Universidade Estadual de Campinas
Instituto de Computação

Celso Augusto Raposo Lisboa Brennand

Fog Computing-based Traffic Management Support for Intelligent Transportation Systems

Suporte a Gerenciamento do Trânsito Baseado em Computação na Névoa para os Sistemas de Transporte Inteligentes

CAMPINAS
2020

Celso Augusto Raposo Lisboa Brennand

Fog Computing-based Traffic Management Support for Intelligent Transportation Systems

Suporte a Gerenciamento do Trânsito Baseado em Computação na Névoa para os Sistemas de Transporte Inteligentes

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

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor/Orientador: Prof. Dr. Leandro Aparecido Villas
Co-supervisor/Coorientador: Prof. Dr. Daniel Ludovico Guidoni

Este exemplar corresponde à versão final da Tese defendida por Celso Augusto Raposo Lisboa Brennand e orientado pelo Prof. Dr. Leandro Aparecido Villas.

CAMPINAS
2020

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

B75f
Brennand, Celso Augusto Raposo Lisboa, 1980-
Fog computing-based traffic management support for intelligent transportation systems / Celso Augusto Raposo Lisboa Brennand. – Campinas, SP : [s.n.], 2020.

Orientador: Leandro Aparecido Villas.
Coorientador: Daniel Ludovico Guidoni.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Sistemas inteligentes de veículos rodoviários. 2. Sistemas inteligentes de controle. 3. Sistemas de comunicação móvel. 4. Trânsito - Congestionamento. 5. Redes ad hoc veiculares (Redes de Computadores). I. Villas, Leandro Aparecido, 1983-. II. Guidoni, Daniel Ludovico. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Suporte a gerenciamento do trânsito baseado em computação na névoa para os sistemas de transporte inteligentes
Palavras-chave em inglês:
Intelligent transportation systems
Intelligent control systems
Mobile communication systems
Traffic congestion
Vehicular ad hoc networks (Computer networks)
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Leandro Aparecido Villas [Orientador]
Robson Eduardo De Grande
Richard Werner Nelem Pazzi
Islene Calciolari Garcia
Thiago Henrique Silva
Data de defesa: 14-10-2020
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0003-2961-2679
- Currículo Lattes do autor: http://lattes.cnpq.br/1245516173722185

Universidade Estadual de Campinas
Instituto de Computação

Celso Augusto Raposo Lisboa Brennand

Fog Computing-based Traffic Management Support for Intelligent Transportation Systems

Suporte a Gerenciamento do Trânsito Baseado em Computação na Névoa para os Sistemas de Transporte Inteligentes

Banca Examinadora:

- Prof. Dr. Leandro Aparecido Villas
  Instituto de Computação / UNICAMP
- Prof. Dr. Robson Eduardo De Grande
  Department of Computer Science / Brock University / Canadá
- Prof. Dr. Richard Werner Nelem Pazzi
  Faculty of Business and Information Technology / Ontario Tech University / Canadá
- Profa. Dra. Islene Calciolari Garcia
  Instituto de Computação / UNICAMP
- Prof. Dr. Thiago Henrique Silva
  Departamento Acadêmico de Informática / UTFPR

Ata da Defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 14 de outubro de 2020

Dedico este trabalho:
Ao meu pai Geraldo (Campeão);
À minha mãe Sônia;
Ao meu irmão Brenno;
À minha esposa Camilla.
Pelos seus esforços, sacrifícios,
coragem e dedicação que me levaram até aqui.
Pessoas mais amadas na vida.

There are no impossible obstacles;
there are just stronger and weaker wills,
that's all!. (Jules Verne)

Acknowledgements

First of all, I would like to thank God and everyone who helped me throughout this journey that was not limited to obtaining a doctoral degree, but also in life and academic experiences making me a better person both intellectually and morally.

I would like to thank my beloved wife Camilla for her companionship, encouragement, help, and a lot of patience throughout the journey.

I am grateful to my parents José Geraldo Lisboa Brennand and Sônia Raposo Brennand, who always supported me and guided me to study without whom I would not be able to achieve this victory.

To my brother Brenno for his companionship and my grandparents, Anacleto, Angélica, Ana, Heider for their unconditional love.

To all my family members for their unrestricted trust.

I sincerely thank my advisor Prof. Leandro Aparecido Villas for his teachings, encouragement, dedication, and life examples.

To my co-advisor Prof. Daniel Ludovico Guidoni for his advice, discussions, friendship, support, and great contribution to the work.

"Leandro and Guidoni were my guides with torches guiding my way in this incredible and challenging journey. Leandro was in the front and lit my path and Guidoni at the back preventing me from retreating".

To Prof. Azzedine Boukerche for his valuable teachings, for his great dedication to me and my work, and for welcoming me during the sandwich internship at the University of Ottawa, Canada.

To the researchers Guilherme, Felipe, Heitor, Thiago Pedpano, Torsten Braun, Antonio Loureiro, Sandra Avila, Robson De Grande, Eduardo Cerqueira, Luiz Bittencourt and Roberto Sadao for their support, collaboration and friendship.

To my friends, from the Computer Networks Laboratory (LRC), Hernâni, Takeu, Pedro, Helder, Cristiano, especially to Geraldo Pereira and Rodolfo Meneguetti, to friends made at PARADISE Laboratory (UOttawa) and to all other friends and employees from UNICAMP that somehow helped me in this achievement, either through distracting conversations or through collaboration at work.

Resumo

O trânsito nos grandes centros urbanos contribui com problemas que vão desde diminuição da qualidade de vida e segurança da população até o aumento de custos financeiros às pessoas, cidades e empresas. Um dos motivos para um maior tráfego de veículos é o vertiginoso crescimento populacional dos centros urbanos. Além disso, o fluxo de veículos é prejudicado por situações adversas recorrentes nas vias, como o aumento súbito do tráfego durante os horários de pico, gargalos nas infraestruturas de transporte, e acidentes de trânsito. Com o avanço das tecnologias de comunicação, processamento e sensoriamento, os Sistemas de Transporte Inteligentes (ITS) surgem como uma alternativa para mitigar esses problemas. A interoperabilidade dos ITS com novas tecnologias tais como as redes veiculares (VANETs) e computação em névoa, os tornam mais promissores e eficazes. As VANETs preveem que veículos possuam poder computacional e capacidade de comunicação sem fio com outros veículos e com as infraestruturas fixa de comunicação. Assim, uma nova gama de serviços de segurança e entretenimento aos motoristas e passageiros podem ser desenvolvidas. Entretanto, estes tipos de serviços, em especial o de gerenciamento de trânsito, demandam uma análise contínua das condições de fluxo de veículos nas vias e um vasto recurso de rede e processamento, tornando o desenvolvimento de soluções para ITS mais complexo e de difícil escalabilidade. A computação em névoa é uma infraestrutura de computação descentralizada na qual dados, processamento, armazenamento e aplicações são distribuídos na borda da rede, assim, aumentando a escalabilidade do sistema. Na literatura, os sistemas de gerenciamento de tráfego não tratam de maneira adequada o problema de escalabilidade, implicando em problemas relacionados ao balanceamento de carga e tempo de resposta. Esta tese de doutorado propõe um sistema de gerenciamento de tráfego baseado no paradigma de computação em névoa, para detectar, classificar e controlar o congestionamento de tráfego. O sistema proposto apresenta um framework distribuído e escalável que reduz os problemas supracitados em relação ao estado da arte. Para tanto, utilizando da natureza distribuída da computação em névoa, a solução implementa um algoritmo de roteamento probabilístico que faz o balanceamento do tráfego e evita o problema de deslocamento de congestionamentos para outras regiões. Utilizando as características da computação em névoa, foi desenvolvida uma metodologia distribuída baseada em regiões que faz a coleta de dados e classificação das vias em relação às condições do trânsito compartilhadas pelos veículos. Finalmente, foi desenvolvido um conjunto de algoritmos/protocolos de comunicação que comparado com outras soluções da literatura, reduz a perda de pacotes e o número de mensagens transmitidas. O serviço proposto foi comparado extensivamente com outras soluções da literatura em relação às métricas de trânsito, onde o sistema proposto foi capaz de reduzir em até 70% o tempo parado e em até 49% o *planning time index*. Considerando as métricas de comunicação, o serviço proposto é capaz de reduzir em até 12% a colisão de pacotes alcançando uma cobertura de 98% do cenário. Os resultados mostram que o framework baseado em computação em névoa desenvolvido, melhora o fluxo de veículos de forma eficiente e escalável.

Abstract

Traffic in large urban centers contributes to problems that range from decreasing the population's quality of life and security to increasing financial costs for people, cities, and companies. One of the reasons for increased vehicle traffic is the population growth in urban centers. Moreover, vehicle flow is hampered by recurring adverse situations on roads, such as the sudden increase in vehicle traffic during peak hours, bottlenecks in transportation infrastructure, and traffic accidents. Considering the advance of communication, processing, and sensing technologies, Intelligent Transport Systems (ITS) have emerged as an alternative to mitigate these problems. The interoperability of ITS with new technologies, such as vehicular networks (VANETs) and Fog computing, make them more promising and effective. VANETs ensure that vehicles have the computing power and wireless communication capabilities with other vehicles and with fixed communication infrastructures; therefore, a new range of security and entertainment services for drivers and passengers can be developed. However, these types of services, especially traffic management, demand a continuous analysis of vehicle flow conditions on roads. Thereby, a huge network and processing resources are required making the development of ITS solutions more complex and difficult to scale. Fog computing is a decentralized computing infrastructure in which data, processing, storage, and applications are distributed at the network edge, thereby increasing the system's scalability. In the literature, traffic management systems do not adequately address the scalability problem, resulting in load balancing and response time problems. This doctoral thesis proposes a traffic management system based on the Fog computing paradigm to detect, classify, and control traffic congestion. The proposed system presents a distributed and scalable framework that reduces the aforementioned problems in relation to state of the art. Therefore, using Fog computing's distributed nature, the solution implements a probabilistic routing algorithm that balances traffic and avoids the problem of congestion displacement to other regions. Using the characteristics of Fog computing, a distributed methodology was developed based on regions that collect data and classify the roads concerning the traffic conditions shared by the vehicles. Finally, a set of communication algorithms/protocols was developed which, compared with other literature solutions, reduces packet loss and the number of messages transmitted. The proposed service was compared extensively with other solutions in the literature regarding traffic metrics, where the proposed system was able to reduce downtime by up to 70% and up to 49% of the planning time index. Considering communication metrics, the proposed service can reduce packet collision by up to 12% reaching 98% coverage of the scenario. The results show that the framework based on Fog computing developed improves the vehicles' flow efficiently and in a scalable way.

