notebooks*, câmeras, impressoras, entre outros, em um raio efetivo de 10 a 100 metros. A velocidade de conexão geralmente é feita em uma velocidade inferior a 1 Mbps, utilizando a especificação padronizadora IEEE 802.15.1 [7]. Expandindo essas características e focando em baixíssimo consumo de energia, foi desenvolvido a versão *Bluetooth Low Energy* (BLE)[9], excelente para uso em dispositivos IoT;
- *ZigBee*: é um dos protocolos desenvolvidos para aprimorar o funcionamento de redes de sensores sem fio, tendo baixo custo, baixa taxa de transmissão de dados, alcance de cerca de 100 metros, escalabilidade, alta confiabilidade e flexibilidade. O interessante desse protocolo é que pode ser implementado em topologias do tipo estrela, árvore *cluster* e *mesh*. É muito utilizado em automação residencial, agricultura digital, controle industrial, monitoramento médico e sistemas de energia [7];
- *Wireless Sensor Networks* (WSN): se trata de uma rede constituída de dispositivos autônomos distribuídos espacialmente, utilizam sensores para cooperativamente monitorar condições físicas ou ambientais, tais como temperatura, som, vibração, pressão, movimento ou poluentes, em diferentes locais [7];
- *Wireless Fidelity* (Wi-Fi): é uma tecnologia que permite computadores e outros dispositivos comunicarem entre si via rede sem fio. Hoje é uma tecnologia muito utilizada, seja em residências, escritórios, empresas, aeroportos, hotéis, etc. Essa tecnologia é definida por diversos padrões, tais como 802.11a, 802.11b, 802.11g e 802.11n [7];
- *Internet Protocol* (IP): é o protocolo primário da Internet, desenvolvido na década de 1970. Está disponíveis nas versões IPv4 e IPv6, definindo diversos subprotocolos e camadas [7]. Com o endereçamento de 128-bits do IPv6, é possível que cada dispositivo IoT seja identificado de forma única e exclusiva na Internet, o que não era possível com os 32-bits do IPv4;
- *Border Gateway Protocol* (BGP): é o protocolo principal utilizado na Internet para roteamento entre Sistemas Autônomos. Ele permite que redes troquem informações de acessibilidade com outros sistemas BGP [10].

Para dar vida aos conceitos aqui apresentados, nos próximo capítulo serão apresentados diversos casos de uso relacionado às tecnologias *Fog Computing* e IoT.

Capítulo 3

Casos de Uso e Desafios Tecnológicos

Este capítulo apresenta diversos casos de uso referentes às tecnologias apresentadas no capítulo anterior. Além disso, serão apresentados também alguns desafios encontrados para difusão de *Fog Computing* no mercado atual, incluindo uma explicação e reflexão sobre como realmente funciona a latência das conexões feitas na Internet, onde nem sempre as rotas com menor latência são escolhidas para tráfego de dados.

3.1 Casos de Uso de *Fog Computing*

Fog Computing é um paradigma computacional com potencial imenso e que traz diversas vantagens em relação ao modelo convencional de *Cloud Computing*. Dentre essas vantagens e potenciais, vale ressaltar que essa tecnologia pode:

- Dar suporte à mobilidade e à distribuição geográfica: um dos maiores potenciais de *Fog Computing* é a densa distribuição de *Fog Nodes* ao longo de rodovias, estradas e torres de telefone celular. Em adição, *Fog Computing* provê um suporte melhor à mobilidade do que a nuvem, devido à possibilidade do *Fog Node* estar próximo ao usuário, uma vez que estão alocados na região geográfica dos usuários móveis. [5]
- Diminuir a latência: outro potencial é a proximidade dos *Fog Nodes* aos usuários finais, em oposição aos *data centers* de *Cloud*. Conseqüentemente, a latência da transmissão de dados será reduzida. Isso mitiga as desvantagens dos atrasos da camada *Cloud*, principalmente para aplicações criticamente dependentes em latência, tais como transmissões em tempo real, jogos multijogador e sistemas de monitoramento de saúde. Um estudo feito pela Akamai Networks mostrou que, ao reduzir a distância entre servidores e usuários, a latência e tempo de download são reduzidos; [5]

- Reduzir a carga da nuvem: recentemente houve um aumento significativo de dispositivos baseados em localização, tais como registradores de atividade de esportes, e o aumento no volume de dados gerados por esses dispositivos. A maior parte desses dados seria recebido pelos *datacenters* de *Cloud* como linhas de informações de usuário, tais como velocidade, temperatura e localização. Um estudo feito pela Endomondo mostrou que o número de linhas de informação gerada mensalmente chega a casa de 2.8 a 6.3 bilhões. Considerando a grande quantidade de dados gerados diariamente por dispositivos IoT e enviada para os servidores *Cloud*, eles podem acabar sobrecarregados. A camada de *Fog Computing* pode ser adotada para filtrar os dados desnecessários antes de serem enviados para *Cloud*, assim, reduzindo a carga da nuvem; [5]
- Melhorar a eficiência energética: segundo Haouari et. al. [5], os pesquisadores de [11] realizaram um estudo comparativo entre a eficiência energética de se executar aplicações baseadas em nuvem na própria nuvem versus em pequenos servidores *Fog*. Os resultados desse estudo mostraram que aplicações interativas *online* consumiram uma grande quantidade de energia devido aos grandes *overheads* da interação com os servidores da nuvem em comparação com os servidores *Fog*. Uma outra abordagem para reduzir o consumo de energia é a prática de cacheamento de dados em servidores *Fog*, onde dados frequentemente utilizados da nuvem, tais como os vídeos populares mais assistidos, ficam armazenados nas proximidades do usuário [5]. Essa prática de cacheamento é comumente utilizada por redes de distribuição de dados, chamadas de *Content Delivery Network* (CDN);

Com base nessas características, vantagens e potenciais de uso para *Fog Computing*, é possível encontrar na literatura diversas possíveis aplicações e casos de uso, principalmente quando se trata de aplicações relacionadas à tendência da Internet das Coisas. Dentre as aplicações e casos de uso mais interessantes, vale ressaltar:

- Cidades Inteligentes: nos últimos anos, o conceito de Cidade Inteligente tem atraído bastante interesse devido à promessa de aprimorar a qualidade de vida. Uma Cidade Inteligente é uma região urbana onde vários setores colaboram para alcançar resultados sustentáveis por meio da análise de dados em tempo real. Construir cidades inteligentes impõe desafios de garantir resposta acurada e rápida durante o monitoramento de infraestruturas vitais para o funcionamento das cidades, tais como redes de gás, óleo, túneis subterrâneos, rodovias, sistemas de distribuição de energia elétrica, etc. Além disso, a quantidade massiva de dados gerados por dispositivos de monitoramento e sensores tende a gerar dificuldades para se analisar essas grandes quantidades de dados [5]. Nesse contexto, *Fog Computing* pode ser

implantado como uma camada de filtragem regional de dados, diminuir a carga e a largura de banda utilizadas entre os sensores e a nuvem;

- Transportes Inteligentes: o conceito de Sistemas de Transporte Inteligentes (do Inglês, *Intelligent Transportation Systems* (ITS)) é um dos principais componentes das Cidades Inteligentes, que foi introduzido para melhorar a gestão de tráfego e melhorar a segurança nas ruas e rodovias. Existe também a chamada Internet de Veículos (do Inglês, *Internet of Vehicles* (IoV)), que consiste em veículos com diversos sensores conectados de forma permanente na Internet que coletam dados tais como trânsito e condição das rodovias e encaminham para as aplicações ITS. A quantidade de dados coletados será massiva devido ao aumento da quantidade de veículos conectados, o que impõe a necessidade da adoção de análise baseadas em *Big Data*. Nesse sentido, como já foi mencionado anteriormente, a nuvem possui a desvantagem de ter latência alta, então o processamento de grandes quantidades de dados para ITS na nuvem se torna um desafio, levando em conta a necessidade de uma resposta rápida aos veículos para garantir segurança [5]. Ademais, pensando em um futuro onde carros autônomos serão o padrão, fica clara a necessidade de se implantar uma rede de *Fog Computing* bem projetada e robusta para garantir o bom funcionamento e a segurança desses novos meios de transporte;
- Distribuição de Multimídia e CDNs: como foi dito anteriormente, na discussão sobre diminuição de latência, é possível implantar uma arquitetura *Fog* para intermediar e realizar *cache* dos dados mais utilizados na nuvem, assim sendo, segundo Haouari et. al. [5], existem estudos que propõem a utilização de uma plataforma baseada em *Fog* que pode fornecer recursos computacionais, principalmente armazenamento, com baixo custo e baixa latência para distribuição de mídia. Na verdade, esse conceito é exatamente o conceito implantado por grandes empresas que implementam CDNs, tais como Netflix, Google e CloudFlare [12];
- Monitoramento Remoto: outro caso de uso interessante de *Fog Computing* está relacionado com o monitoramento remoto por meio de circuitos fechados de televisão (câmeras de vigilância), aplicando-se algoritmos de inteligência artificial e pós-processamento para detecção de possíveis ameaças [5], caracterização de objetos, identificação de pessoas. Nesse contexto, é possível implantar processamento de vídeo na camada *Fog* a fim de conter dados de gravação do cotidiano em regiões próximas, enviando pra nuvem apenas os eventos mais importantes;
- Setor de Saúde: a inclusão de tecnologias da Internet das Coisas no setor de saúde garante uma crescente melhora no custo e na qualidade do atendimento médico por meio da automação de tarefas que eram anteriormente realizadas por humanos.

Além disso, essa inclusão permite monitorar pacientes de forma remota, por meio de sistemas inteligentes de monitoramento de saúde [5]. Assim como os outros casos citados, a inclusão desses dispositivos de monitoramento geram imensas quantidades de dados, que necessariamente precisam ser processados em algum lugar e, dependendo da criticidade do tipo de monitoramento, por exemplo, em uma UTI, caso algum parâmetro saia do controle, é preciso acionar a equipe médica para tomar providências o mais rápido possível, ou seja, se trata de uma aplicação que depende em baixa latência. Assim sendo, é também possível implantar a camada de *Fog Computing* para processamento e filtragem desses inúmeros dados gerados;

3.2 Desafios para Implantação de *Fog Computing*

Antes de comentar a respeito da proposta de implantação prática de *Fog Computing*, é conveniente dissertar mais a respeito dos desafios encontrados para consolidação de uma abordagem que respeite todas as características e vantagens apresentadas sobre o tema. Infelizmente não é fácil encontrar na literatura apanhados objetivos sobre quais desafios são enfrentados ao se construir uma infraestrutura de *Fog Computing* na prática. Assim, será feita uma breve reflexão sobre as possíveis dificuldades que poderiam surgir ao se implantar um sistema de *Fog* de forma efetiva. Vale ressaltar que, nessa seção, indo contra às orientações convencionais de um trabalho científico, muitas das informações trazidas são de vivências e conhecimentos práticos adquiridos pelo autor, Samuel James, durante seus anos de trabalho como arquiteto de redes em um provedor regional na cidade de Belo Horizonte, arquiteto de redes de uma grande empresa hospedagem de servidores de jogos no Brasil, arquiteto de redes de um *datacenter* de pequeno porte em São Paulo, além de um projeto pessoal de hospedagem de servidores, entre os anos de 2020 e 2022. Vale ainda comentar que o autor trabalha na área de desenvolvimento e hospedagem de serviços online desde o ano de 2013, além de ter certificação *Cisco Certified Entry Networking Technician* (CCENT), que cobre diversos conhecimentos sobre redes, configuração de roteadores, funcionamento dos protocolos da Internet, funcionamento de *datacenters* e muito mais. Por fim, é notória a experiência prática do autor com tecnologias do mundo da Internet das Coisas, hoje sendo líder de uma equipe de desenvolvimento de *Firmware* para dispositivos IoT.

Para entender quais desafios surgem para implantação de *Fog Computing*, é interessante realizar uma comparação sobre o modelo centralizado das nuvens com o modelo descentralizado de *Fog*. Nesse sentido, a construção dos projetos de *Cloud Computing* sempre esteve voltada para a centralização e concentração de recursos, isto é, na construção de *datacenters* muito grandes e muito poderosos, capazes de prover elasticidade e

disponibilidade ilusória de recursos infinitos para o usuário. Isso, por um lado, garante um excelente aproveitamento de recursos e facilita diversas questões, principalmente no que se diz respeito a logística, segurança, contração de links de Internet, construção de infraestrutura, fornecimento de energia elétrica redundante, etc. Por outro lado, nessa abordagem, surgem as já comentadas limitações de latência devido à grande distância aos usuários. Indo na direção contrária, os projetos de *Fog Computing*, por sua natureza distribuída, devem vencer inúmeros desafios e lidar com questões que comumente não são enfrentadas nos projetos dos grandes centros de processamento da nuvem. Tais desafios podem incluir, mas não se limitam a: Construir *datacenters* regionais com custo-benefício efetivo; Garantir boa conectividade de rede; Contratação de pessoal qualificado (nas regiões diversas) para trabalhar nos *datacenters*; Garantir a segurança e a privacidade dos dados[13], seja fisicamente por sistemas de controle de acesso, seja virtualmente por meio da implantação de firewalls e outros métodos de segurança da informação; Realizar investimento em infraestrutura elétrica redundante, seja por meio de geradores à gasolina/diesel, seja por bancos de bateria e placas solares; Realizar investimento em sistemas de resfriamento com eficiência energética razoável; Entre outros.

Outrossim, durante a realização deste projeto de conclusão de curso, entre os meses de março e maio de 2023, foram feitas pesquisas com o intuito de se encontrar exemplos concretos de *Fog Computing* no mercado das grandes provedoras de nuvem. Nessa ocasião, foram encontradas plataformas como *Google Distributed Cloud*, *AWS For the Edge*, *AWS Local Zones* e *Azure Edge Computing*. Apesar de todas essas plataformas terem investimento por parte dos maiores fornecedores de serviços de nuvem do mundo, percebeu-se uma limitação grande no quesito de distribuição geográfica, principalmente quando se diz respeito a países menos desenvolvidos como o Brasil. Isso mostra que, apesar desses grandes fornecedores conseguirem investir em infraestruturas centralizadas nas principais capitais do mundo, existe um vácuo enorme quando se fala em cidades menores. Nesse último caso, o que tem-se visto efetivo no mercado são as parcerias feitas entre provedores regionais e grandes empresas de mídia, como a Netflix, para instalar alguns servidores de cache local para servir os clientes regionais - o que já poderia ser visto como uma aplicação prática de *Fog Computing*, uma vez que se provê conteúdo em menor latência, economiza-se banda e tráfego com as centrais de nuvem e, conseqüentemente, traz uma redução enorme de custos, seja para os provedores locais, seja para própria Netflix, que reduz seus custos de saída de dados da nuvem.

Em suma, existe ainda um longo caminho a ser trilhado e explorado para trazer à vida a tão sonhada e pesquisada *Fog Computing* como uma tecnologia competitiva de mercado. Nesse contexto, é inevitável que, para que uma tecnologia atinga o seu real potencial - principalmente na área de Computação - seja necessário forte investimento do setor

privado, visando competitividade de mercado, utilização rentável de recursos e difusão global por meio da prestação de serviços que utilizam dessa tecnologia. Portanto, é por meio da solução desses desafios que surge a oportunidade de investimento na construção de infraestrutura para suportar o futuro da Internet das Coisas, Sistemas de Transportes Inteligentes, Cidades Inteligentes, etc. Dando continuidade ao tema, na próxima seção será apresentado um problema que deve obrigatoriamente ser resolvido para que *Fog Computing* possa entregar a tão prometida baixa latência e se tornar uma tecnologia difundida no mercado.

3.3 Contradições de Latência nas Rotas da Internet

Um dos maiores desafios para difusão da camada *Fog*, sob o ponto de vista do autor, é solucionar as contradições existentes na latência das rotas da Internet. Para entender tais contradições, será feita uma análise baseada no seguinte cenário: um usuário final (U) se localiza atualmente em uma cidade (X) e possui uma conexão de Internet de um certo provedor (P). Na cidade X, está instalado um *datacenter* que provê serviços de *Fog Computing* (F). Em outra cidade (Y), localizada a 500km de X, estão instalados grandes *datacenters* que prestam serviço para uma plataforma de *Cloud Computing* (C). Utilizando recursos tanto em F quanto em C, foi implantada uma aplicação (A) cujo algoritmo sempre escolhe os servidores com a menor latência para se conectar.

No cenário descrito, o usuário (U) abre a aplicação (A) em seu computador. Nesse caso, a aplicação irá se conectar com os servidores de *Fog* (F) ou de *Cloud* (C)? A resposta mais lógica, visto que a aplicação escolhe sempre os servidores com a menor latência, seria a conexão com os servidores F, uma vez que estão na mesma cidade (X) que o usuário, uma distância bem menor que 500km. Entretanto, para responder a essa pergunta é necessário saber, além da distância física entre servidor e usuário, a distância lógica entre ambos, isto é, quais as rotas utilizadas para trafegar pacotes entre U e F, entre U e C e também o caminho inverso, uma vez que as rotas da Internet nem sempre refletem os caminhos de menor distância física. Portanto, suponhamos que:

- A cidade X, onde está o provedor regional e os *Fog Nodes*, seja Belo Horizonte;
- A cidade Y seja São Paulo, onde estão localizados os maiores *datacenters* de nuvem do Brasil, inclusive o nosso principal Ponto de Troca de Tráfego (PTT), o Ponto de Troca de Tráfego de São Paulo (PTT-SP).

Nesse cenário, existem duas situações muito prováveis:

- Situação A: O provedor (P) de Internet do usuário (U) tem rotas de conexão direta com *datacenter* de *Fog* (F), isto é, fazem *Local Peering*. Nesse caso, se a rota estiver

disponível e não estiver congestionada, a latência entre U e F será próxima a 1ms, variando geralmente entre 1ms e 5ms para conexões cabeadas.

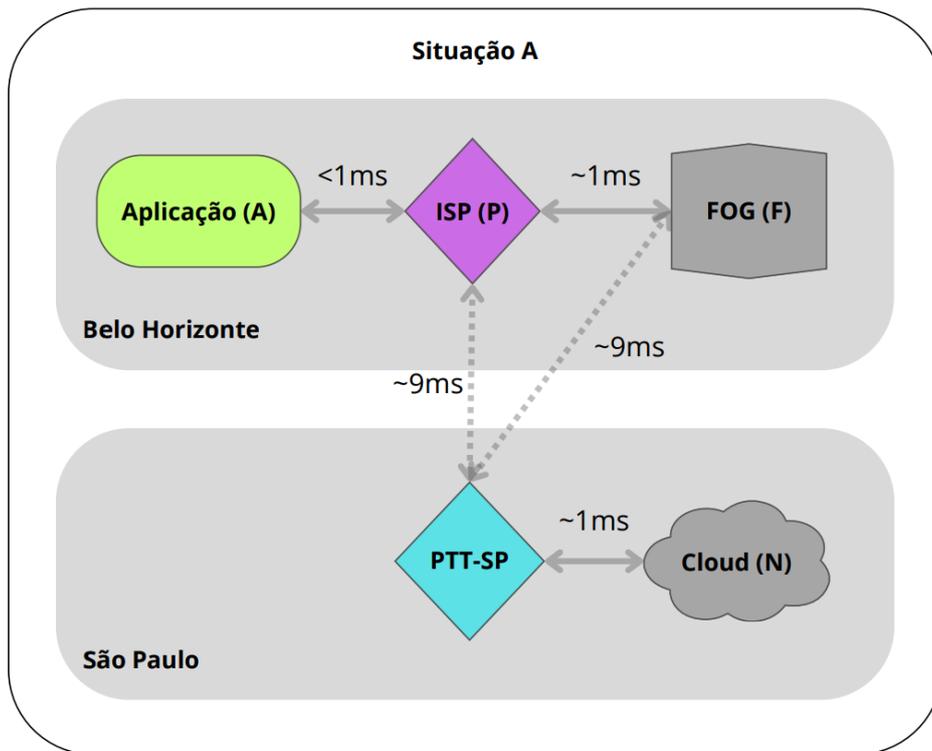


Figura 3.1: Ilustração, feita pelo autor, para exemplificar o cenário disposto na Situação A.

- Situação B: O provedor (P) de Internet do usuário (U) não tem rotas diretas ao *datacenter* de *Fog* (F), mas tem rotas diretas para o PTT-SP, ou seja, fazem *Remote Peering*. Assumir também que o *datacenter* de *Fog* também possui rotas diretas para o PTT-SP. Nesse caso, a latência entre U e F poderá ser de cerca de 12ms a 30ms, a depender da latência entre Belo Horizonte e São Paulo, que varia em média de 6ms a 15ms. Para simplificar, supor que latência entre U e F seja de 18ms nesse caso.

Para continuar a análise, assumir também que a latência entre Usuário (U) e a nuvem (C) em São Paulo é de 10ms, conforme foi exemplificado na Figura 3.1 e na Figura 3.2.

Finalmente, após todos os dados apresentados, fica claro que a resposta pro questionamento levantado depende diretamente das rotas entre U e F e entre U e C. Por um lado, na situação A, a aplicação irá se conectar aos servidores *Fog*, pois a latência de U para F de 1ms é muito menor que U para C de 10ms. Por outro lado, na situação B, a aplicação irá preferir se conectar diretamente à nuvem, mesmo estando fisicamente mais distante que a *Fog*, já que a latência entre U para F é maior que de U para N, por conta da inexistência de rotas diretas entre U e F.

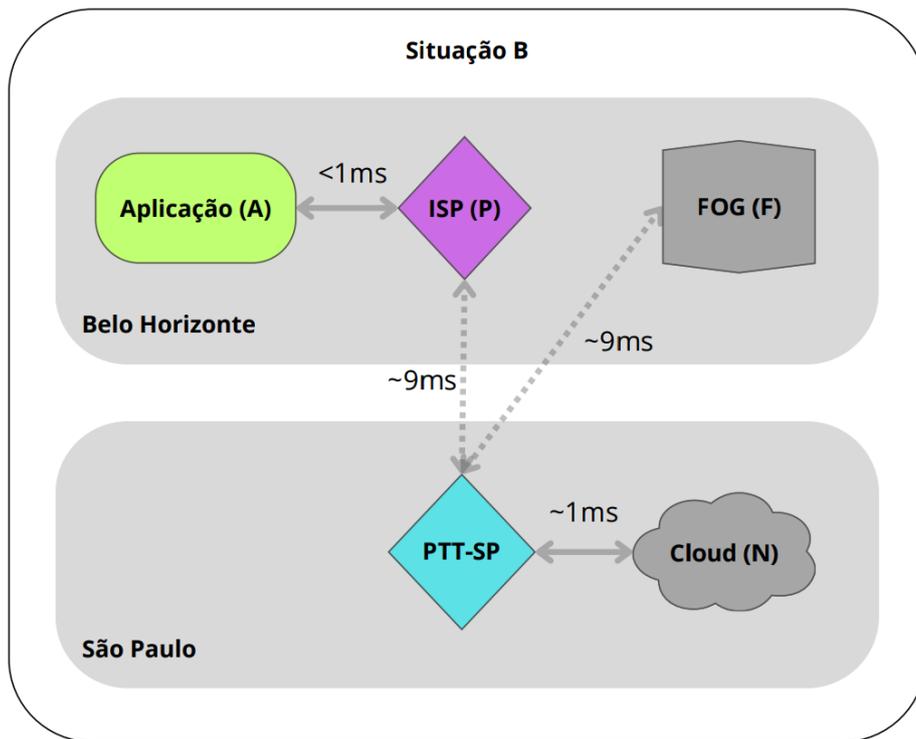


Figura 3.2: Ilustração, feita pelo autor, para exemplificar o cenário disposto na Situação B.

Infelizmente a situação de inconsistência de latência apresentada é muito comum, inclusive no Brasil, onde a maioria dos provedores regionais opta por contratar conexões de transporte direta ao nosso principal PTT, o PTT-SP. Essa situação pode ser constatada de forma prática pelo artigo escrito por Mazzola (2022) [10], onde encontrou-se que, em uma situação onde há PTTs, também conhecidos interacionalmente pelo nome *Internet Exchange Point* (IXP), boa parte das rotas são do tipo *Remote Peering*, e, ainda segundo o artigo, o protocolo *Border Gateway Protocol* (BGP), principal protocolo de roteamento utilizado na Internet, nem sempre escolhe as rotas de menor latência.