List of Figures

List of Tables

List of Acronyms

ITS	Intelligent Transportation Systems
VANET	Vehicular Ad Hoc NETworks
WAVE	Wireless Access in a Vehicular Environment
CCH	Control Channel
SCH	Service Channel
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
V2X	Hybrid Architecture
RSU	Road Side Unit
5G	5th generation mobile networks
OBU	On-Board Unit
DSRC	Dedicated Short-Range Communications
TCP	Transmission Control Protocol
VLC	Visible Light Communications
GPS	Global Positioning System
eNB	Evolved Node B
IEEE	Institute of Electrical and Electronics Engineers
FCC	Federal Communications Commission
ETSI	European Telecommunications Standards Institute
CAM	Cooperative Awareness Message
QoS	Quality of Service
WSN	Wireless Sensor Networks
M2M	Machine-to-Machine
LOS	Level-of-Service
HCM	Highway Capacity Manual
LTE	Long Term Evolution
MAC	Medium Access Control
IP	Internet Protocol
OSI	Open Systems Interconnect

IDE	Integrated Development Environment
IVC	Inter-Vehicular Communication
ARP	Address Resolution Protocol
IoT	Internet of Things
Veins	Vehicles in Network Simulation
OMNet++	Objective Modular Network Testbed
MiXiM	Mixed Simulator
SUMO	Simulation of Urban Mobility
VCC	Vehicular Cloud Computing
3GPP	3rd Generation Partnership Project
GSM	Global System for Mobile Communications
HSPA	High Speed Packet Access
LiFi	Light Fidelity
WiFi	Wireless Fidelity
iTL	intelligent Traffic Light
EMA	Exponential Moving Average
ECDF	Empirical Cumulative Distribution Function
AoK	Area of Knowledge
MQTT	Message Queuing Telemetry Transport
PTI	Planning Time Index
FOXS	Fast Offset Xpath Service
FOXS-GSC	Fast Offset Xpath Service with HexagonS Communication

Contents

Chapter 1

Introduction

The unplanned development of urban centers is often associated with severe socio-economic problems. Such uncontrolled urban growth typically causes significant stress on city structures due to the unexpected demand of various resources and services. One of the most affected sectors is urban transport systems, in which inefficiencies may lead to many negative consequences. Among them is the increase in greenhouse gas emissions and many hours stuck in traffic congestions, thus resulting in health issues and monetary losses. For instance, the congestion cost in the United States, the United Kingdom, and Germany were almost $461 billion in 2017 [35].

The INRIX 2019 Global Traffic Scorecard [34] shows that drivers from the city of São Paulo/Brazil spend an average of 152 hours per year on traffic congestion and their average speed is 21kph (13mph) while in Rio de Janeiro city, 190 hours are lost in traffic with an average speed of 17kph (11mph). The cities of São Paulo and Rio de Janeiro occupy the 5th and 2nd places in the ranking of the most congested cities in the world. Figure 1.1 shows the rank of the 8 most congested cities in the world, as well as the annual time wasted by drivers and average speed in the traffic.

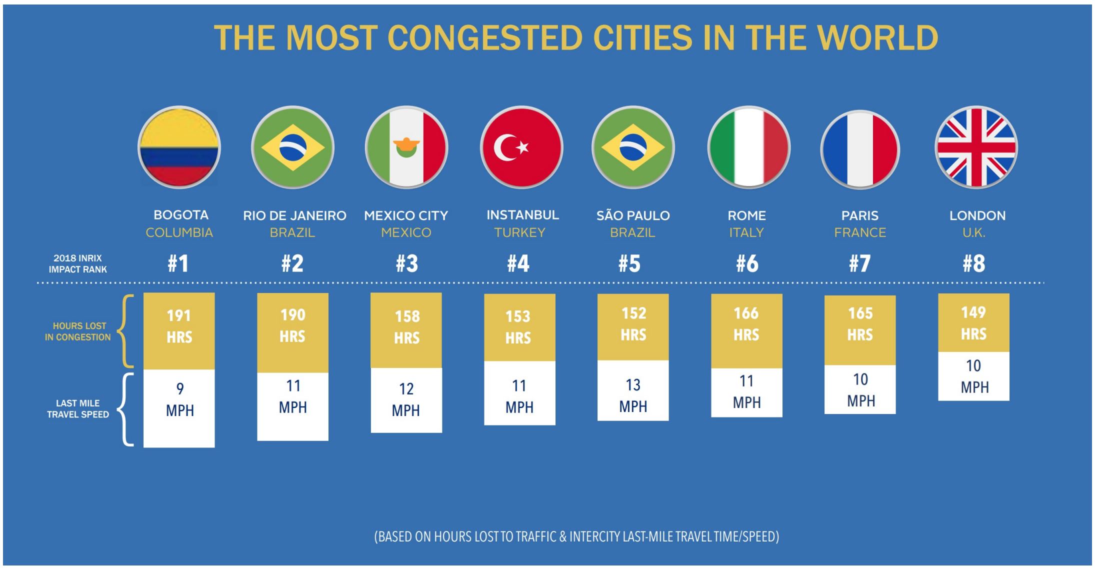


Figure 1.1: Most Congested Cities in the World (2019) (Source [34]) .

This problem is generally generated by the increase in the number of vehicles and also lack of investment in public transportation to support this increase. According to [59], the number of vehicles in the world will reach 2 billion in 2030. This urban growth typically causes significant stress on city structures due to the increased demand for various resources and services, besides serious socioeconomic problems. Urban transport systems, which are an indispensable part of city activities, are one of the most affected sectors [100].

However, it is not always possible to improve the road network due to various factors such as space constraints to build new roads, environmental issues and local city policies. One approach to alleviate these problems is the development of an Intelligent Transport System (ITS). An ITS uses communication, processing and sensing technologies to improve urban traffic and, consequently the flow of vehicles in urban roads. Moreover, an ITS does not only aim to provide traffic management services (for instance, to prevent traffic jam) but also security management services and infotainment applications to drivers, passengers and pedestrians [134, 133, 25, 38, 135].

The basis for an efficient ITS is the collaborative approach where each element of the system such as vehicles, sensors and mobile devices contributes by providing important information to the system [107]. In ITS, vehicles are equipped with sensors (e.g., GPS and Galileo), processors and wireless communication modules. Thus, vehicles can communicate with other vehicles through vehicle-to-vehicle (V2V) communication and with the network infrastructure (e.g., RSU–Road Side Unit) through vehicle-to-infrastructure (V2I) communication. Some ITS applications are road hazard warnings where the driver is warned of possible path problems, such as ice formation on roads, accident warnings, and route suggestion services.

Concerning the improvement of urban traffic, congestion detection and route suggestion services are highlighted as shown in Figure 1.2. This scenario illustrates traffic congestion that occurs on roads. Through VANET communication, traffic jam information occurring in Road A is disseminated to overcome vehicles, so that drivers can view the warning on vehicle On-Board Unit (OBU) screens to take a decision. These applications require communications with rigorous reliability, low latency, and high availability, which are not met by current wireless network standards [81, 38, 135].

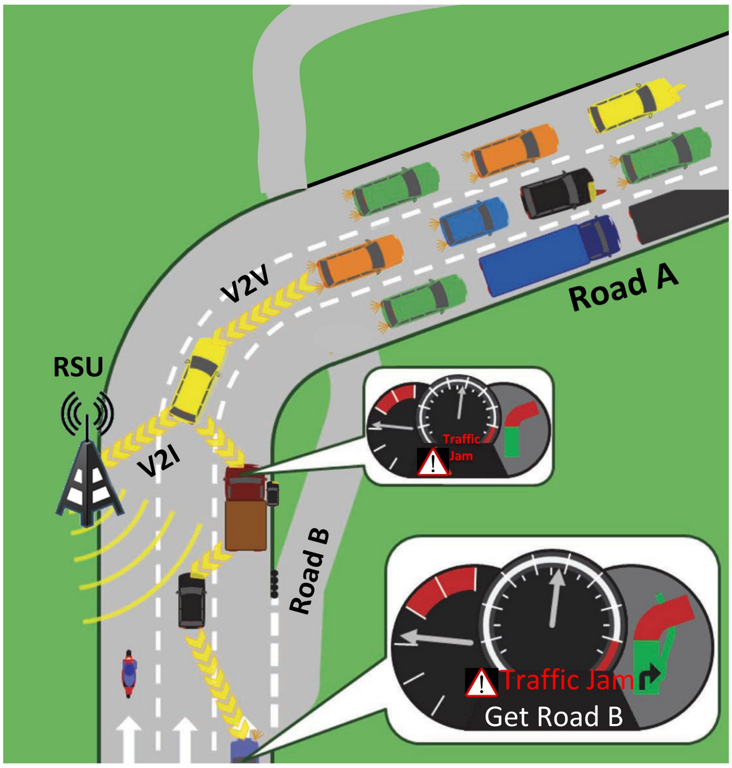


Figure 1.2: ITS Route Suggestion Service (based on [64]).

ITS services and applications have intrinsic characteristics regarding the way they process, store and disseminate a vast amount of data generated in ITS [115, 70, 49]. These characteristics imply in some issues for ITS service requirements such as mobility, frequent network disconnections, networking latency, end-to-end response time and spatial context awareness. Thus, designing ITS services that have a required quality of service (QoS) is a challenging [2, 70, 49]. Traditional centralized architecture is not able to provide these requirements mainly due to the exponentially increasing number of vehicles and new services with very strict constraints of computational and network resources such as route suggestions and accident warning services. Thus, applying the Fog computing paradigm in ITS can reduce its challenges. Fog computing is a decentralized computing paradigm where computing resources (processing/network/storage) are at the edge of the network. Thus, resources are located more logically, efficiently, and close to users/devices.

The main benefits to designing an ITS with Fog paradigm are [20]:

- *Low latency* – some ITS data have strict time constraints, such as data for re-route systems;
- *Predominant wireless access* – modern ITS systems heavily rely on wireless communications;
- *Wide geographical distribution* – ITS has geographically spread sensors. However, the scope of the data gathered is restricted to the location of the sensors that generated such data;
- *Real-time interaction* – re-routing systems have real-time requirements;

- *Mobility* – an ITS is used to optimize the mobility of vehicles in the city. However, the ITS may also leverage mobility to perform data delivery activities to various stakeholders;
- *Scalability* – an ITS needs to be scalable due to the high number of vehicles and sensors;
- and *Extensibility* – if the city grows, the ITS infrastructure also needs to grow to support the expanded region.

These characteristics enable Fog Computing to offer an ideal platform for a highly dynamic and heterogeneous ITS environment [49].

Generally, route suggestion services rely on data from specific regions, such as traffic conditions, which may be irrelevant to other regions of a city [26]. In this scenario, this service may exchange a large amount of data from heterogeneous data sources [115, 139] to monitor traffic conditions in a particular region. Moreover, the data may have real-time constraints and it can be disseminated using different communication technologies [23] and considering the dynamic topology, frequent network disconnections and cooperative communication [5]. It is worth noticing that sending data to a single central entity (e.g., Cloud) is a waste of system resources, such as the network bandwidth. Moreover, data transmissions are more vulnerable to specific problems, such as delays, data loss, scalability, and communication disruption. Hence, route suggestion services in ITSs are not well suited to centralized architectures such as Cloud computing [23, 65, 88, 139, 76, 70].

In this scenario, a route management service that takes advantage of the features of the Fog computing paradigm is extremely desirable in ITS. This happens because the Fog computing paradigm moves its resources (storage and processing) to the edge of the network, thus bringing the available resources as close as possible to end-users without the assistance of the Internet [20]. The Fog computing paradigm is based on entities called Cloudlet which have processing and communication capabilities (e.g., micro-data centers) and are geographically distributed to be closer to the access networks [119]. Since Cloudlet resources are closer to the end devices, they allow a faster response time and a local service decision. Thus, the Fog paradigm provides geo-computation and faster and less costly communication when compared to a Cloud. Although the Fog paradigm has lower computing capacity when compared with the Cloud, it can use Cloud data centers whenever necessary. This approach forms a multi-tier architecture (see Figure 1.3), which is hierarchically organized with varying types of capabilities and end-user proximity. Cloud computing, represented in Figure 1.3 Tier **A**, has a more powerful resource. However, the longer distance to retrieve data and the presence users beyond congested connections due to the use of the Internet often limit the real-time services and increase the network cost, especially considering a high dynamic vehicular network topology. Fog paradigm and Cloudlet environment are shown in Figure 1.3 Tier **B**, where the resources are closer to the end devices permitting a faster response time and a local service decision. Finally, user and sensor devices (e.g., vehicles, road sensors, cameras) are represented in Figure 1.3 Tier **C**.

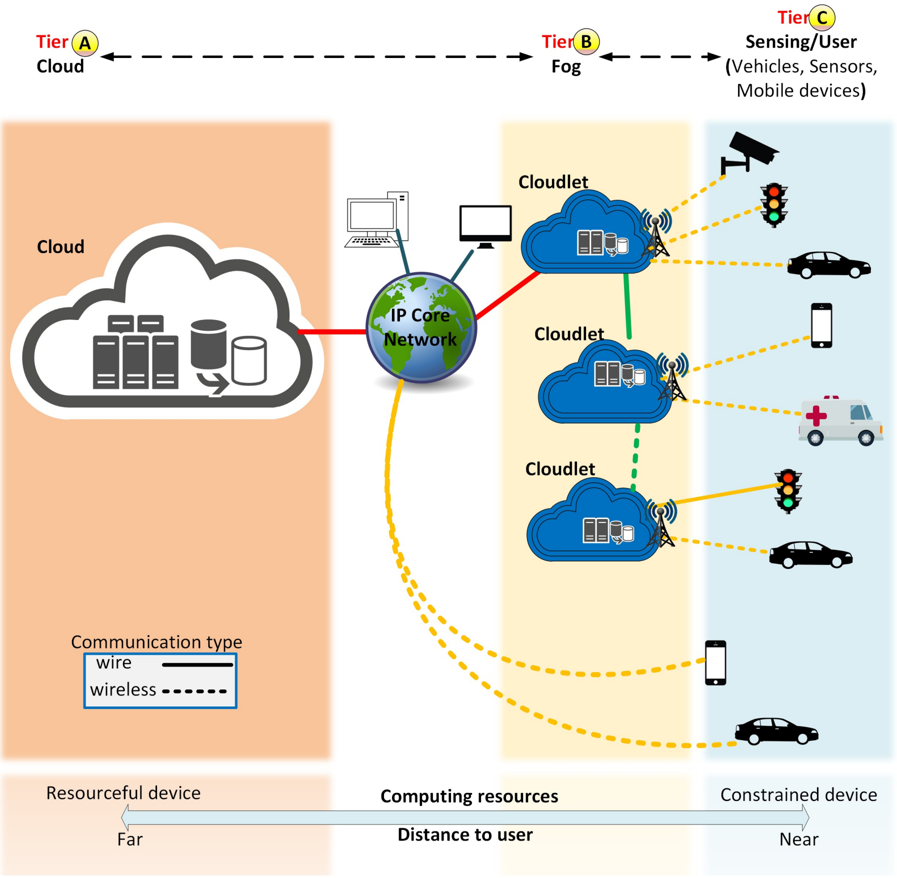


Figure 1.3: Cloud and Fog representation.

1.1 Motivation

In the literature, several studies address the problem of route management in urban centers [105, 46, 7, 93, 148, 44, 146, 21, 57, 85, 69, 138].

Jeong et al. [69] and Pan et al. [105] proposed a centralized solution for traffic optimization where road data is collected and sent to a central server that will classify the traffic and perform routing calculation, thus sending a new route to vehicles. However, as the processing is centralized and the communication uses the Internet, the network latency and system scalability are a problem.

Meneguette et al. [93] and Gomides et al., [57] present a fully distributed vehicle solution in which traffic motoring, classification, and route computation are made by vehicles on the system. However, VANET characteristics, such as data dissemination, hamper theses solutions acquiring full knowledge of the map containing the traffic information and characteristics of the environment (e.g., roads, maximum speed).

Sousa [44] and Younes et al. [148] present infrastructured distributed solutions that use RSUs to classify the traffic and compute the route. However, RSUs do not have full knowledge of the map because the RSU does not have communication between them. Wang et al. [138] use intelligent Traffic Light (iTL) to gather traffic data and as an Internet gateway for vehicles requires a new route to the central server. Thus, having the same problem of centralized solutions.

In most of them, this kind of service uses an architecture for carrying out the monitoring and traffic control that rely on information about the vehicles, as well as the characteristics of the routes. However, these architectures also have to exchange, process and store a considerable amount of data generated by the devices that are embedded in vehicles and that are used for monitoring city traffic. Thus, problems related to processing (e.g., load balance, response time) and data transmissions (e.g., delays, data loss and communication disruption) become a concern. Besides, in route suggestion services, the response time to perform the decision-making process must be within an acceptable time frame so that the information is still useful in order for the vehicle's driver to carry out the necessary route changes.

Given the aforementioned limitations, this thesis proposes a framework solution based on the Fog computing paradigm for Intelligent Transport System services to detect, classify, and control traffic congestion. The proposed framework uses Cloudlets to monitor traffic conditions and to calculate the vehicle route. Thus, it enables computational power to reside closer to where it is most required, thus dividing the system load and increasing the overall scalability of the system and holding the capability to collect, process and store large volumes of data. To do this, the framework uses the network infrastructure Road side Units (RSUs) as a Cloudlet entity, which is deployed in the city to manage the traffic of vehicles. For its operation, a mechanism that gathers all necessary data from vehicles and road sensors was developed. This mechanism optimizes the delivery rate and reduces the number of messages in the system. Therefore, with the data collected by the corresponding Fog entities, the level of congestion of the roads is estimated. Finally, according to the conditions of the roads, the corresponding Fog entities calculate a new route as a suggestion.

1.2 Objectives and Contributions

The general objective of this thesis is twofold. First, we provide a general discussion for traffic management services and the related area of this thesis, which is Intelligent Transportation Systems (ITS), VANET, and Fog computing. The second objective was to propose, design, and evaluate the performance of solutions for traffic management services considering different scenarios.

For the first main goal, i.e., to identify open issues, understand the requirements of ITS traffic management services besides checking the implications of using the Fog computer paradigm in ITS. Objectives:

- Survey the state-of-art about traffic management service, VANET communication technologies, and Fog computing;
- Assess the architecture and methods of traffic management service identified in the literature review;
- Identify drawbacks of current proposals; and

The second main goal is to propose a new traffic management service based on the Fog computing paradigm. The aims are to:

- Design a set of algorithms and mechanisms to increase the packet delivery rate in the VANET environment and the RSU distribution algorithm;
- An algorithm/methodology for traffic classification;
- An algorithm to compute and choose better routes for vehicles;
- An ITS framework based on Fog computing to provide route traffic management services;
- A distributed protocol for multihop communication in VANET;
- Evaluate the proposed approach against state-of-art route traffic management services solutions.

To reach the proposed solution presented in this thesis, the student has published/ submitted a number of papers.

The first article related to the thesis was published in *2015 IEEE Symposium on Computers and Communication (ISCC '15)* [25]. This paper presents the probabilistic algorithm for suggesting routes for vehicles. The algorithm, probabilistically, suggests alternative routes to vehicles in a balanced way, avoiding the creation of new congestion. The service was evaluated in a simulated manner with different conditions and parameters to understand the behavior of the solution.

Although the solution presented reduces congestion, problems inherent to scalability, low latency, and network conditions are not addressed. Thus, article *Fox: A traffic management system of computer-based vehicles fog*[23] proposes using the Fog computing paradigm to address these problems in the route suggestion service proposed. To use this paradigm, the vehicle routing algorithm was redesigned, thus allowing the sharing of information on road conditions between all Cloudlets. Thus, each Cloudlet can execute the algorithm independently and using only information necessary for vehicle routing in its region. In this article, Cloudlets are distributed in the scenario obtaining full coverage. Thus, all communication is carried out with a single hop. Moreover, an evaluation of the traffic quality metrics (e.g., fuel consumption, travel time) and metrics related to the proportion of system users who accept the suggestion provided by the service were made.

The third article published in *2017 IEEE Symposium on Computers and Communications (ISCC)*[22], in addition to improving the routing service, presents an assessment of the communication network and its impact on the proposed service.

The work published in *Sensors, 19(18)*[26] presents the updated algorithm and methodologies used in the routing suggestion service. Among the improvements we can mention, the implementation of the packet scheduling algorithm in the MAC layer to decrease the number of collisions and increase the delivery rate. An algorithm to reduce the number of RSUs required for full scenario coverage was developed. The article also presents the new road classification system based on the Level-of-Service (LOS), thus giving more realistic information about the road conditions to the route suggestion algorithm. Finally, a thorough evaluation using different urban scenarios was presented and compared with several solutions in the literature, thus showing its effectiveness.