Assim sendo, para a implantação efetiva de *Fog Computing*, fica clara a necessidade de se trabalhar em políticas de *peering* entre ISPs regionais e *datacenters* de *Fog* a fim de viabilizar e otimizar as rotas de tráfego no Brasil e em toda a Internet.

Em suma, este capítulo trouxe um apanhado de diversos casos de uso das tecnologias de *Fog Computing* e Internet das Coisas, incluindo limitações existentes e desafios da implantação dessas tecnologias no contexto da atualidade. Para dar continuidade ao tema, o próximo capítulo apresentará uma proposta de implementação prática dessas tecnologias idealizada pelo autor.

Capítulo 4

Proposta e Implementação

Com o constante crescimento dos centros urbanos e conseqüente aumento da quantidade de veículos nas rodovias, é notável a necessidade de investimento em tecnologias que aprimorem os meios de transporte e visem solucionar problemas relacionados à locomoção urbana. Nesse contexto, para que haja uma melhora significativa no fluxo de veículos, principalmente nos horários de pico e nas avenidas mais movimentadas, provavelmente seja necessário priorizar o fluxo coletivo em detrimento do individual. Para tal, é necessário que exista um ente controlador central que possa tomar decisões com base no contexto geral, não somente com base nas condições locais, como é o caso do controle individual que cada motorista pode exercer em seu veículo.

Nessa abordagem, seria possível implementar algoritmos de otimização de tráfego que tornem o trânsito cada vez mais fluido e organizado, pois a decisão e organização não dependeria da decisão individual de cada motorista, mas levaria em conta inúmeras variáveis que somente um sistema computadorizado conseguiria administrar. Nesse contexto, surge, portanto, a ideia deste trabalho, denominado, pelo autor, por Modo Colônia, que consiste justamente em um sistema de controle de tráfego centralizado que permite o controle do fluxo de veículos de forma organizada e em grupo. A ideia para o nome Colônia vem das Colônias de insetos, como formigas ou abelhas, uma vez que esses seres vivem em uma sociedade organizada e, de alguma forma, conseguem se comunicar de forma a colaborar como um grupo.

Esse trabalho traz uma proposta de implementação desse conceito com o uso de tecnologias de *Fog Computing* e IoT no contexto de uma Cidade Inteligente, por meio do desenvolvimento da simulação de um sistema de controle de tráfego coletivo com processamento de dados em baixa latência. Mais especificamente, será apresentada uma proposta de sistema de controle de tráfego para um cruzamento de trânsito, cujos carros que trafegam nesse cruzamento são autônomos e estão conectados em rede. A camada de *Fog Computing* é essencial nesse contexto devido à necessidade de se ter tempo de resposta

rápido, estabilidade de conexão e também devido ao grande volume de dados gerados pelos dispositivos IoT, os quais, neste exemplo são: carros autônomos e demais sensores distribuídos ao longo das rodovias.

4.1 Visão Geral

Primeiramente, vale ressaltar que esse trabalho tem foco na parte de infraestrutura computacional e aplicação de conceitos de *Fog Computing* no controle de trânsito, não tendo grande rigorosidade quanto aos termos técnicos da Engenharia de Trânsito e áreas relacionadas, assim sendo, podem haver termos com definição informal mas que são facilmente entendidos pelo uso cotidiano do público geral. Além disso, as soluções aqui apresentadas estão baseadas em um provável cenário futuro, onde:

- Semáforos serão dispositivos IoT, terão a capacidade de se conectar a um sistema de controle centralizado e poderão ter seu comportamento alterado de acordo com a situação do trânsito, podendo atuar no modo de controle de tráfego em colônia ou no modo de controle de tráfego legado;
- Existirá uma infraestrutura de suporte de tráfego baseada em IA para coleta e processamento de vídeo em tempo real e dados de diversos sensores instalados nas rodovias. Essa infraestrutura estará conectada ao sistema de controle de tráfego e servirá de base de tomada de decisões e também como forma de se garantir a segurança da via;
- Veículos autônomos com conectividade de rede serão a maioria. Esses veículos possuirão a capacidade de detectar diversos tipos de objetos e coletar diversas informações a respeito do trânsito na rodovia, além de estarem configurados para receber controle da central de trânsito. Para o funcionamento correto do controle de tráfego, os veículos deverão operar em modo autônomo e não permitirão interação forte do usuário, isto é, toda a parte de tomada de decisão a respeito de rotas, escolha de velocidade, decisão de quais faixas de trânsito trafegar, manutenção da distância entre veículos, etc, serão controlados pelo próprio veículo em contato com a central de controle. Portanto, fica definido como carro autônomo um veículo que possui todas essas características de operação via sistema computadorizado, assim, por exclusão, carro convencional fica definido como sendo um veículo que não possui alguma dessas características apresentadas.
- A conectividade de rede móvel nas rodovias terá a qualidade e confiabilidade necessárias para o funcionamento do sistema de controle de tráfego. Assume-se também

que o problema da contradição de rotas na Internet, apresentado no capítulo anterior, já foi solucionado na região onde está sendo implantado o sistema de controle de tráfego. Nesse contexto, vale ainda ressaltar que será necessário uma infraestrutura de *Fog* regional para atender às demandas de trânsito regional.

Será discorrido a seguir a respeito de como foi feita a implementação do projeto, descrevendo detalhes da aplicação desenvolvida, sua lógica de funcionamento, entidades simuladas e a classificação de cada módulo dentro da arquitetura de três camadas do modelo de *Fog Computing*.

4.2 Estrutura da Aplicação Desenvolvida

A aplicação desenvolvida possui código fonte aberto e está disponível para consulta de forma pública e gratuita por meio da página do repositório de projeto no *GitHub*. A aplicação idealizada e desenvolvida se trata de uma simulação de um cruzamento de trânsito, onde trafegam veículos autônomos conectados a um Sistema de Controle de Tráfego (SCT) via rede. A aplicação possui 2 módulos principais, são eles:

- Ambiente de Simulação: desenvolvido na linguagem de programação *JavaScript*, utilizando a biblioteca gráfica de código aberto *p5.js* [14], podendo ser reproduzida em qualquer *Browser* de Internet moderno. Este módulo, que encontra-se no diretório *sim/* do repositório, simula o movimento dos veículos autônomos trafegando nas rodovias, por meio de equações de movimento aplicadas ao longo do tempo. A simulação ocorre em tempo real, por meio de laços de repetição que são executados a cada *frame* renderizado. O Ambiente de Simulação conta com diversas entidades, tais como carro, pista, curva e semáforo. Cada carro e cada semáforo possui também integração com um módulo de conexão de rede que troca mensagens com o SCT para operação em Modo Colônia. Dentro da arquitetura de três camadas do modelo *Fog*, o ambiente de simulação representa a camada IoT, por meio dos carros e semáforos conectados na rede;
- SCT: desenvolvido também na linguagem de programação *JavaScript*, utilizando a biblioteca de código aberto *Node.js*. Este módulo, que encontra-se no diretório *core/*, possui a capacidade de receber conexões dos semáforos e dos veículos, realizando o processamento dos dados e distribuindo a tomada de decisão a respeito do movimento de cada um dos veículos. Dentro da arquitetura de três camadas do modelo *Fog*, o SCT é executado nas camadas *Fog* e *Cloud*, levando em conta características de latência e posicionamento geográfico.

4.3 Ambiente de Simulação

O ponto de partida da simulação está no arquivo `index.html`, que carrega todos os arquivos necessários para funcionamento do ambiente `p5.js` e da simulação. O ambiente `p5.js` funciona por meio da chamada de duas rotinas, uma de inicialização - a rotina `setup()` - e uma de renderização - rotina `draw()`. A implementação dessas rotinas estão no arquivo `sketch.js`.

Primeiramente, a rotina de inicialização é responsável por carregar as entidades e preparar o ambiente de simulação com todas as variáveis necessárias para o seu funcionamento. Essa rotina, por sua vez, além de preparar a janela de renderização dentro do documento, chama também a rotina de carregamento `loadSimulation()` do arquivo `loader.js`. A rotina `loadSimulation()` possui a configuração e o posicionamento inicial de todas as entidades simuladas, onde cada entidade possui um identificador único (`id`), gerado sequencialmente e um tipo (`type`).

Logo após o carregamento bem sucedido da simulação, o ambiente `p5.js` chama a rotina de renderização em um laço de repetição contínuo. Essa rotina é responsável por desenhar na tela o estado atual da simulação a cada *frame*, para tal, todas as entidades simuladas possuem uma rotina específica de renderização (`display()`) que é chamada em ordem pela rotina de renderização principal. Além disso, para que a renderização aconteça de forma correta, todas as entidades possuem uma posição vetorial bidimensional, variáveis ponto flutuante `x` e `y`, geralmente agrupadas em uma variável `position`. A renderização utilizada neste trabalho acontece em um plano bidimensional, representando uma vista de cima para baixo da Cidade Inteligente.

A velocidade de renderização, isto é, a quantidade de *frames* renderizados por segundo, pode variar de acordo com a quantidade de elementos sendo renderizados ou de acordo com a performance do *Hardware* utilizado. Para amenizar essas diferenças e para garantir uma simulação mais estável, toda a lógica de atualização de posicionamento da simulação é feita com base em equações matemáticas iterativas, utilizando a variável tempo em milissegundos como parâmetro. Dessa forma, a quantidade de *Frames* por Segundo (FPS) não afeta a simulação de forma significativa. O ambiente `p5.js` fornece uma configuração de FPS desejada através da rotina `frameRate()`, neste caso, foi escolhido o valor de 60 FPS (`frameRate(60)`), taxa de quadros comum em monitores da atualidade e permite uma visualização fluida do movimento, desde que o *Hardware* suporte.

4.3.1 Ciclos de Gerenciamento e Constantes de Simulação

Além das rotinas padrão do ambiente `p5.js`, foram implementadas também rotinas auxiliares, denominadas ciclos de gerenciamento de entidade. Um ciclo de gerenciamento de en-

tidade é uma rotina executada nas entidades mais complexas, tais como carro e semáforo, uma vez que precisam executar uma lógica de forma independente à renderização. Para garantir esse desacoplamento, a rotina de gerenciamento é executada em uma *Thread* separada para cada entidade e, para isso, foi utilizada a função `setTimeout(fun, timeout)` da linguagem de programação *JavaScript*, na qual é passada uma função (`fun`) para ser executada de forma assíncrona, e um tempo de espera (`timeout`) para que se inicie a execução. Quando é omitido o tempo de espera, o mesmo é pré-definido como 0 e a rotina se inicia de forma imediata. A rotina passada para o ciclo de gerenciamento das entidades carro e semáforo possuem um laço de repetição *while* que é executado a cada unidade de tempo de simulação, conforme trecho de código da Figura 4.1.

```

startThread() {
  this.thread = setTimeout(async ()=>{
    while (1) {
      await this.manage();
      await sleep(SIMULATION_TICK_PERIOD);
      this.time += SIMULATION_TICK_PERIOD;
    }
  });
}

```

Figura 4.1: Trecho de código utilizado para iniciar as *Threads* de ciclo de gerenciamento.

```

const STREET_WIDTH = 30; // 3.0 m
const STREET_LENGTH = 500; //50.0 m
const CAR_WIDTH = 19; // 1.9 m
const CAR_LENGTH = 35; // 3.5 m
const CAR_MAX_SPEED = kmh_dms(60); // 60 km/h
const CAR_TURN_SPEED = kmh_dms(30); // 30 km/h
const CAR_ACCEL = 25; // 2.5 m/s^2
const CAR_BRAKE = 70; // 7.0 m/s^2

const SIMULATION_TICK_PERIOD = 25; // 25 ms
const CROSS_MAX_LATENCY = 4 * SIMULATION_TICK_PERIOD; // 100 ms

```

Figura 4.2: Trecho de código onde são feitas as definições de importantes constantes da simulação.

Ainda na Figura 4.1, pode ser visto o uso da unidade de período de tempo de simulação (`SIMULATION_TICK_PERIOD`), que foi definida como sendo 25 milissegundos. Assim, a cada 25 milissegundos, toda a lógica de gerenciamento dos carros e do semáforo é executada novamente por meio da chamada (`this.manage()`).