Finally, in the study FOXS-GSC - Fast Offset Xpath Service with HexagonS Communication (in the submission process), a service is proposed that uses a hybrid VANET communication and Fog computing paradigm. The main contribution of this work is the development of an efficient communication protocol in which it will allow even vehicles that are far from a Cloudlet to receive and send information about their routes.

Following this line of development, this thesis proposes FOXS. FOXS decreases the drawbacks found proposing a set of algorithms/methods to solve it. The main drawbacks addressed in this thesis are: (i) the traffic jam classification; (ii) the packet dissemination on VANET; (iii) the computation method for the best route suggestion; and (iv) an ITS architecture to provide the route service.

To solve it (i) an algorithm based on LOS was developed. To improve the delivery rate on the network (problem ii), a mechanism for the 802.11p Medium Access Control (MAC) layer was developed to analyze and schedule the packet sent. For (iii) a probabilistic route suggestion that takes into account the traffic jam shift was designed. Finally, for (iv) a framework based on Fog computing created to bring Fog characteristics (e.g., proximity, scalability) to the traffic management service was proposed. This solution is detailed presented in Chapter 4. Evaluations by simulation results and comparisons with literature solutions demonstrated the efficiency of the proposed solution for different scenarios.

To reduce communication and infrastructure dependence of service providers, Chapter 5 presents a vehicle/Cloudlet distributed multihop communication protocol that was implemented in a FOXS solution.

1.3 Outline

The remainder of the text is organized as follows.

Chapter 2 provides an overview of Fog computing, Vehicular Ad-hoc Networks, and Intelligent Transport System. Moreover, the chapter introduces application and service requirement classifications and the ITS flowchart use in the proposed solution. Chapter 3 presents an overview of the literature about approaches to minimize congestion in urban centers. Furthermore, a review of Data dissemination is presented.

Chapter 4 presents the FOXS traffic service and its design and components. The chapter gives a detailed description of the FOXS solution, which is based on the Fog computing paradigm. The chapter also presents the performance evaluation of FOXS, as well as the methodology used and the results. Chapter 5 presents a distributed communication protocol called FOXS-GSC. The chapter describes the multihop protocol and its integration with traffic service. Furthermore, a performance evaluation is also shown. Finally, Chapter 6 shows our final remarks and some possible future research.

Chapter 2

Background

This chapter summarizes the concepts and characteristics of communication technologies and the computing paradigm related to the Fog based framework proposed in this thesis. The Intelligent Transport Systems is also introduced presenting its characteristics, components, and interaction between them.

2.1 Fog computing

In recent years, the number of new types of applications and devices connected to the Internet to communicate and interact has been increasing dramatically. Some examples are sensors and actuators present in industrial/home automation, smartphones, connected/autonomous vehicles among other everyday electronic devices such as refrigerators and televisions. This technological revolution is called the Internet of Things (Internet of Things (IoT)) [9]. These new devices, in addition to requiring more network resources (e.g., bandwidth), demand new requirements such as mobility support, geographic distribution (location matters) and low communication latency. Thus, to support these requirements, a new computing paradigm called Fog Computing (also known as Edge Computing) was developed [19].

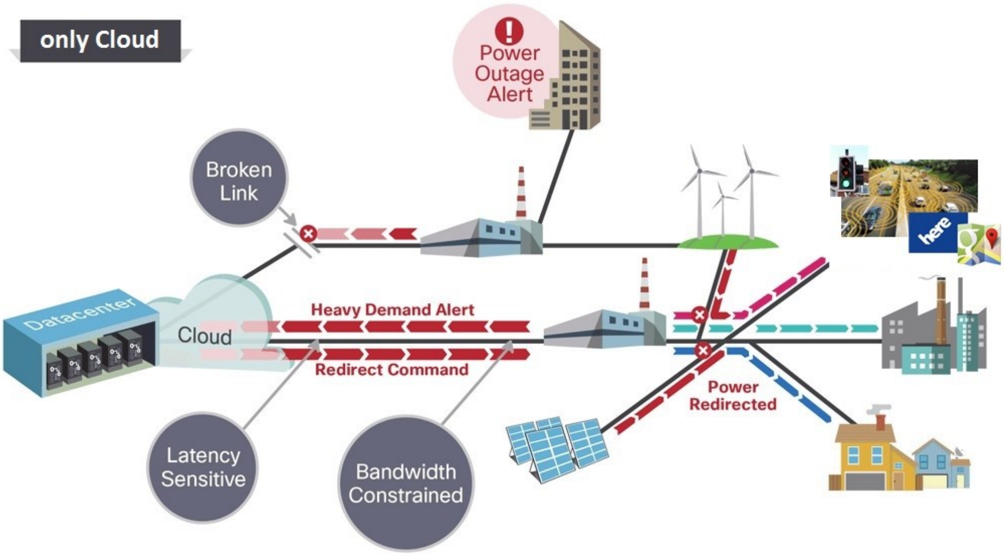
Fog computing extends the Cloud computing paradigm by bringing the processing, communication, and storage capabilities that are located in the core of the Internet to the edge of the network. Thus, improving efficiency and reducing the amount of data transmitted to the Cloud for processing, analysis and storage [20]. In this paradigm, the end devices, also known as Fog nodes, connect directly to the Fog entity, called Cloudlet, which is located on the edge of the network. Devices powered computationally located on the network edge (e.g., routers, servers) can be Cloudlets to provide some service to Fog nodes (e.g., content caching, local service information). This is a significant improvement over the Cloud because the longer the distance to send data, the higher the sending costs on the network is [127]. Moreover, certain physical properties are unchanging, such as the propagation delay. Therefore, the latency increases proportionally with the distance between communicating entities [92]. A delay of hundreds of seconds can cause a disaster in a system that requires real-time analysis such as sensor data in industries or road accident warning services.

As the implementation of Fog is on the edge of the network, unlike the Cloud, it provides geographic data distinction, low latency and lower cost. Thus, implying a general improvement in the Quality of Service (QoS) of the system. Deploying the Fog computing paradigm will not make the Cloud computing extinct, but rather Fog will be a new layer between users and the Cloud working synergistically.

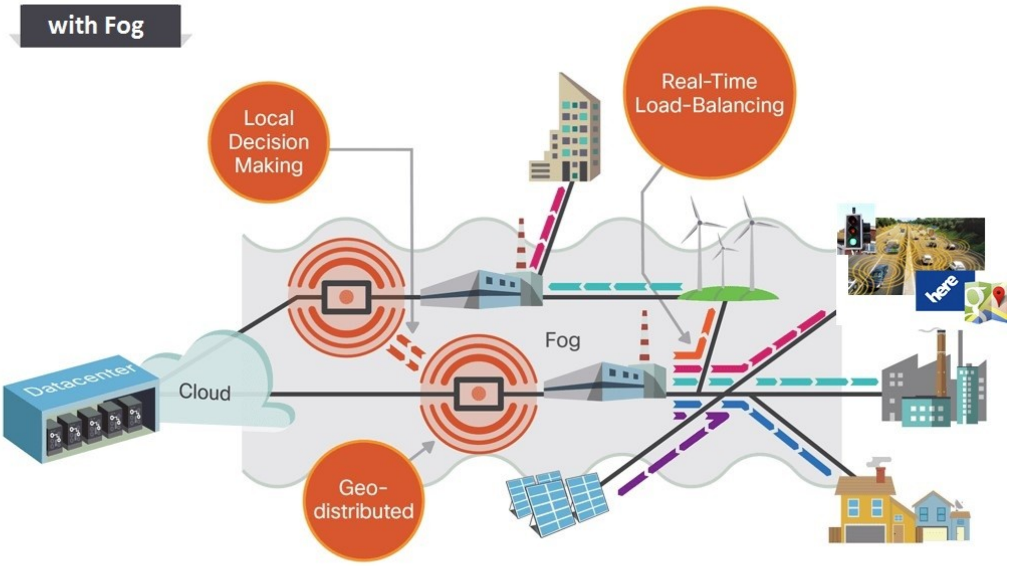
The main characteristics of Fog computing are [20, 2, 145]:

- Location awareness: is supported by Fog paradigm as Cloudlets can be placed in different locations and the Fog connects to them;
- Low latency: it is provided due to the proximity to the end devices;
- Geographical distribution: Unlike Clouds, services and applications are provided by Fog in a distributed manner and can be deployed anywhere;
- Scalability: the distributed nature of Fog computing can work with a wide range of end devices such as a sensor network which monitors the surrounding environment;
- Support for mobility: the ability to connect mobile devices directly allows the use of distributed addressing protocols such as the locator ID separation protocol (LISP);
- Real-time interactions: Fog computing services and applications enable real-time interaction between Fog nodes instead of batch processing used in Cloud computing;
- Heterogeneity: Fog nodes are designed by different manufacturers with different characteristics such as communication type and response time requirements. Therefore, Fog can work on different platforms;
- Interoperability: the ability of Cloudlets to communicate with each other allows Fog components to interoperate and work with different domains and service providers.
- Support for online analytics and interaction with the Cloud: the Fog is placed between the Cloud and end devices performing an important role in the absorption and processing of the data close to end devices.

Figure 2.1 makes a comparison between Fog and Cloud, as well as some Cloud limitations that Fog can solve. Note that in Cloud-only architecture, services with latency and bandwidth requirements can be hampered because the resources are farther from the end device leading to higher latency and limited bandwidth. Already using the Fog computing paradigm, the proximity to with end devices, as well as network load balancing avoids these limitations and still provides other advantages such as increased privacy as the data is in a local domain, and increased system scalability as the service provider (Cloudlet) may be distributed. Table 2.1 summarizes its comparison.



(**a**) without *Fog*



(**b**) with *Fog*

Figure 2.1: Compassion of Fog Computing and Cloud Computing (based on [111]).

Fog computation variations applicable to ITS include the following: the Vehicular Cloud Computing (VCC), where nearby vehicles (e.g., vehicle convoy), form a temporary Fog providing processing, sensing, and communication during its existence [55]; and Park-Cloud, where vehicles parked at a mall or airport form a temporary Fog that can rent or lend resources to ITS services.

Table 2.1: Compassion of Fog Computing and Cloud Computing (based on [111]).

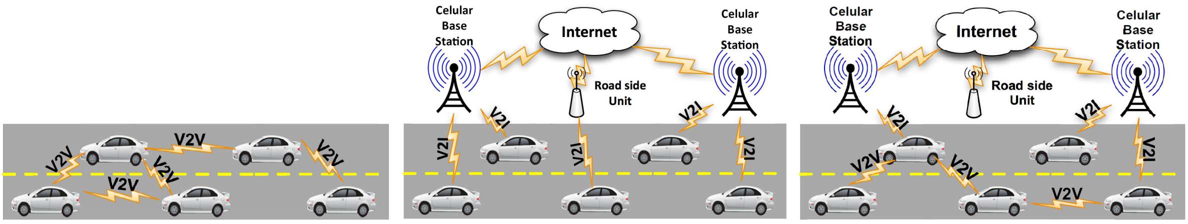
	Fog Computing	**Cloud Computing**
Target User	Mobile users	General Internet users
Service Type	Restricted to local information services related to specific deployment regions	Global information collected from worldwide
Hardware	Constraint storage, compute power and wireless interface	Resourceful and scalable storage space and compute capability
Distance to Users	Near to end-users and communicate through a single-hop wireless connection	Faraway from users and communicate through core networks
Working Environment	Outdoors (streets, parklands, etc.) or indoors (restaurants, shopping malls, etc.)	Warehouse-size building with air conditioning systems
Deployment	Centralized or distributed in regional areas by a local business (local telecommunication vendor, shopping mall retailer, transportation company, etc.)	Centralized and maintained by large companies such as Amazon, Microsoft, Google, etc.

2.2 Vehicular Networks

Vehicular networks (VANET) are a fundamental part of ITS, providing communication between its components, vehicles and infrastructure. Vehicles communicate with other nearby vehicles and fixed infrastructure in three ways: vehicle-to-vehicle (Vehicle-to-Vehicle (V2V)), vehicle-to-infrastructure (Vehicle-to-Infrastructure (V2I)) or hybrid architecture (Hybrid Architecture (V2X)) [135].

- Vehicle-to-Vehicle (V2V): are based on ad-hoc networks [112], V2V allows vehicle-to-vehicle communication directly, without infrastructured networks such as mobile phone base stations or wireless access points. Vehicles can create a spontaneous network while moving on roads and spreading data across the network by routing data between vehicles by multiple hops;
- V2I: Network infrastructure allows direct communication between vehicles and Road Side Unit (RSU) or telephone communication towers (e.g. Evolved Node B (eNB)) with Long Term Evolution (LTE) or 3G technologies. Its advantage is the possibility of integration with other network domains such as the Internet also improving the connectivity. However, the connection may have an operator fee.
- Hybrid (V2X): this combines V2V and V2I. Thus, vehicles can communicate with the infrastructure by single-hop or using multiple hops (through V2V) to increase the network connectivity accordingly with node location. The hybrid architecture allows long-distance communication with the Internet or with vehicles and devices, even when vehicle density is not sufficient to reach a multi-hop gateway.

Wireless communication from VANETs can use a variety of communication technologies depending on the architecture. The IEEE 802.11p standard is used for V2V communication. V2I the vehicles can communicate with the IEEE 802.11p standard or other communication technologies such as LTE, Wi-Fi, 5G, etc. Figure 2.2 presents V2V, V2I and V2X communication architectures of VANET.



(**a**) V2V (**b**) V2I (**c**) Hybrid (V2X)

Figure 2.2: VANET communication architectures (Source [38]).

Some particular features of VANETs are [5, 38, 81, 102, 68]:

- Predictable mobility: due to the limited mobility of vehicles moving in a defined road scenario and the necessity to obey traffic signs and traffic lights, this results in predictable mobility;

- Without power restrictions: unlike other mobile node networks, such as MANETs, vehicles can provide constant power to sensors and computational devices. Thus, they can have more powerful processing, storage, and communication capability;
- Variable Density: VANET density changes according to traffic in the region, where the density is high at traffic jam occurrences and very low on suburban regions or at nighttime.
- Highly dynamic topologies: due to the vehicle's high mobility and velocity, the topology changes fast. Moreover, vehicle connection times vary depending on the vehicles direction travel and the range of the communication interface.
- High computational power: as mobile VANET nodes are vehicles, they can be equipped with various sensors (e.g., GPS, cameras) and devices with high computational power. Thus, it is possible to have more reliable communication and collect information regarding the current position, speed, and direction.

2.2.1 VANET Scenarios

Vehicular network scenarios are generally classified into three groups: urban, rural, and highway [102]. Details of each of them are as follows:

- Urban scenario: The behavior of this scenario is characterized by the reduced speed of the vehicles and the high density of the roads. Consequently, V2V communication is favored in this context. However, this scenario is highly dynamic regarding time and road events. During the daytime, the main characteristic of the scenery is maintained, but at night the number of vehicles decreases, harming V2V communication. Furthermore, road works and accidents can influence traffic behavior, hence impair communication;
- Highway scenario: This scenario is characterized by the absence of traffic lights and higher traffic speed. Vehicles traveling in the same direction tend to form clusters in this way increasing the connection time between them. However, vehicles in opposite directions have very short communication time and all communication between them has to be done in this short time;
- Rural scenario: This scenario is characterized by the low vehicle density resulting in a network with intermittent connectivity. Thus, the network is partitioned into small groups of vehicles forming clusters. Since connectivity is intermittent, developing new ways to disseminate data is necessary as implementing RSUs.

2.2.2 VANET Applications

OBU is a device mounted on vehicles that has processing power, storage, and allows user interaction and VANET communication with other vehicles' OBUs or RSUs. Integrating OBU with a vehicle, which may have many types of sensors and GPS receivers, allows the vehicle to gather, process, and disseminate information about conditions of the environment and itself, allowing the design of a wide range of applications.

VANET applications are classified in three categories [107, 71]: i) Safety applications; ii) Traffic efficiency and management applications; and iii) Comfort/Infotainment applications.

- Safety applications: Accidents involving inattention when crossing an intersection, disobedience of traffic signs and rules, and even those involving wild animals on motorways, cause the loss of thousands of lives worldwide each year. To reduce traffic accidents and consequently reduce injuries and fatalities, the safety application class has been created. This class aims to provide information and assistance to drivers and transients to avoid collisions. Thus, vehicles share information with each other and with RSU in order to predict collisions. Some shared information is vehicle speed, position, and accident alert. Moreover, information exchanged may alert about hazardous locations such as slippery roads or potholes, and can automatically notify emergency authorities (e.g., rescue, police) about any road events.

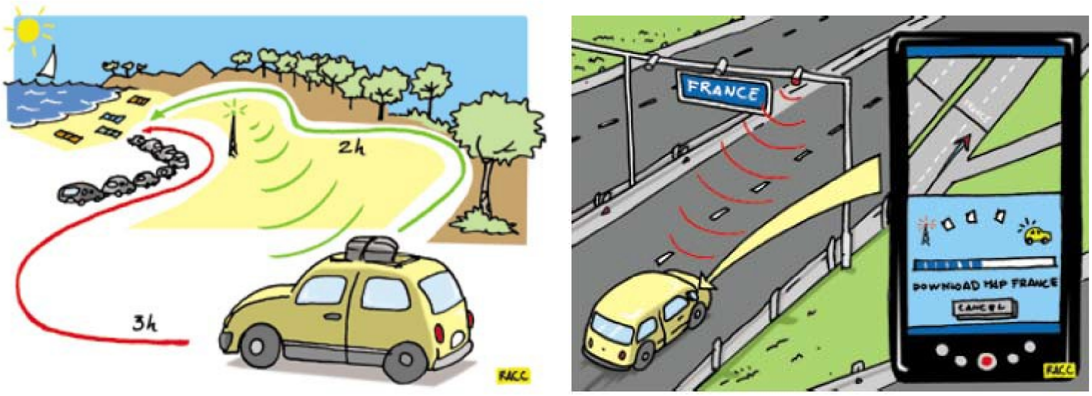
  Figure 2.3(a) presents an accident alert application in actions where a vehicle is alerted about an accident that has occurred on its route so a new route to avoid this is suggested. Figure 2.3(b) a road obstruction is detected by the vehicle ahead and a warning message is broadcasted on the network to alert the ensuing drivers.

(a) Region Routing and Control **(b)** Road Status Report

Figure 2.3: Safety Applications (Source [32]).

  Because the VANET uses a shared communication environment, security applications have priority over any other application.

- Traffic efficiency and management applications: Aims to improve traffic conditions and efficiency by distributing vehicle flow and providing assistance to drivers by disseminating information about traffic conditions, locations, maps, road max speed, and route suggestion. As the drivers need the disseminated information in a restricted time to make decisions during their trip, these classes of applications require high availability. Figure 2.4(a) shows a route suggestion application to help the driver choose a better route predicting traffic congestion and delays. Figure 2.4(b) shows a location context application whereby an RSU sends region map updates and other relevant local information to a vehicle OBU.



(a) Route Suggestion **(b)** Location Context Application

Figure 2.4: Traffic Efficiency and Management Applications (Source [32]).

- Comfort/Infotainment applications: These are applications focused on the comfort and entertainment of drivers and passengers. Vehicles can receive information regarding the point of interest and local business advertising, sharing media files, instant messaging services, tourist information, games, and serving as an Internet gateway to other nearby vehicles. Communication patterns can happen directly among vehicles or between vehicles and RSU. The demand bandwidth varies accordingly with the application, for example, video-conference uses a high bandwidth while business advertising needs less bandwidth. Ideally, the information should be tailored to the users' context. Keeping the context information up-to-date while vehicle-RSU synchronization is maintained is a challenge considering the VANET characteristics such as vehicle mobility [38].

  Figure 2.5 presents an Internet on-vehicle application that can be used by the driver or passengers.

Figure 2.5: Comfort/Infotainment Applications (Source [32]).

Figure 2.6 presents some VANET applications of each category.