Além da unidade de tempo de simulação, foram definidas algumas outras constantes importantes, conforme pode ser visto no trecho de código da Figura 4.2, sendo:

- **STREET_WIDTH**: representa a largura padrão de uma faixa de rodagem de uma rodovia. É dado na unidade de decímetros (um décimo de metro). O valor padrão utilizado nesta simulação é de três metros (3m), ou trinta decímetros (30dm);
- **STREET_LENGTH**: representa o comprimento padrão de uma faixa de rodagem de uma rodovia, também dado decímetros, definido inicialmente como cinquenta metros (50m), ou quinhentos decímetros (500dm);
- **CAR_WIDTH**: representa a largura padrão de um carro, também dado decímetros, definido inicialmente como dezenove decímetros (19dm = 1.9m);
- **CAR_LENGTH**: representa o comprimento padrão de um carro, também dado decímetros, definido inicialmente como trinta e cinco decímetros (35dm = 3.5m);
- **CAR_MAX_SPEED**: representa a velocidade máxima dos veículos nas faixas de rodagem, definido inicialmente como sessenta quilômetros por hora (60km/h). Pode-se observar a chamada de uma rotina auxiliar `kmh_dms()`, responsável por fazer a conversão de unidade de quilômetros por hora para decímetros por segundo, unidade padrão da simulação;
- **CAR_TURN_SPEED**: representa a velocidade máxima dos veículos nas curvas, definido inicialmente como trinta quilômetros por hora (30km/h), por meio da rotina `kmh_dms()`;
- **CAR_ACCEL**: representa a aceleração de um carro, em decímetros por segundo por segundo ($\frac{dm}{s^2}$). O valor padrão escolhido foi de $2.5 \frac{m}{s^2} = 25 \frac{dm}{s^2}$;
- **CAR_BRAKE**: representa a aceleração de frenagem de um carro, também em decímetros por segundo por segundo. O valor padrão escolhido foi de $7.5 \frac{m}{s^2} = 75 \frac{dm}{s^2}$;
- **CROSS_MAX_LATENCY**: representa o tempo de resposta máximo permitido para que um cruzamento continue operando no Modo Colônia, conforme será detalhado adiante.

Todos os valores utilizados nessas constantes foram escolhidos de forma a se aproximar bastante da realidade, porém, sem muito rigor teórico, uma vez que o enfoque deste projeto é a parte computacional e não a criação de uma simulação totalmente fiel à realidade. Ainda assim, esses valores podem ser facilmente modificados conforme for desejado, a fim de se criar diferentes cenários para a simulação, podendo ainda servir como objeto de pesquisas futuras na área.

4.3.2 Entidades Simuladas

Nessa simulação existem algumas entidades que necessitam de uma localização e orientação espacial, para tal, utilizam uma Posição Alinhada (classe `AllignedPosition` em `sim/utills.js`). Essa estrutura consiste em uma grandeza vetorial de três componentes: posição (\mathbf{x}) ao longo do *eixo* x , em decímetros; posição (\mathbf{y}) ao longo do *eixo* y , em decímetros; ângulo em radianos (`dir`) em relação à direção positiva do *eixo* x . Esse ângulo representa, por exemplo, qual é a direção do movimento de um veículo, ou ainda qual é a direção correta de movimentação dentro de uma faixa de rodagem. A Posição Alinhada possui também alguns métodos úteis para cálculos vetoriais e é a base de posicionamento dentro da simulação. Assim, as principais entidades da simulação são as seguintes:

- Pista (`sim/entity/street.js`) e Curva (`sim/entity/turn.js`): uma Pista, da forma aqui modelada, representa uma faixa de rodagem de trânsito de veículos totalmente plana e em linha reta, onde os carros podem trafegar em velocidade máxima. A Curva, por sua vez, representa um trecho da faixa de rodagem em ângulo, portanto, a velocidade em que os carros trafegam é mais baixa. Tanto a Pista quanto a Curva possuem características de interface semelhantes, principalmente:
 - Uma Posição Alinhada de início da faixa de rodagem (`getStart()`) e uma Posição Alinhada final (`getEnd()`);
 - Uma função paramétrica (`path(t, calculateDir)`) que retorna todas as posições contidas dentro do trecho da faixa de rodagem. Isso significa que dada uma entrada $t \in [0, 1]$, essa função retorna uma Posição Alinhada que está entre a posição inicial e a posição final, sendo $t = 0$ para a posição inicial e $t = 1$ para a posição final. A segunda entrada da função, `calculateDir`, indica se deve ser feito o cálculo da direção da Posição Alinhada, quando a entrada é verdadeira, ou se apenas a posição cartesiana (x, y) basta, quando a entrada é falsa. A função paramétrica é importantíssima para que o movimento dos carros aconteça de forma correta nos trechos em curva, além de acontecerem em sentido correto.
- Cruzamento (`sim/entity/crossroad.js`): um cruzamento é um tipo de compartilhamento de espaço bidimensional de vias que tem direções diferentes, geralmente perpendiculares, sendo bastante implementado em locais onde não há espaço ou orçamento suficiente para construção de uma solução tridimensional, tal como um viaduto. Dessa forma, é de extrema importância que um projeto de controle de tráfego para Cidades Inteligentes contenha uma boa implementação de controle de cruzamentos. Nesta simulação, o cruzamento representa um trecho crítico de faixa

de rodagem que necessita ter seu fluxo controlado por meio de um semáforo, quando opera em Modo Legado, ou por meio do Sistema de Controle de Tráfego, quando opera em Modo Colônia. Os modos de operação legado e colônia serão explicados adiante;

- Carro (`sim/entity/car.js`): representa um veículo autônomo com capacidade de se movimentar ao longo de faixas de rodagem de forma independente. A entidade carro possui, além da sua posição, uma variável que representa a sua velocidade. Essa velocidade é utilizada para atualizar a posição do carro a cada *frame* renderizado, ou a cada ciclo de gerenciamento. A operação detalhada do carro será explicada adiante;

4.4 Funcionamento do Modo Colônia

O ponto de interesse deste projeto consiste em aprimorar a mobilidade urbana em Cidades Inteligentes. Para tal, é interessante focar em resolver problemas nos pontos críticos do sistema de trânsito, dentre os quais está o cruzamento, uma vez que, em um cruzamento, é necessário que apenas uma direção de tráfego passe por vez, o que gera tempo de espera e bastante desperdício energético - já que os veículos geralmente ficam com o motor ligado esperando o semáforo abrir, sem falar que, em horários de pico, trechos com cruzamento comumente ficam totalmente congestionados. Nesse contexto, surge a ideia de se aplicar um sistema de controle centralizado para o cruzamento, possuindo um modo de operação colônia, conforme será discutido adiante.

O Modo Legado de funcionamento, no contexto desse projeto, representa o bem conhecido funcionamento de um semáforo padrão, contendo as colorações de controle para Pare (Vermelho), Atenção (Amarelo) e Siga (Verde). O Modo Legado de funcionamento pode funcionar para qualquer tipo de veículo, seja um carro autônomo, seja um carro convencional. A grande limitação desse modo de funcionamento é o fato de que apenas uma direção de tráfego pode transitar no trecho crítico por vez e, para isso, é implementado um controle de temporização para cada uma das colorações de controle. Esse controle está implementado na rotina `legacyOperation()` da classe Cruzamento (`CrossRoad` em `sim/entity/crossroad.js`).

O Modo Colônia de funcionamento representa uma situação onde todos os veículos localizados nas faixas de rodagem de entrada do cruzamento, inclusive o próprio controlador de semáforo do cruzamento, estão conectados a um Sistema de Controle de Tráfego (SCT), em uma conexão estável e de baixa latência. Nesse modo de funcionamento, cada veículo recebe informações em tempo real da central, incluindo qual velocidade deve trafegar e se está autorizado a passar no cruzamento ou não, no momento correto. Assim,

é de total responsabilidade do SCT garantir que não haja nenhuma colisão entre veículos que se aproximam do cruzamento, além de garantir um fluxo otimizado de veículos.

A fim de integrar o sistema de controle de tráfego a parte da infraestrutura de semáforos já existente, fica proposto a utilização da cor azul para indicar que o sistema de controle de tráfego está ativo e operante em Modo Colônia. Por exemplo, para indicar que um cruzamento está operando em Modo Colônia, será utilizada uma quarta cor no semáforo, conforme está ilustrado na Figura 4.3.



Figura 4.3: Ilustração da quarta cor utilizada no semáforo para indicar o Modo Colônia, a cor azul.

Sempre que a luz azul estiver acesa em um trecho da rodovia, apenas veículos autônomos conectados à central estarão autorizados a trafegar naquele trecho, sendo estritamente proibido o tráfego de um veículo comum. Além disso, a luz azul indicará que todo o sistema de controle de tráfego autônomo está operando corretamente naquela região, indica também que todos os carros detectados por meio dos sensores e câmeras estão conectados à central, em conexão estável e com latência dentro dos limites aceitáveis. Durante a operação em Modo Colônia, caso aconteça qualquer distúrbio, tal como a desconexão repentina de um veículo ou do semáforo com a central, o sistema automaticamente realizará *fallback* para o modo de operação legado.

A Figura 4.4 ilustra um cruzamento que está trabalhando no Modo Colônia, as barras azuis nas entradas do cruzamento indicam a cor atual do semáforo. Na Figura, é possível observar que o veículo (na cor azul) que está trafegando na direção norte acabou de passar pelo cruzamento e se move a mais de $40km/h$ e, ainda assim, dois veículos que se aproximam na pistas de direção leste/oeste já receberam autorização para atravessar o cruzamento, sem necessidade de parar e esperar no semáforo vermelho. Isso tudo só foi possível porque estavam conectados ao SCT, que coordenou toda a operação dentro do Modo Colônia.

Em contrapartida, a Figura 4.5 ilustra um cruzamento que está trabalhando no Modo Legado, isto é, utilizando o controle de tráfego baseado nas cores verde, amarelo e vermelho. Nesse cenário, é possível visualizar dois veículos, nas pistas leste/oeste, aguardando no sinal vermelho, ao passo em que os veículos das pistas norte/sul acabaram de iniciar a passagem, ainda estando em baixa velocidade, apenas $14km/h$.

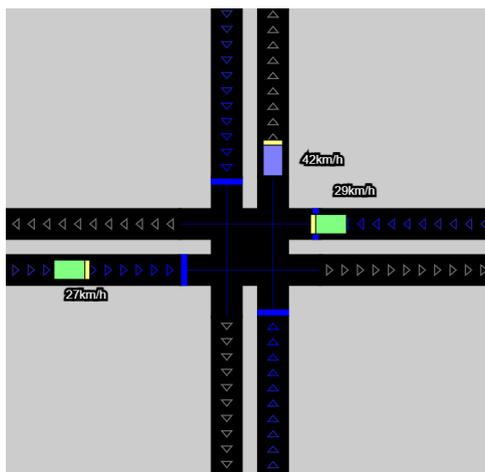


Figura 4.4: *Screenshot* da simulação do cruzamento operando em Modo Colônia.

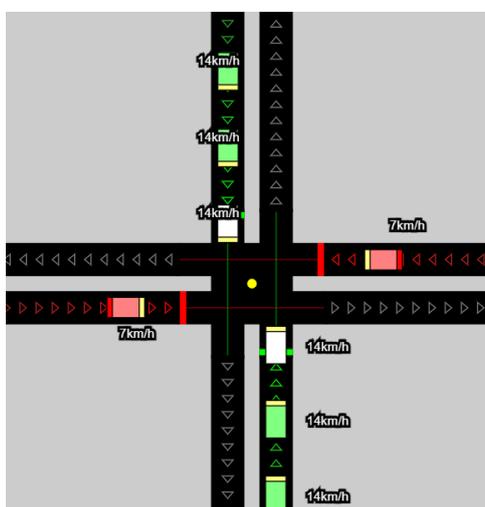


Figura 4.5: *Screenshot* da simulação do cruzamento operando em Modo Legado.