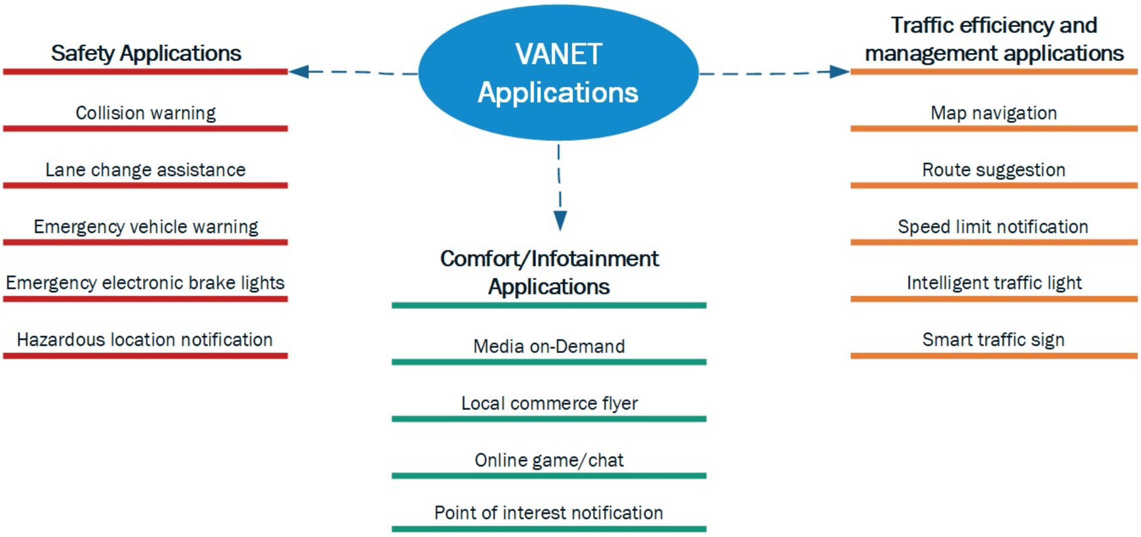


Figure 2.6: VANET Applications per Category.

2.2.3 Data Dissemination

The characteristics of VANETs as a variable density and highly dynamic topology are a major challenge for data dissemination. These characteristics are a problem considering that vehicles usually have short contact time [79]. The network dissemination uses a broadcast mechanism. This mechanism simplifies the communication process since vehicles do not need to know the route and address of the destination vehicles, eliminating the process of routing discovery and topology management [106].

In addition to these problems, each scenario (urban, road and rural) has a different behavior and demands a specific solution. In urban environments, where the density is high, the solution can be focused on broadcast suppression to reduce packet collision that reduces network performance. In rural environments, the store-forward method can be used since this scenario presents a sparse and low-density network [38, 135]. Due to these characteristics, data dissemination has some challenges:

- Broadcast Storm: This occurs when a large number of vehicles attempt to transmit a packet simultaneously. Thus, increasing the network traffic, the network congestion, the packet collisions, and the additional delay to MAC layer control.
- Network partition: This occurs when there are obstacles (e.g., buildings, trees) between the connected vehicles thus, preventing the continuation of the dissemination process.
- Temporal network fragmentation: This occurs due to the high mobility of the vehicles that generate a temporary network partition. Thus, interrupting the packet transmission process until communication is restored.

2.2.4 IEEE 802.11p communication standard

For wireless communication of vehicular networks, the IEEE 802.11p standard is used, which is an evolution of the IEEE 802.11a wireless communication standard [71].

Its main changes were the channel width from 20Hz to 10Hz and the frequency band that was changed from 2.4GHz to 5.9GHz to support the Dedicated Short-Range Communications (DSRC) (Dedicated Short-Range Communications). DSRC is licensed to operate in a frequency band range of $5.85 - -5.925GHz$ and is allowed for public safety applications and private services [72]. The DSRC characteristic is a transmission speed of $6 - -27Mbps$, a coverage range of $300 - -1000meters$ and support for vehicle cruising speeds of up to $200km/h$. As the DSRC is not yet an international standard, in the United States, the Federal Communications Commission (FCC) has allocated 75 MHz of the frequency band of $5.850 - -5.925GHz$ while in Europe, the European Telecommunications Standards Institute (ETSI) allocated $70MHz$ with the frequency range $5,855 - -5,925GHz$. The band frequency is divided into $10MHz$ channels, 6 of which are allocated as service channels (SCH) and 1 to the control channel (CCH) as shown in Figure 2.7. Each channel is assigned to an application type: channels 172 and 174 are dedicated to general applications; channels 176, 178 and 180 are assigned to safety and traffic efficiency applications. Channel 180 is CCH also used to send periodic control messages known as beacon; and channels 182 and 184 are allocated for future applications. Each channel has a time interval of 50 ms and messages have two priority levels, low priority using SCH and a high priority on CCH. When the CCH channel is active, all nodes are required to stop communicating during the CCH interval to receive or transmit a security message in the CCH channel.

Figure 2.7: Multichannel operation in VANETs according to the IEEE 802.11p European standard (Source [51]).

To standardize short-range communication for the vehicular environment (V2V, V2I, V2X), the IEEE 1609 protocol was proposed. The combination of the IEEE 802.11p standard and the 1609 protocol is called Wireless Access in a Vehicular Environment (WAVE) (Wireless Access in Vehicular Environments) [71].

The stack protocol (physical layer and MAC) of WAVE are standardized by $IEEE1609$ where: $IEEE1609.1$ describes the WAVE architecture, its application manager interactions, and format of messages; the $IEEE1609.2$ standard defines the formats of safety messages, processing, and situations that will be used; the $IEEE1609.3$ standard defines transport layer services such as addressing and routing aided by secure data transmission of the WAVE architecture; and the $IEEE1609.4$ standard improves the 802.11p MAC layer to support the operations of the WAVE architecture. Figure 2.8 presents the $IEEE1609$ protocol and its layers.

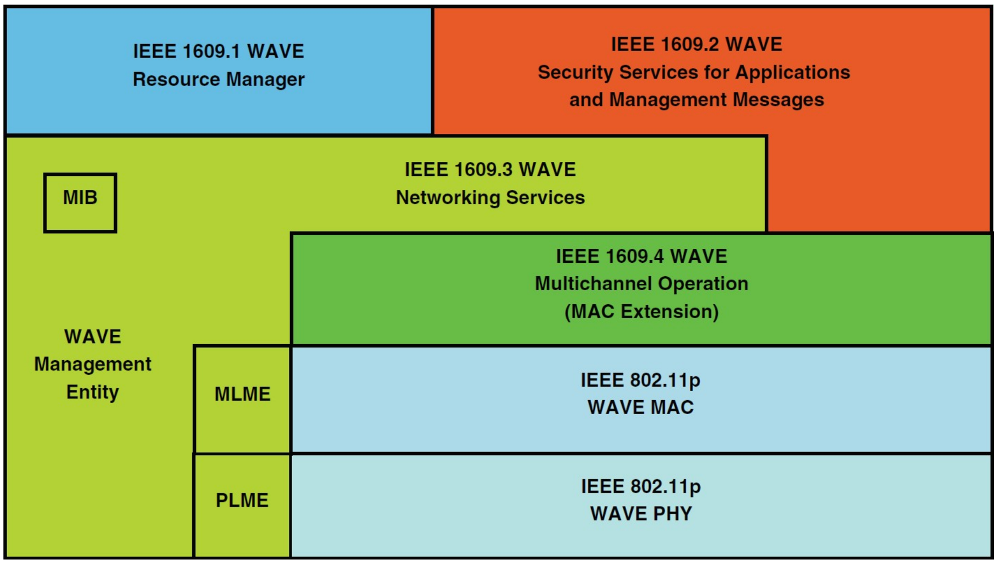


Figure 2.8: The IEEE 1609 (WAVE) reference architecture and relationship to the IEEE 802.11p MAC and physical layers (Source [58]).

IEEE 802.11p synchronization

The WAVE standard has seven non-overlapping channels in the 5.85GHz frequency range, which are the following: six channels as the Service Channel (SCH), used for communication between safety applications, and one channel as the Control Channel (CCH) for safety applications communication. However, the WAVE standard does not require the use of multiple antennas, making it necessary to use a channel hopping scheme. Channel hopping between CCH and SCH is made every T_n seconds, specified as $T_n = 50ms$ by the standard. A guard interval period of $5ms$ is used at the beginning of the channel operation and during this period the channel is treated as busy. Given this, when the MAC layer receives a CCH message to send, but the SCH channel is currently active, this message will have to wait until the CCH channel makes it active. However, this scheme introduces the resynchronization problem.

Figure 2.9 presents the hypothetical situation where two vehicles receive a message (SCH type) to relay from a common neighbor. At time T_1, both vehicles schedule the message to be sent to the MAC layer. Note that Vehicle A schedules the message for T_2 and, when it arrives at the time Vehicle B sends the message to T_3, it would listen to the busy channel and cancel the transmission. However, since the currently active channel is CCH, the SCH message on both vehicles is buffered in the MAC layer to be sent when the SCH channel becomes active again in T_4. Thus, both vehicles transmit at the same time causing a collision.

2.2.5 Other communication standards also used in VANETs

Other forms of wireless communication technologies have been proposed for deploying VANETs. They include Long Term Evolution (LTE) and Visible Light Communications (VLC).

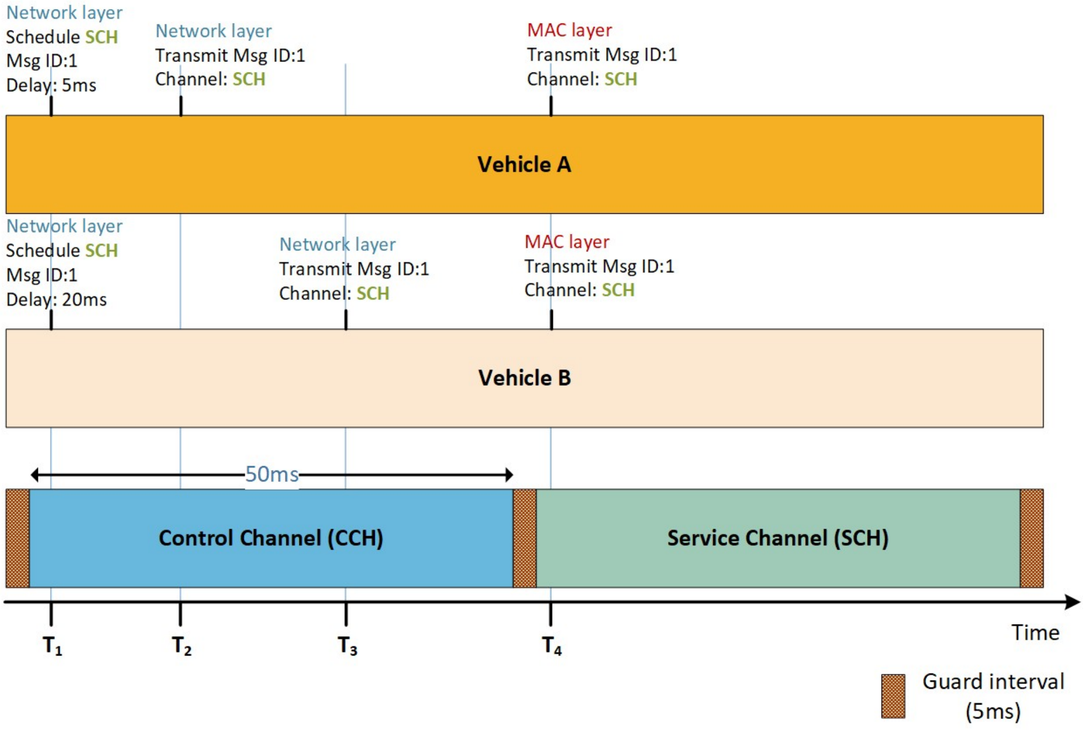


Figure 2.9: The synchronization effect introduced by the IEEE 802.11p MAC layer.