4.4.1 Operação dos Veículos Autônomos

De forma simétrica aos modos de operação de um cruzamento, um veículo autônomo pode também operar em Modo Legado ou em Modo Colônia. Sendo assim, no contexto do veículo, o Modo Legado representa que está trabalhando apenas com base em seus próprios sensores e processamento de vídeo de câmeras instaladas no veículo, de forma desacoplada e desconectada ao SCT. Nessa operação, assume-se que o veículo é capaz de detectar outros veículos em trechos de rodagem a sua frente, regulando sua velocidade de forma a não colidir. Assume-se também que o veículo é capaz de visualizar as colorações do semáforo e tomar decisões de cruzar ou não com base nas cores.

Para auxiliar na visualização do funcionamento da tomada de decisão por parte dos

veículos e por parte do SCT, foi criado um código de cores que indica o estado atual de um veículo:

- Modo Legado, Veículo Desconectado, Cor Branca: indica que um veículo ainda não conseguiu realizar a conexão com o SCT e, portanto, trabalha em Modo Legado;
- Modo Colônia, Veículo Conectado, Cor Azul: indica que um veículo está conectado de forma estável e confiável ao SCT, recebendo todas as informações de controle;
- Cor Verde: indica que um veículo tomou a decisão ou recebeu a autorização para atravessar um cruzamento;
- Cor Vermelha: indica que um veículo, que está se aproximando a um cruzamento, tomou a decisão ou recebeu a ordem de não atravessar um cruzamento ainda.

Vale ressaltar que essa coloração dos veículos foi criada apenas a nível de simulação para facilitar a visualização do que acontece e que, para aplicação futura do Modo Colônia em situações reais, será ainda necessário realizar diversos estudos para definir formas de sinalização visual e/ou em rádio-frequência entre os veículos para evitar acidentes e garantir a segurança, de forma redundante ao SCT.

4.5 Sistema de Controle de Tráfego

Para o funcionamento correto do Modo Colônia, é necessário que seja feita a construção de um SCT eficiente e bem distribuído, com aplicações executando nas camadas de *Fog Computing* e *Cloud Computing*. Por se tratar de um sistema muito crítico, é preciso prover o máximo de redundância e segurança possível. Existem diversos problemas a serem resolvidos para a implementação desse sistema, desde questões de segurança da informação, segurança dos veículos e de seus passageiros até questões legais e burocráticas. Entretanto, o escopo desse trabalho se restringe a uma prova de conceito, portanto, foi implementado uma versão inicial e simplificada de um SCT, contendo apenas o módulo de controle de cruzamento. Fica como ideia para projetos futuros expandir a arquitetura aqui idealizada por meio da aplicação do Modo Colônia em outras situações comuns no trânsito de veículos.

A implementação do SCT deste projeto foi feita utilizando a biblioteca Node.js, que já prove diversas funcionalidades importantes para se construir uma aplicação do tipo servidor em *JavaScript*. A aplicação desenvolvida utiliza *WebSockets* para realizar a conexão entre o Ambiente de Simulação e o SCT, para gerenciar essa conexão, foi desenvolvido uma interface e um protocolo de troca de mensagens, implementado tanto no módulo

de simulação quanto no módulo de controle de tráfego. O protocolo de troca de mensagens desenvolvido se restringe ao funcionamento da simulação para prova de conceito. Portanto, existem algumas simplificações e características que, provavelmente, não serão herdadas em uma implementação de sistema real.

O objetivo principal do SCT é tomar decisões e controlar o tráfego de veículos. Para isso, é necessário que os veículos transmitam informações para a central e recebam comandos de controle. Isso pode ser feito por meio da definição de um protocolo de troca de mensagens. Para simplificar a integração da conexão das entidades carro e semáforo com esse protocolo de troca de mensagens, foi desenvolvido um módulo de conexão ao servidor (`sim/net/server_connection.js`). Esse módulo gerencia a parte de conexão de rede e oferece suporte à simulação do Modo Colônia.

Para o funcionamento do sistema de controle, todos os veículos autônomos que se conectarem devem continuamente transmitir informações a respeito da sua posição atual, velocidade e possivelmente informações sobre rota desejada. Assim, com base nos dados recebidos dos veículos e de demais sensores presentes nas rodovias, pode ser feita a tomada de decisão por parte do SCT. Além disso, é necessário também que sejam feitas verificações do estado da conexão e latência continuamente, pois, caso esses parâmetros estejam fora dos limites desejados, é necessário interromper o funcionamento do Modo Colônia. Portanto, é importante implementar diversas instâncias do SCT, de preferência em infraestruturas *Fog* localizadas a distâncias da ordem de dezenas de quilômetros.

4.5.1 Autorização de Travessia e Segurança do Cruzamento

Após todas as conexões bem estabelecidas, incluindo do próprio semáforo, o SCT autoriza a operação em Modo Colônia e passa a realizar o controle do tráfego, enviando constantemente mensagens para todos os veículos e para o semáforo. Caso o semáforo detecte qualquer instabilidade na conexão, ele irá imediatamente voltar para o funcionamento em Modo Legado, começando por luzes vermelhas, ou seja, interrompendo o fluxo de veículos.

Nesse contexto, surge a problemática de garantir que não irão ocorrer acidentes caso ocorra uma falha na operação ou na conectividade de rede com o SCT. Para isso, o sistema de controle de cruzamento trabalha com autorizações para travessia de cruzamento. Ao receber uma autorização para travessia de cruzamento, o veículo autônomo deve acelerar até a velocidade da via e seguir com a travessia, não podendo voltar atrás na decisão, mesmo que a conexão seja interrompida. Uma autorização de travessia somente será emitida pelo SCT quando todas as variáveis garantirem a segurança para o cruzamento deste veículo a tempo.

Os veículos autônomos devem seguir na velocidade informada pela central até atingir a zona de limite de frenagem. A zona limite de frenagem representa a distância limite,

com base na velocidade atual, para que o veículo possa realizar uma parada completa em segurança antes de cruzar o semáforo. Caso uma autorização de travessia não seja recebida, o veículo irá automaticamente reduzir sua velocidade dentro da margem de segurança.

Outra condição importante para garantir a segurança da operação é trabalhar com velocidades de tráfego decrescentes proporcionalmente à posição do veículo na fila de passagem, isto é, o veículo que estiver na primeira posição da fila de cruzamento terá sua velocidade definida para que possa atravessar o cruzamento no momento mais eficiente e com segurança, em seguida, todos os veículos das posições seguintes terão sua velocidade calculada com base em previsões de cruzamentos futuros, porém, limitada à velocidade do veículo a sua frente.

Finalmente, para garantir a segurança da operação, é importante que sejam definidos limites de operação bem estritos, onde perda de pacotes ou aumento de latência devem levar o sistema a reduzir a velocidade máxima de tráfego ou, nos piores casos, desligar o Modo Colônia por completo naquele cruzamento. Esses limites de operação serão testados de forma constante dentro do protocolo de troca de mensagens, incluindo medição de latência e qualidade de conexão entre cada veículo e a central de controle.

Respeitando todas essas condições, caso ocorra um desligamento repentino do Modo Colônia, por qualquer motivo que seja, todos os veículos já terão informações suficientes para continuar trafegando em segurança. Os veículos que já tinham sido autorizados irão atravessar o cruzamento. Os veículos que não foram autorizados, como respeitam a margem de segurança da zona limite de frenagem, terão espaço suficiente para frenagem, sem que ocorram acidentes.

4.5.2 Algoritmos para Controle de Velocidade e Prioridade de Travessia

Além de garantir que não irão ocorrer colisões entre veículos, a outra tarefa principal do SCT é otimizar o fluxo e maximizar a eficiência do processo e, para tal, deve trabalhar constantemente na solução de duas problemáticas: definir a ordem de cruzamento de veículos; definir a velocidade desejada para cada veículo trafegar. Ou seja, se tratam de problemas de priorização, ordenação e otimização.

O Algoritmo utilizado para fazer o controle do cruzamento pode ser construído utilizando abordagens distintas. Como prova de conceito, a implementação deste projeto utiliza um algoritmo que prioriza a passagem de veículos de acordo com o tempo de espera. O tempo de espera é definido como o tempo decorrido desde que o veículo entra na faixa de rodagem até que atravesse o cruzamento. Além disso, cada uma das faixas de

rodagem orientadas no sentido do cruzamento pode ser vista como uma fila de veículos. Assim, o algoritmo prioriza a passagem da fila de veículos cujo primeiro colocado está com o maior tempo de espera. Dessa forma, é possível obter um equilíbrio na passagem de veículos, tanto na pista leste/oeste quanto na pista norte/sul.

Ainda na parte de priorização da passagem de veículos, é necessário levar em conta que um cruzamento de quatro sentidos de tráfego pode permitir tráfego simultâneo de veículos com sentidos opostos, pois não há cruzamento entre as pistas de mesma direção e sentidos opostos. Assim, ao decidir quanto à autorização da passagem de veículos, após decidir qual fila priorizar, o algoritmo desenvolvido também calcula quanto tempo levaria para o veículo da fila de sentido oposto realizar a passagem. Caso o tempo previsto para essa passagem seja suficientemente pequeno, o algoritmo prioriza a passagem do veículo de sentido oposto.

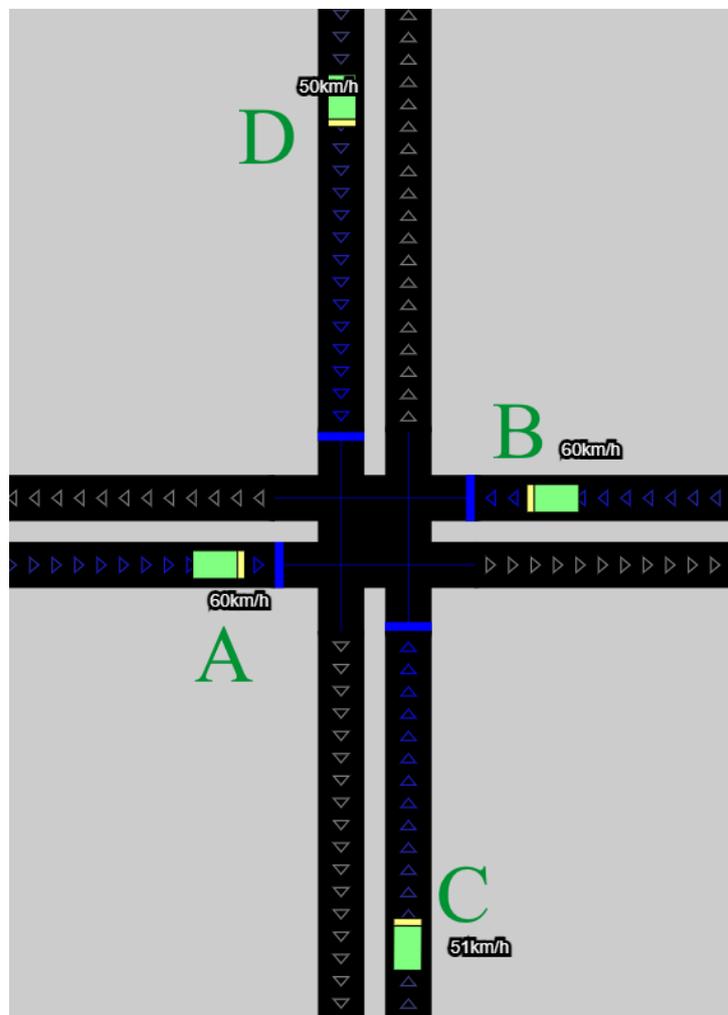


Figura 4.6: Momento de simulação onde foi priorizada a passagem de veículos em sentidos opostos durante operação em Modo Colônia.

A Figura 4.6 demonstra a priorização de passagem para veículos em sentidos opostos. No momento capturado, o carro que está na primeira posição da fila é o destacado com a letra A. Lembre-se que a coloração verde do veículo quer dizer que ele já recebeu autorização para cruzar a via. Portanto, o veículo destacado com a letra B também já recebeu autorização de travessia. Dessa forma, os veículos A e B irão passar no cruzamento de forma praticamente simultânea. Os efeitos de permitir ou não essa passagem em sentidos opostos serão analisados no Capítulo 5.