Long Term Evolution (LTE)

The LTE standard is a new generation of wireless mobile communication defined by 3rd Generation Partnership Project (3GPP) [39, 120]. LTE can provide a transmission rate of up to 300 Mb/s, 5 ms latency, mobility support up to 350 km/h, and a communication coverage of up to 30 km. Its radio frequency spectrum is $700--2690Mhz$. Base stations (eNB) spread in the scenario are responsible for managing radio features and functionality, such as mobility, handoff, and interaction with the user equipment. In order for the application to meet the QoS requirements the eNB schedules packets to be transmitted according to data traffic and priority. Another positive feature of the LTE standard is that it is compatible with legacy technology from 3GPP such as Global System for Mobile Communications (GSM) and High Speed Packet Access (HSPA).

Visible Light Communications (VLC)

VLC is a technology that uses the visible light spectrum (380-780nm) for message exchange. Because the visible light spectrum is 10,000 times wider than the spectrum of the radio waves used in Wireless Fidelity (WiFi) [63], its transmission rate can reach speeds of up to 10Gb/s [83]. The $IEEE802.15.7$ [1] standard also known as Light Fidelity (LiFi) (Light-Fidelity) defines the network architectures, physical layer, and access control layer (MAC) of VLC. Its main advantages are low deployment costs and high transmission rates. Its limitations are that the visible light spectrum cannot traverse frosted objects and the reduced performance in outdoor environments [74, 122].

Figure 2.10 presents LiFi communication applied to VANET. Traffic lights and vehicle lights are used to exchange messages between them.

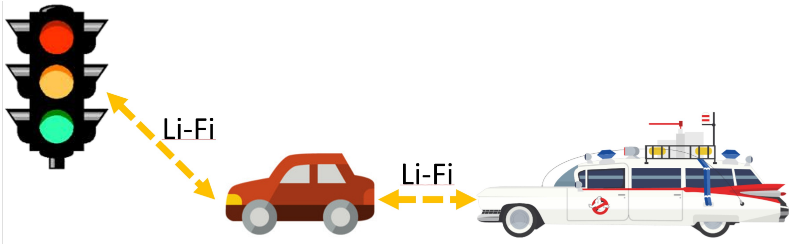


Figure 2.10: Example of LiFi communication applied to VANET.

2.3 Intelligent Transport Systems – ITS

Intelligent transport systems aim at transport management by applying cutting-edge technologies, information and concepts, which interact synergistically. Therefore, improving the safety, efficiency and sustainability of transport networks, reducing traffic congestion and its damage, in addition to improving drivers´ experiences in general [134, 133, 25, 38, 135, 101].

ITS is not only intended to improve vehicle traffic conditions but also intends to make the transport sector safer, more sustainable and efficient by avoiding the inconvenience caused by congestion in urban traffic by improving the management of traffic in city resources such as roads and public transportation also improving people's convenience through Infotainment services. As a result, the vehicle traffic conditions are improved, reducing time spent on traffic jams, reducing fuel consumption and CO_2 emissions, and reducing monetary losses [51].

To this end, ITS relies on collecting real-time vehicle traffic data from various sources such as vehicles (e.g., GPS and speed sensors), network infrastructures (e.g., RSU, eNB), wireless sensor networks (Wireless Sensor Networks (WSN)), surveillance cameras, traffic control systems (e.g., intelligent traffic lights) among others.

Participatory sensor networks (PSNs) are a new source of sensing where people using portable devices such as smartphones and tablets, with built-in sensors (e.g., GPS, temperature) and social applications (e.g., Instagram, Foursquare and Waze) participate as "social sensors" reporting features and events occurring in certain regions [124]. Thus, being able to provide important information to ITS that standard sensors (for example, traffic sensors, speed) cannot provide, such as the occurrence of protests or festive events.

By processing the data acquired by sensors, ITS services can identify and characterize traffic events that generally degrade traffic flow, such as congestion, accidents, infrastructure bottlenecks, work zones, climate, and social issues as manifestations.

ITS services can be classified into two main categories: Comfort and Management Services; and Security Services.

- **Comfort and Management Services:** This type of service aims to increase passenger convenience and traffic efficiency by charting routes to their destination. Some examples of comfort services are Internet access, music downloads, network games, weather information and ancillary services that show you where gas stations, hospitals and information are located about product prices in the region.

- **Security Services:** Services in this category aim to increase passenger and traffic safety in general by exchanging relevant data (e.g., reporting an emergency) between vehicles and between infrastructures. Information can be obtained through sensors embedded in vehicles and by sensors or authorities along the way. This information is disseminated on the network and can be used by vehicle's OBU, drivers and authorities to avoid situations such as an accident or traffic violation. Because ITS services work for the most part in a shared network environment with unsecured and limited bandwidth (e.g., VANET), security services always take priority over comfort and management services.

Figure 2.11 shows an ITS, with its interactions and components.

Figure 2.11: Intelligent Transport System - ITS (based on [73])

In short, ITS services increase overall transport system efficiency by maximizing vehicle traffic flow, improving security by informing critical areas, increasing mobility with traffic balancing, and reducing travel time, thus decreasing monetary losses, and increasing productivity and general happiness.

2.3.1 ITS Workflow

This workflow describes functions required by ITS services (e.g., accident alert, route suggestion, traffic information aggregation), the location of physical entities and subsystems (e.g., the environment, vehicles, a RSU, storage), as well as the data/information streams that connect the physical entities and subsystems. The work presented in this thesis was developed using this workflow to facilitate understanding and future improvement and integration with other ITS services.

Figure 2.12 presents the ITS workflow which is defined in three stages: (i) Data Gathering; (ii) Data Processing; and (iii) Service Delivery.

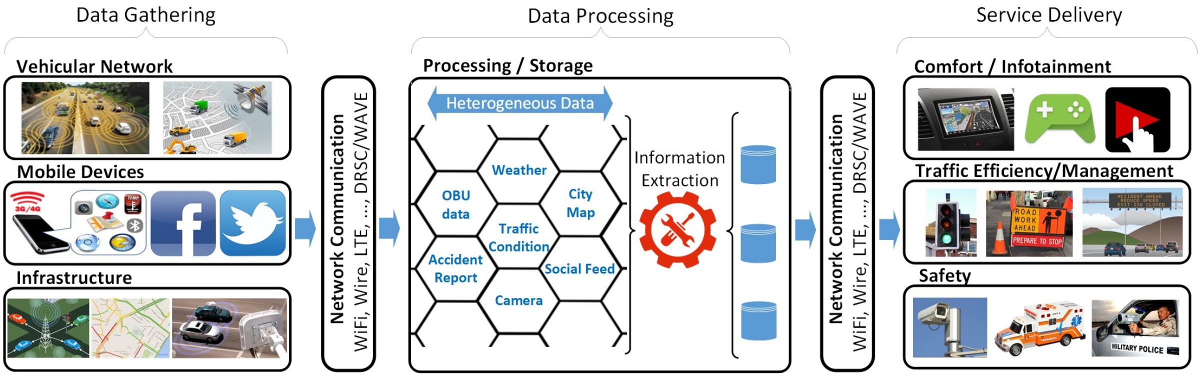


Figure 2.12: The ITS Workflow.

Data Gathering

This stage is responsible for sensing the environment and sending the gathered data to the next stage, Data Processing. For proper functioning of the ITS services, the gathered data must be available when needed, as some information has real-time constraints.

The frequency in data collection, size of the sensed region and heterogeneity of data sources, influence directly the effectiveness of the services [40].

Data can be gathered from several sources as:

Vehicles: vehicles collect data from their sensors, such as speed, odometer, fuel consumption. It is also possible to estimate road traffic conditions, such as vehicle density, through the number of beacon control messages received [14];

Social networks and Participatory sensor networks (PSNs): Data acquired from diverse sources such as mobile networks [77, 53] and social networks [78, 95] increase the accuracy and richness of information that describes the behavior and current situation of the city related to transportation and traffic when compared to the ITS traditional sensing. Some of this data obtained from social networks (e.g., Facebook, Twitter, Foursquare) and from mobile data, may inform events in a certain region of the city such as sports games, accidents, protests, road works, temperature, etc.;

Road Sensors: implementing a good infrastructure for city sensing and data gathering is essential for proper system operation. Sensors such as cameras, radars, as well as roadside traffic sensors that collect traffic images, flow, and average road speed.

Data Processing

At this stage, the gathered data are processed and stored in order to extract useful information, to classify and to aggregate its data accordingly to their characteristics.

Thus, reducing the amount of data transmitted, as well as enabling the production of more consistent information by crossing the correlated collected data [45].

Due to the lack of standardization of monitoring of traffic systems (e.g., vehicles, PSN), the data collected may have different types of metadata, formats, granularity levels, and periodicity of gathering. Therefore, making it difficult to process this data accordingly with individual service requirements [130, 114, 29].

Information Processing (storage and processing) can be done either centrally on mainframes and Cloud or distributed across RSU and Fog entities. These different paradigms can influence in the performance of services due to the limited information acquired, by the processing capacity, or by the proximity of users who use the service.

Service Delivery

In this stage, services are computed and provided to the users based on the information processed in the Data Processing stage. Services can run in any processing entity with network capabilities (e.g., mobile phone, OBU, Cloudlet, Cloud). The service can be designed to run in one of various entities accordingly with its service requirement/specification.

Some services are *traffic management* which orchestrate the components that control city traffic based on the information acquired (e.g., traffic lights, road signs, dynamic road traffic direction), *route management service* where routes are suggested for vehicles to avoid traffic jams and to balance traffic in the city, and *emergency services* responsible for detecting and alerting accidents, suggesting priority routes for emergency vehicles, and other features.