Ainda observado a Figura 4.6, nota-se que os veículos C e D também já receberam autorização para passar no cruzamento. Isso aconteceu por conta de uma segunda otimização implementada no algoritmo. Pode-se chamar essa otimização de adiantamento de autorização. Esse adiantamento pode ocorrer para um veículo desde que o sistema garanta que todos os veículos na frente da fila já terão finalizado a travessia no momento em que o mesmo iniciará a sua travessia. Ainda no exemplo da Figura 4.6, é possível perceber que os veículos A e B estão trafegando na velocidade máxima definida na configuração da simulação, 60km/h , e por meio das equações de movimento, é possível garantir que, caso a velocidade seja mantida, ambos veículos A e B terão finalizado a travessia antes que os veículos C e D atinjam a zona limite de frenagem.

Em suma, este capítulo trouxe um apanhado das principais características da proposta do Modo Colônia e das implementações e simulações concluídas até então. Em seguida, no próximo capítulo, serão apresentadas análises de dados coletados dentro das simulações, a fim de levantar um paralelo comparativo entre o Modo Colônia e o funcionamento padrão dos cruzamentos. Dessa forma, será possível entender quais otimizações e benefícios de fato podem ser atingidos implementando as ideias aqui apresentadas.

Capítulo 5

Análise e Discussão dos Resultados

Este capítulo apresentará a análise a respeito do SCT implementado, incluindo comparações do funcionamento da proposta do Modo Colônia em relação ao Modo Legado. Nesse contexto, considera-se que o Modo Legado seja o comportamento padrão de um cruzamento comum da atualidade, sem a implantação da tecnologia do SCT para organização do fluxo de veículos autônomos. Assim, será feita uma análise das otimizações adquiridas por meio da implantação do SCT operante em Modo Colônia, principalmente no quesito velocidade de tráfego, para um cruzamento de duas avenidas perpendiculares. Finalmente, serão apresentados uma breve análise sobre a largura de banda consumida pela atual implementação e sugestões para trabalhos futuros baseados nessa proposta do Modo Colônia.

Para coleta e análise de dados do SCT implementado e posterior comparação do Modo Colônia com o Modo Legado, foi iniciada uma simulação contendo um cruzamento central, oito faixas de rodagem adjacentes, formando dois ciclos, um no sentido Leste/Oeste e outro no sentido Norte/Sul, conforme a Figura 5.1. O eixo vertical pode ser considerado o eixo Norte/Sul, sendo Norte a direção para cima. Conseqüentemente, o eixo horizontal é considerado o eixo Leste/Oeste. Além disso, foram inseridos quatro carros autônomos para trafegar em cada ciclo, totalizando oito carros na simulação. Todas as dimensões e configurações utilizadas para as entidades da simulação estão no padrão definido no capítulo anterior.

5.1 Análise da Velocidade de Tráfego

Em primeiro lugar, a variável mais importante e intuitiva que diz a respeito da otimização do tráfego de veículos em uma determinada região é a velocidade com que estes veículos trafegam na via. Nesse sentido, pode-se considerar que quanto mais alta for a média das velocidades dos veículos na região, mais otimizado está o tráfego. Portanto, será feita uma

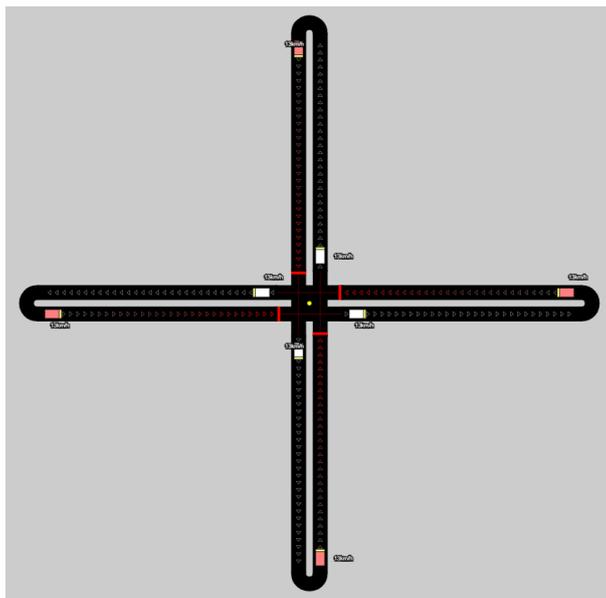


Figura 5.1: Configuração do ambiente de testes e coleta de dados.

análise comparativa de dados coletados referente às velocidades de veículos trafegando na simulação operante em Modo Colônia versus na simulação operante em Modo Legado.

Para esta análise será considerado que todos os veículos respeitam os limites de velocidade da via, uma vez que todos os carros são autônomos e: ou estão operando no Modo Legado, ou estão operando no Modo Colônia, mas nunca estão operando sob os comandos de um motorista. Dessa forma, nesse ambiente controlado, é possível isolar apenas as diferenças e otimizações advindas da implantação da proposta apresentada.

Para realizar análise da velocidade dos veículos na simulação, foram coletados dados a cada um segundo, incluindo a média da velocidade de todos os veículos e a velocidade de um veículo individualmente, além do odômetro de cada veículo a cada instante de tempo. A média das velocidades em um determinado instante consiste em somar todas as velocidades dos veículos nesse instante e dividir pela quantidade de veículos presentes. A coleta de dados foi feita durante o intervalo de 120 segundos, uma vez para o Modo Legado e uma vez para o Modo Colônia.

Os dados de velocidade coletados para cada um dos testes foram processados e formatados em gráficos, para melhor visualização. Os dados referentes ao Modo Legado estão dispostos nos gráficos da Figura 5.2, já os referentes ao Modo Colônia foram apresentados nos gráficos da Figura 5.3. Em ambas as figuras, por um lado, o gráfico da parte superior representa a média das velocidades dos veículos para cada instante de tempo; por outro lado, o gráfico da parte inferior representa a velocidade instantânea de um dos veículos para cada instante de tempo.

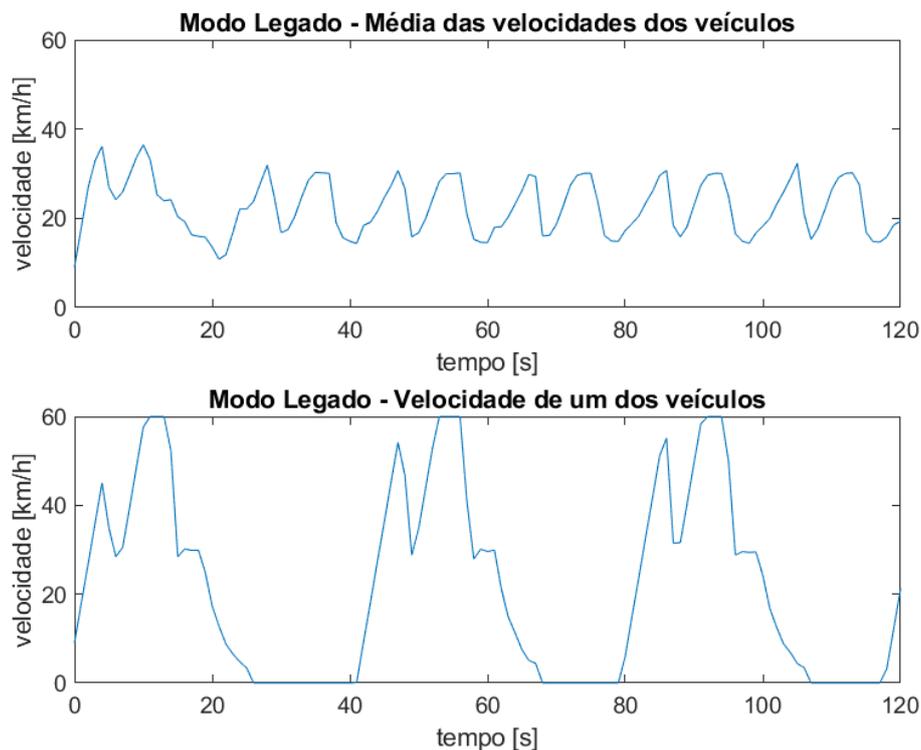


Figura 5.2: Dados de velocidade para Modo Legado.

5.1.1 Análise da Velocidade Individual

Primeiramente, analisando o segundo gráfico da Figura 5.2, referente à velocidade de um dos veículos operando em Modo Legado, é possível perceber um padrão de movimento cíclico com período de cerca de quarenta segundos. A cada ciclo, percebe-se que o veículo inicia o movimento - quando o semáforo fica verde - e segue acelerando até quase atingir a velocidade máxima, então, reduz a velocidade para cerca de 30 km/h para fazer uma curva. Em seguida, o veículo segue no sentido oposto da via e consegue passar em velocidade máxima (60 km/h) pelo cruzamento, pois o semáforo ainda estava verde naquela direção. Finalmente, o veículo faz a outra curva, reduzindo sua velocidade para 30 km/h, mas não acelera novamente, pois o sinal à sua frente já está vermelho novamente.

Na sequência, analisando o segundo gráfico da Figura 5.3, referente à velocidade de um dos veículos operando em Modo Colônia, é possível perceber um padrão de movimento diferenciado. Existe uma etapa inicial onde a velocidade do veículo permanece em torno de 20 km/h - essa é a fase onde o SCT ainda está inicializando e está adaptando as velocidades dos veículos para otimizar a performance do trânsito. Em seguida, cerca de 15 segundos após a inicialização, percebe-se que o veículo recebe autorização para acelerar até a velocidade máxima da via e, depois, reduz um pouco a velocidade para fazer a primeira curva a 30 km/h. Novamente, o veículo se aproxima do cruzamento, mas em

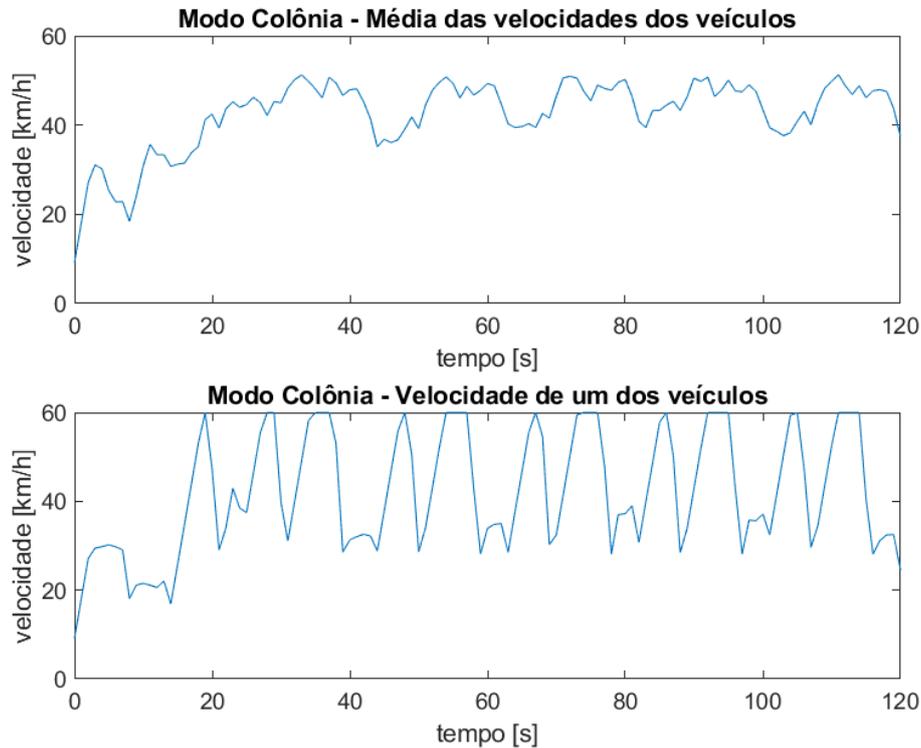


Figura 5.3: Dados de velocidade para Modo Colônia.

sentido oposto. Entretanto, em nenhuma das aproximações o veículo teve sua velocidade reduzida a 0 km/h, ou seja, não foi necessária uma parada completa em nenhum momento. Finalmente, é possível observar que o veículo permanece o resto da simulação em um ciclo com velocidades variando entre e 30 km/h e 60 km/h.

Nesse contexto, comparando o comportamento da velocidade individual do Modo Legado com o Modo Colônia, é possível perceber uma otimização extremamente importante: não existem paradas no Modo Colônia. O SCT conseguiu exercer seu papel de otimização e controlou a passagem de cada um dos oito veículos no cruzamento sem que fosse necessário que nenhum deles chegasse a uma parada completa. Dessa forma, os passageiros dos veículos provavelmente tiveram uma experiência de viagem fluida e consistente. Em contrapartida, os passageiros do veículo que trafegava no cruzamento operando em Modo Legado tiveram que parar diversas vezes para aguardar o semáforo vermelho. A consequência dessa otimização pode refletir em diversos pontos, incluindo redução no desperdício de energia e gasto de combustível enquanto se aguarda no semáforo vermelho, redução do tempo médio das viagens urbanas e aumento da fluidez do movimento, eliminando paradas e arranques por completo.

5.1.2 Análise das Médias das Velocidades

Outrossim, ao se avaliar a performance do Modo Colônia é importante observar, além do comportamento da velocidade individual de cada veículo, o comportamento geral dos veículos, por meio das médias das velocidades. Com base nos primeiros gráficos das Figuras 5.2 e 5.3, observa-se que ambos possuem duas etapas marcantes. A primeira acontece na etapa de início da simulação, que pode ser chamada de etapa de convergência, ou estado transiente, na qual as velocidades médias partem do repouso e são reguladas pelo meio externo, até que cheguem à segunda etapa marcante, um estado de equilíbrio ou estado estacionário. No caso do Modo Legado, é possível perceber que se atingiu um equilíbrio por volta dos 25 segundos de simulação. No caso do Modo Colônia, pode-se dizer que esse equilíbrio começa a acontecer por volta dos 30 segundos de simulação, pois, depois disso, observa-se uma oscilação estacionária.

Nesse contexto, é possível observar que: Por um lado para o Modo Legado, a média das velocidades dos veículos, para todos os instantes do estado de equilíbrio, oscilou entre cerca de 15 km/h até cerca de 30 km/h, o que representa cerca de 25% e 50% da velocidade máxima permitida na via; Por outro lado, para o Modo Colônia, a média das velocidades oscilou dentro de um patamar de cerca de 35 km/h até cerca de 50 km/h, o que representa, respectivamente, 58% e 83% da velocidade máxima da via. Portanto, comparando os estados estacionários, os menores valores das médias das velocidades do Modo Colônia esteve 5 km/h acima dos maiores valores para o Modo Legado, ou seja, um aumento consistente e considerável na performance do sistema, mesmo quando comparando o pior momento do Modo Colônia com o melhor momento do Modo Legado.

5.1.3 Análise dos Deslocamentos Individuais

Além da coleta de velocidades, foi coletado também o odômetro total de todos os veículos nas simulações, ou seja, dados referentes ao deslocamento total de cada um dos veículos dentro do intervalo de tempo observado, de 120 segundos. As distâncias percorridas por cada um dos veículos durante os 60 últimos segundos da simulação foram inseridas na tabela 5.1. A coluna *id* representa a identificação única de cada veículo, ΔS_L representa o deslocamento total no Modo Legado e ΔS_C representa o deslocamento total no Modo Colônia. Com base nesses dados, foram extrapoladas as colunas V_L , que representa a velocidade média do veículo no Modo Legado e, finalmente, V_C , velocidade média no Modo Colônia.

Observando a tabela 5.1 é possível perceber que, para o Modo Legado, a velocidade média de tráfego dos veículos foi de 22 km/h apenas, com valores variando desde 17 km/h a 28 km/h, isto é, uma variação de 5 km/h da média para mais ou para menos.

id	ΔS_L	V_L	ΔS_C	V_C
17	277 m	17 km/h	746 m	45 km/h
18	287 m	17 km/h	758 m	46 km/h
19	457 m	27 km/h	746 m	45 km/h
20	467 m	28 km/h	775 m	46 km/h
21	277 m	17 km/h	750 m	45 km/h
22	283 m	17 km/h	755 m	45 km/h
23	458 m	27 km/h	762 m	46 km/h
24	467 m	28 km/h	768 m	46 km/h
Média	372 m	22 km/h	757 m	45 km/h

Tabela 5.1: Dados comparativos entre deslocamentos do Modo Legado versus Modo Colônia.

Isso mostra que os veículos trafegando no Modo Legado estão sujeitos a desigualdades na priorização de passagem no cruzamento, dependendo de fatores que estão fora do controle dos motoristas e dos carros autônomos. Por outro lado, é possível notar que todos os carros, quando trafegam em Modo Colônia, possuem uma velocidade média estável e balanceada, chegando a 45 km/h, que representa 75% da velocidade máxima permitida nas fixas de rodagem de linha reta. Vale ressaltar que o valor poderia se aproximar mais ainda de 60 km/h caso não houvessem curvas nas extremidades das pistas. Finalmente, comparando as velocidades médias finais da tabela 5.1, obteve-se, no Modo Colônia, praticamente o dobro da velocidade média do Modo Legado.

5.2 Análise de Utilização de Recursos

Uma outra informação importante para implementação de sistemas computacionais é observar qual a quantidade de recursos utilizados para que uma aplicação execute. Nesse sentido, foi feita a coleta de informações referentes ao total de quantidade de mensagens enviadas por segundo, bem como a largura de banda média utilizada. A análise foi feita apenas para os dados que são enviados dos dispositivos IoT para o SCT, o caminho inverso é bem similar, praticamente simétrico. Não foram feitas análises referentes a utilização de recursos como memória e processamento, uma vez que o tamanho da simulação feita foi pequeno. Para melhor entender como o sistema se comportaria de forma real, seria necessário implementar módulos adicionais para organização do sistema em nível global ou nacional, com várias regiões de disponibilidade e algoritmos para determinação de quais *datacenters* utilizar para processamento das informações. Dadas essas ressalvas, serão observadas apenas informações básicas referentes à troca de mensagens entre os dispositivos e a central.

N_{cars}	T_{update}	b_{rate}	m_{rate}
8	25ms	360 Kbps	340/s
100	25ms	4.5 Mbps	4.3K/s
1000	25ms	45 Mbps	43K/s
8	5ms	1.8 Mbps	1.7K/s
100	5ms	22.5 Mbps	21.3K/s
1000	5ms	225 Mbps	213K/s

Tabela 5.2: Estimativa de utilização de recursos de rede para funcionamento do Modo Colônia.

Utilizando as mesmas configurações da simulação feita para coleta de dados de velocidade, isto é, as mesmas configurações de pista e mesma quantidade de veículos, sendo oito no total, obteve-se o seguinte resultado: largura de banda utilizada de trezentos e sessenta Kilobits por segundo (360 Kbps) para 340 mensagens por segundo. Nesse caso, haviam nove dispositivos conectados no SCT, sendo um semáforo e oito carros. A taxa de atualização de posição dos veículos e do estado do semáforo é feita a cada 25 milissegundos, conforme configurado na simulação, o que significa 40 mensagens por segundo, por dispositivo. Nesse caso, houve uma taxa média de cerca de 1000 bits (ou 1 kilobit) por mensagem trocada. Com base nessas informações, é possível realizar uma extrapolação para estimar a utilização de banda conforme a quantidade de veículos aumenta, ou ainda, variando a frequência da taxa de atualização. Essa extrapolação de dados está disposta na tabela 5.2. Vale ressaltar que esses dados podem conter imprecisões por conta da simulação, portanto, servem apenas como uma aproximação e, para aplicação real, seria necessário um teste mais preciso.

A tabela 5.2 apresenta em suas colunas, respectivamente:

- N_{cars} : número de veículos autônomos conectados no sistema. Quanto maior a quantidade de veículos, maior a quantidade de recursos utilizados, maior largura de banda e maior quantidade de pacotes por segundo. Portanto, a utilização de recursos é diretamente proporcional à quantidade de veículos conectados;
- T_{update} : intervalo de tempo entre as atualizações de posicionamento dos veículos. Quanto menor o intervalo utilizado, maior a quantidade de recursos utilizados, porém, melhor performance, como será discutido a seguir;
- b_{rate} : largura de banda, *Bitrate*, quantidade de bits trafegados por segundo. É uma das principais métricas para se avaliar a carga da rede;
- m_{rate} : quantidade de mensagens trocadas por segundo. Representa quantas mensagens foram recebidas pelos dispositivos IoT por segundo durante a simulação.

É importante ressaltar que a variação da frequência da taxa de atualização afeta diretamente a performance e a segurança do funcionamento do sistema de controle de tráfego, pois afeta diretamente a precisão das informações utilizadas para tomada de decisão por parte do SCT. Nesse contexto, quanto maior a frequência de atualização de posição utilizada, melhor o comportamento do SCT, desde que se mantenha uma taxa suportada pela rede e pelos recursos computacionais utilizados no processamento desses dados, pois, um aumento exagerado da quantidade de tráfego e dados pode causar falhas no sistema. Nesse sentido, quanto mais precisas e quanto menor o atraso das informações coletadas pelo SCT, maior as chances de se permitir uma velocidade de tráfego elevada, mantendo elevados padrões de segurança. Não foi o escopo deste trabalho, mas é preciso que o SCT atue reduzindo a velocidade dos veículos caso as condições de rede fiquem afetadas, por qualquer motivo que seja. Ainda, quanto maior a latência da conexão, maior deve ser também o intervalo de atualização de posição, uma vez que não adianta aumentar drasticamente a taxa de atualização em conexões de alta latência.

Antes de realizar a coleta dos dados de largura de banda, o autor imaginou que a largura de banda utilizada pela implementação feita seria alta, uma vez que a taxa de atualização de posição é relativamente alta e se trata de um sistema de controle muito crítico. Entretanto, a taxa de bits observada, na realidade, fica dentro de padrões bem razoáveis para as conexões de rede existentes na atualidade, onde trafegar vários Megabits por segundo já é trivial para conexões moveis, por exemplo, para transmissão de vídeo em tempo real. Esses resultados trouxeram uma perspectiva positiva para viabilidade do sistema proposto, uma vez que, com base na taxa de bits observada, é possível concluir que já seria viável para implantação do sistema proposto utilizando tecnologias de conectividade da atualidade, tais como 4G/LTE (*Long Term Evolution*) ou 5G, já presentes de forma vasta nas principais capitais do país e em diversos outros centros urbanos. Vale a pena ressaltar que essa largura de banda mantém-se constante para os veículos de forma individual, sendo por volta de 45 Kbps para uma taxa de 40 atualizações por segundo, ou seja, os valores nas casas de megabit referenciam a largura de banda necessária para funcionamento do servidor de processamento e não dos dispositivos IoT em si.

5.3 Considerações Finais das Análises

Em síntese, com base nas análises realizadas neste capítulo, obteve-se, ao implantar o Modo Colônia:

- Eliminação de praticamente 100% das paradas e arranques: no Modo Colônia, como todos os veículos trabalham em harmonia e, por meio da troca de mensagens com a central, é possível balancear o tráfego, fazendo com que alguns veículos reduzam um

pouco a velocidade, dando passagem para outros veículos, reduzindo a necessidade de parada completa para valores muito próximos de zero; Isso implica em uma redução de desperdício energético considerável, uma vez que o movimento veicular tem perdas energéticas maiores nas seções de aceleração e desaceleração;

- Otimização na velocidade média de tráfego: ao se aplicar o Modo Colônia, observou-se que a velocidade média do tráfego é praticamente o dobro, para o contexto em que foi aplicado. Isso acontece pois os algoritmos de controle de tráfego visam utilizar da forma mais eficiente possível os recursos de rodagem disponíveis, eliminando por completo a competição entre veículos por espaço nas pistas;
- Tráfego com velocidade média de 75% da velocidade máxima permitida: ao se aplicar o Modo Colônia, obteve-se, no estado estacionário da simulação, uma velocidade média de mais de cerca de 45 km/h, o que representa 75% da velocidade máxima permitida de 60 km/h. Esse é apenas um começo, utilizando um algoritmo simples, pode ser melhorado ainda mais e, talvez, ainda aumentar a velocidade máxima permitida nas vias, uma vez que a segurança viária seria consistentemente melhorada devido ao controle baseado em sistemas;
- Redução significativa, praticamente total, da variância da velocidade média de tráfego entre os diferentes veículos: além do aumento da velocidade de tráfego, houve também uma redução das divergências entre velocidades dos diferentes veículos que trafegam nas vias. Isso significa um cenário mais justo e equilibrado, o que traz ganhos para todos;
- Possibilidade de implantação utilizando tecnologias de rede da atualidade: conforme foi observado, levando como base o quesito largura de banda, já seria possível implantar o sistema proposto utilizando as tecnologias de rede da atualidade. Entretanto, é necessário ainda realizar estudos mais aprofundados sobre latência e trabalhar em cima do problema das contradições de latência existentes nas rotas da Internet apresentado no capítulo de desafios.

Para encerrar, por hora, a discussão do tema e da aplicação apresentada, no próximo capítulo serão feitas as considerações finais deste projeto, incluindo perspectivas futuras de pesquisas na área, uma vez que este projeto se limitou apenas a uma prova do conceito, visando provar a viabilidade do sistema proposto.

Capítulo 6

Considerações Finais

Este capítulo apresenta as considerações finais deste projeto de conclusão de curso. Por meio da realização deste trabalho foi possível fazer um levantamento teórico a respeito de diversas tecnologias vigentes na atualidade e que certamente serão de extrema importância para viabilização de diversos projetos e soluções desta década e das décadas seguintes. Nesse sentido, mostrou-se que *Fog Computing* possui inúmeras aplicações práticas, principalmente quando está inserido no contexto da Internet das Coisas, trazendo baixa latência e otimizações baseadas na localização geográfica. Em seguida, foi também discutido que ainda existem diversos desafios a serem vencidos para difusão de serviços de *Fog Computing* a redor do globo, incluindo o desafio de se atualizar as políticas de roteamento na Internet a fim de priorizar baixa latência para conexões regionais entre diferentes provedores de serviço. Finalmente, foi trago uma proposta de aplicação prática de *Fog Computing* no contexto de IoT, voltada para otimização do trânsito em Cidades Inteligentes, seguido de detalhes sobre a aplicação desenvolvida e resultados obtidos por meio de uma simulação para prova de conceito do Modo Colônia. Portanto, vale agora destacar os principais resultados obtidos e comentar a respeito de futuros trabalhos que podem ser desenvolvidos em cima das ideias apresentadas neste projeto.

6.1 Síntese dos Resultados Obtidos

A realização deste trabalho trouxe diversas informações interessantes a respeito do mundo de *Cloud Computing*, *Fog Computing* e Internet das Coisas, além das discussões e resultados obtidos por meio da proposta do Modo Colônia. Nesse contexto, vale ressaltar a relação entre *Fog* e *Cloud*, sendo que *Fog* não veio para substituir a nuvem, mas, para agregar e trabalhar em cima das limitações de latência e distância geográfica das nuvens. Ademais, notou-se uma afinidade imensa entre as tecnologias de IoT e a plataforma de *Fog*, onde diversos casos de uso muito interessante estão disponíveis, havendo espaço

para investimentos, pesquisas e aplicações práticas, seja na indústria ou no uso cotidiano das pessoas. Nesse sentido, o SCT apresentado traz espaço para integração das diversas tecnologias citadas em uma aplicação prática nas Cidades Inteligentes.

A implantação teórica do Modo Colônia, dentro das condições simuladas, trouxe resultados e otimizações bastante interessantes no contexto da inserção de tecnologias de automação voltadas para o controle de trânsito, principalmente: A eliminação de praticamente 100% das paradas e arranques, sem necessidade de se parar em sinais vermelhos; Otimização da velocidade de tráfego média, com velocidades médias atingindo 75% da velocidade máxima permitida; Redução significativa, praticamente total, da variância da velocidade média de tráfego entre os diferentes veículos, provendo balanceamento e igualdade na priorização de tráfego, o que traz maior fluidez e segurança para o trânsito; Por fim, mostrou-se que, com base nas estimativas de utilização de recursos, haveria a possibilidade de implantação utilizando tecnologias de rede da atualidade.

Outrossim, a implementação prática deste trabalho, por meio do Ambiente de Simulação desenvolvido, trouxe a possibilidade de se realizar diversos trabalhos futuros em cima dos temas de *Fog Computing*, Internet das Coisas, Simulações de Trânsito, etc. Entretanto, ainda existem diversas limitações presentes no estado atual deste projeto. Portanto, essas duas questões serão discutidas na seção seguinte.

6.2 Limitações e Trabalhos Futuros

A implementação de um trabalho, projeto ou ainda a tentativa de transformar ideias em realidade consiste em uma constante batalha entre o mundo ideal e o mundo real, isto é, para que se chegue em um resultado final é necessário que sejam feitas simplificações de ideias e limitações de escopo de forma constante, caso contrário, não seria possível chegar a resultado algum, pois sempre existe algo a acrescentar ou aprimorar. Não diferente, durante várias vezes foi necessário limitar o escopo deste projeto a fim de tê-lo finalizado em tempo hábil. Dessa forma, nessa seção serão apresentadas algumas limitações das ideias aqui discutidas e, em seguida, ideias para trabalhos futuros na área.

Primeiramente, a aplicação de tecnologias em setores críticos, tais como segurança, transporte, saúde, energia, etc, deve levar em conta diversos fatores que não estiveram dentro do escopo deste projeto. Nesse sentido, para se implantar um SCT real, é necessário ainda que sejam feitos estudos voltados, por exemplo, para segurança da aplicação, tolerância a falhas, segurança de trânsito, adequação à políticas e legislação, viabilidade técnica, viabilidade econômica e muitos outros. Além disso, o ambiente simulado foi apenas uma simplificação de um cruzamento, contendo apenas uma faixa de rodagem por sentido de tráfego, não havendo, ainda, a possibilidade que os veículos façam conversões

à direita ou esquerda. Por fim, as ideias apresentadas neste trabalho abrem espaço para, por exemplo:

- Aplicação do Modo Colônia para otimização de serviços de urgência, por meio de desenvolvimento de algoritmos de priorização de tráfego para serviços de emergência e segurança, tais como ambulâncias e viaturas de polícia;
- Aplicação do conceito do Modo Colônia em outras estruturas de trânsito, tais como retornos, rotatórias, vias e cruzamentos com múltiplas faixas de rodagem;
- Inclusão do processamento de rotas dos veículos por meio do SCT, isto é, os veículos autônomos podem enviar, além do seu posicionamento atual, qual rota desejam trafegar e qual o destino que desejam chegar. Dessa forma, seria possível ainda desenvolver algoritmos de roteamento de veículos que levassem em conta toda a infraestrutura e informações contidas no SCT, incluindo detalhes precisos à respeito da situação de trânsito em todas as regiões de uma Cidade Inteligente;
- Desenvolvimento de um protocolo de troca de mensagens robusto e eficiente. O canal de comunicação utilizado na implementação deste trabalho foi baseado em *WebSockets* em cima de conexões TCP, por conta da compatibilidade e facilidade de integração com o ambiente de simulação em *JavaScript*. Entretanto, para um sistema real, seria necessário utilizar protocolos com menos *overhead*, tal como o protocolo UDP, que pode permitir a troca de mensagens com menor latência que o TCP, mas necessita de mais cuidado, pois não garante a consistência da conexão e do canal;
- Estudos e desenvolvimento de algoritmos para tratamento de panes e demais situações que existem no mundo real, tais como: animais na pista, passagem de pedestres, problemas no asfalto, buracos nas pistas, acidentes, etc. Nesse sentido, seria possível ainda que os veículos enviassem metadados adicionais, em mensagens de menor prioridade, contendo detalhes a respeito da situação da via. Além disso, seria possível ainda que os veículos enviassem mensagens de emergência ao SCT para que se evite acidentes ou colisões, o que seria bem efetivo, uma vez que a latência provida pelas infraestruturas de *Fog Computing* permite tempos de resposta da ordem de poucas dezenas de milissegundos, o que é menor que o tempo típico de resposta motora de um ser humano, que é da ordem de centenas de milissegundos [15].

Em suma, é possível concluir que os resultados esperados para realização do trabalho foram atingidos, por meio da prova de conceito da ideia do Modo Colônia implementada em forma de simulação. Além disso, os resultados obtidos foram positivos e trazem boas expectativas para inclusão de tecnologias de controle de tráfego nas Cidades Inteligentes.

Finalmente, este trabalho trouxe uma base para realizações de mais projetos práticos na área, o que é de extrema importância para consolidação e validação das ideias e pesquisas do mundo acadêmico.

Referências

- [1] Iorga, Michaela, Larry Feldman, Robert Barton, Michael J Martin, Nedim S Goren e Charif Mahmoudi: *Fog computing conceptual model*. 2018. ix, 1, 6, 7, 8
- [2] Bachiega Jr., J., Costa B. Carvalho L. Oliveira V. Santos W. S. de Castro M. e A Araujo: *From the sky to the ground: Comparing fog computing with related distributed paradigms*. Proceedings of the 12th International Conference on Cloud Computing and Services Science (CLOSER 2022), pages 158-169, 2022. ix, 1, 4, 6, 9
- [3] Chiang, Mung e Tao Zhang: *Fog and iot: An overview of research opportunities*. IEEE Internet of things journal, 3(6):854–864, 2016. ix, 1, 2, 10, 11
- [4] Mell, Peter, Tim Grance *et al.*: *The nist definition of cloud computing*. 2011. 1, 4, 5, 6
- [5] Haouari, Fatima, Ranim Faraj e Jihad M AlJa’am: *Fog computing potentials, applications, and challenges*. Em *2018 International Conference on Computer and Applications (ICCA)*, páginas 399–406. IEEE, 2018. 1, 14, 15, 16, 17
- [6] Gill, Sukhpal Singh, Shreshth Tuli, Minxian Xu, Inderpreet Singh, Karan Vijay Singh, Dominic Lindsay, Shikhar Tuli, Daria Smirnova, Manmeet Singh, Udit Jain *et al.*: *Transformative effects of iot, blockchain and artificial intelligence on cloud computing: Evolution, vision, trends and open challenges*. Internet of Things, 8:100118, 2019. 1, 6, 9
- [7] Madakam, Somayya, Vihar Lake, Vihar Lake, Vihar Lake *et al.*: *Internet of things (iot): A literature review*. Journal of Computer and Communications, 3(05):164, 2015. 2, 10, 11, 12
- [8] Feroz, Bushra, Amjad Mehmood, Hafsa Maryam, Sherali Zeadally, Carsten Maple e Munam Ali Shah: *Vehicle-life interaction in fog-enabled smart connected and autonomous vehicles*. IEEE Access, 9:7402–7420, 2021. 10
- [9] Heydon, Robin e Nick Hunn: *Bluetooth low energy*. CSR Presentation, Bluetooth SIG <https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx>, 2012. 12
- [10] Mazzola, Fabricio, Pedro Marcos, Ignacio Castro, Matthew Luckie e Marinho Barcellos: *On the latency impact of remote peering*. Em *International Conference on Passive and Active Network Measurement*, páginas 367–392. Springer, 2022. 12, 21

- [11] Jalali, Fatemeh, Kerry Hinton, Robert Ayre, Tansu Alpcan e Rodney S Tucker: *Fog computing may help to save energy in cloud computing*. IEEE Journal on Selected Areas in Communications, 34(5):1728–1739, 2016. 15
- [12] Gigis, Petros, Matt Calder, Lefteris Manassakis, George Nomikos, Vasileios Kotronis, Xenofontas Dimitropoulos, Ethan Katz-Bassett e Georgios Smaragdakis: *Seven years in the life of hypergiants' off-nets*. Em *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, páginas 516–533, 2021. 16
- [13] Mukherjee, Mithun, Rakesh Matam, Lei Shu, Leandros Maglaras, Mohamed Amine Ferrag, Nikumani Choudhury e Vikas Kumar: *Security and privacy in fog computing: Challenges*. IEEE Access, 5:19293–19304, 2017. 18
- [14] Lauren Lee McCarthy, Qianqian Ye *et al.*: *p5.js: Biblioteca gráfica de código aberto em javascript*. <https://p5js.org/>, acesso em 2023-12-05. 24
- [15] Souza, Sthephânia Beatriz Oliveira, Brendha Tomé, Rina Marcia Magnani e Flávia Martins Gervásio: *Descrição do tempo de reação motora nas faixas etárias e gênero*. Revista Neurociências, 29:1–22, 2021. 48