2.3.2 ITS – Route management service

There are three ways of dealing with traffic jam problems: reducing the number of vehicles on the roads; changing the main transportation vehicle e.g. from a car to buses; and distributing the traffic between roads by load balancing. However, as traffic involves infrastructure and people with their customs, it is not always possible to reduce the demand for vehicles, such as reducing the number of trips or increasing the number of roads. Convincing people to change their means of transport is not always possible. However, persuading drivers to travel on suggested routes in order to balance the load between roads is more viable and economical [84].

Traffic balancing can be done by an ITS routing service that captures vehicle and sensor data on the roads. Usually, captured data are the vehicle's current position, average vehicle speed on a certain road, and road flow. Hence, the system can infer current traffic characteristics and calculate new routes for vehicles.

Section 3.3 will address several route management solutions.

2.3.3 ITS Application and Service Requirements

All services and applications require particular requirements such as processing capacity, network delay, and bandwidth. For example, accident detection and alerting services require more network requirements while route suggestion services require a high computational power. Thus, the location of the host service and its network and computational characteristics directly influences on its efficiency.

Table 2.2 shows a classification of the ITS applications according to the following characteristics: *Assistance* is the entity (Cloud, Fog) from which the application will receive (e.g., processing, storage) and require the level to use a given entity (based on [149, 107]); *Resources* are the resources (processing, storage, bandwidth) and the level of use of these by the services and applications (based on [84, 12]); *Constraints* are the constraints (location/region that data has importance, latency, type of communication) of applications (based on [71, 10, 5, 110, 89]).

By analyzing the Table 2.2, security applications require less latency and fewer computational processing resources, since *infotainment* services require more computational processing resources and they are more latency tolerant.

Choosing the best entity (Cloud, Fog) to provide assistance for an application is directly related to Resources and Constraints. For example, accident warning applications should quickly report to vehicles and other entities about the event [71]. The distance from server to the users can be a problem. Therefore, the assistance from the FOG would provide the expected requirements. However, congestion prediction services and data analytics that use the history from months to years to predict a congestion is well suited in a Cloud.

2.3.4 Challenges

The main open challenges in ITS services to traffic management are listed below.

Heterogeneous data integration: Although the ITS enables data integration with different sources to improve its overall performance, this is still an open issue. The main challenge is how to do this integration, since we have many different systems and sources with no integration among them, providing a huge amount of data with no standardization. Furthermore, as emerging technologies such as Internet of Things (IoT) will provide data exchange and communication to a plethora of everyday life devices, it is important to use these devices to turn the data collection paradigm into a new one. However, with this integration, many other challenges will arise including tracking and managing the high number of devices that will be involved in such integration. FOXS uses a data pipeline to structure the data based on information and origin region to a format used for the proposed framework.

Data management and big data issues: ITS needs to handle a huge amount of data. Therefore, a standardization in data representation needs to be employed, once many problems may arise if each source uses an independent measurement and formatting. Moreover, many sources may report its data asynchronously, thus a big challenge is how to manage this issue. Moreover, data correlation is another challenge due to non-integration among different systems and sources, in which the same source may provide data in different systems. In other words, as different systems are independent, the data accounting can incur in false positives. However, the challenge is how to correlate such data to a common source. In addition, ITSs need to provide sophisticated mechanisms

Table 2.2: ITS Application Requirements

Application	**Assistance**		**Resources**			**Constraints**		
	Cloud	***Fog***	***Processing***	***Storage***	***Bandwidth***	***Location Region***	***Latency***	***Communication***
Intelligent Traffic Light	Low	High	Low	High	Low	<500m	<10 Minutes	M2M
Route Suggestion	Low	High	High	High	Low	<1000m	Few Seconds	V2V / V2I
Traffic Detection	Low	High	High	High	Low	<1000m	Few Minutes	V2V / V2I
Accident Warning	Low	High	Low	Low	Low	300–1000m	<100 ms	V2V / V2I
Flyer/ Information	Low	High	Low	Low	Low	<500m	Few Minutes	V2I
Social Network	High	Low	Low	High	High	Single vehicle/ neighbors	<500ms	V2I
Video Stream	High	Low	Low	High	High	Single vehicle/ neighbors	<500ms	V2V / V2I
Online Game	High	Low	High	High	High	Single vehicle/ neighbors	<100 ms	V2V / V2I

to fuse, aggregate and exploit data to deal with different data types provided from heterogeneous sources. However the major challenge is how to exploit these big data issues in a vehicular environment, once the current models and algorithms used in big data are physically and logically decentralized, but virtually centralized [45]. FOXS addresses this problem by using Fog computing, decentralizing data by regions of interest, and fusion the data during multi-hop transmission.

Traffic condition representation and hazard identification: After the data exploitation, the knowledge acquired from it needs to be represented in the correct way to represent the real traffic condition. Otherwise, it may incur in false positives or in the wrong information. By doing this, the key challenge is how to converge so much different information into a single traffic condition representation. In other words, which information has more or less importance to the traffic and how each one will impact the traffic. Providing such representation is still a big issue. In addition, many ITS have been proposed to this with such representation in order to detect traffic hazards [43, 104, 93, 105, 47]. However, many of them are inefficient or they can not identify such hazards as soon as they occurs. Thereby, it uses predefined intervals to try to identify these hazards. However, how is the best-predefined interval to try this identification because with a small interval the ITS service may not receive enough information to identify it. Otherwise, with a great interval, the ITSs may identify the hazard much later than its occurrence. Another issue concerns to the hazards identification process, which one is better to provide the result the appropriate time and which information is used in this process are still big issues. FOXS uses a multi-valued graph representing the roads on the map. The information is sent by the vehicles to the Cloudlets. So, each Cloudlet classifies the traffic of the roads in its region using the Level-Of-Services (LOS) [18].

Alternative route guidance: Suggesting and computing alternative routes to avoid traffic hazards are the best way to improve the overall traffic efficiency. However, the main challenge is how to do this in an acceptable time without introducing an undesired overhead. Consequently, avoiding the vehicles getting stuck in some congestion. Although relying on central entities (centralized approach) to compute and suggest alternative routes to all vehicles is more efficient due to its better management and scenario overview, but depending on the number of vehicles to be re-routed and the complexity of algorithm used in the alternative route computation, such an approach may introduce high overheads degrading its performance. With this problem in mind, one solution is to compute the vehicle alternative route in a distributed way. However, the key challenge is how to provide a full scenario overview of the traffic condition to every distributed computing device to enable them to calculate an efficient route without overloading the network. Another concern is how to compute an efficient alternative route without incurring in traffic congestion in other areas in the nearby future, providing a better traffic balance and management. Thus, having a good alternative route guidance trade-off between efficiency and complexity is essential. FOXS implements a decentralized, scalable system that meets the real-time requirements for this type of service. FOXS monitors and controls congestion by suggesting alternative routes to the vehicles.

Security and privacy: Ensuring the information privacy and security in ITS is essential for all involved people, transit agencies, government, and so on as the data may contain personal information and can track people and vehicles [52]. One key challenge is the action of malicious entities which can add or change messages generated by services generating issues such as fake warn messages. One building block of ITS is the VANET [109, 33, 38]. According to [108, 113], for providing security and privacy to the VANETs, several requirements need to be satisfied. Verification of data consistency, that checks the legality and consistency of messages to avoid messages with malicious data. Availability, to ensure continuous operation of the system even under attacks (e.g. DoS by jamming). Real-time constraints, focusing on maintaining communication and computing efficient even with the usage of security techniques. Authentication, legitimizing messages. Furthermore, the new trend of using Cloud computing with ITS increases the complexity for providing security to the system because the inherent security problems in Cloud computing are also added to ITS [144]. FOXS uses the Fog computing paradigm, so this problem is reduced due to its characteristics, since its proximity to users allows data to be distributed and that most data does not have to be transferred over the Internet as in the Cloud paradigm.

Communication: ITS services have several challenges related to communication, mainly regarding integrating the VANET with many other wireless communication technologies such as 4G/LTE, Bluetooth (in-car), Zigbee (in sensors) and traditional WiFi (IEEE 802.11). Seamless communication is one of the open issues for ITS services [123, 38], due to: the large volume of data; the mobility of connected devices (vehicles, smartphones); the heterogeneity of nodes (cars, motorcycles, trucks, buses) with different communication technologies; and the services with different requirements (latency, jitter, packet loss). To reduce these problems, FOXS uses fog computing. Also, FOXS implements a message scheduler to increase the delivery rate of messages and a multi-hop protocol aimed at the vehicle routing service.

RSU Distribution: The Road Side Unit (RSU) increases the bandwidth and the communication reliability for VANETs being an essential component for ITS services such as information and traffic management services [118]. The coverage of the whole city is desired by RSUs and for this, generally, the RSU are installed at intersections. However, this is not a good approach because of the high cost of implementation [13]. Techniques to optimize the deployment of RSU in the city taking into account the quantity of RSU versus environment coverage is desired and is a challenge [31, 16, 30]. FOXS implements an algorithm based on hexagonal binning that takes into account the area of coverage of the RSU. Thus, reducing the number of RSUs necessary to cover the entire environment.

2.4 Final Remarks

This Chapter described the key aspects of Fog computing, Vehicular Ad-hoc Networks, and the Intelligent Transport System that are the key computer networking fields on which this thesis bases. Also presented is the classification of requirements for ITS applications and services, as well as the ITS flowchart that is used as the basis for the traffic management solution presented in this thesis, as will be explained in Chapter 4.

Chapter 3

Related Work

This chapter presents the literature review of works related to this thesis. It is divided as follows: (i) ITS architectures and communication paradigm, (ii) Data dissemination solutions, and (iii) Route management services.

3.1 ITS architectures and communication paradigm

The new breed of emerging communication technologies and computing paradigm as well the new kinds of ITS services require different ways and patterns to system development. Thus, a framework capable of providing the use of these new technologies with their characteristics in the development of ITS services is desirable.

With this intention, some works proposed a cloud computing-based architectures to take advantage of large-scale computation [56, 87, 67, 17, 60]. These architectures are subdivided into three layers: *(i)* the layer with the temporary *Cloud* composed by *Vehicular Cloud Computing-VCC* and *ParkCloud* for short-scale computing; (ii) the communication layer with Internet gateway function, formed by RSU and base stations; (iii) the layer with the *Cloud* for large-scale computing. The major drawbacks of these approaches are the need for a high and constant communication band, a characteristic not always found in VANET communications. Another is a high and inconsistent latency since the Cloud is located on the Internet, making service quality infeasible for some services in particular services that need a low latency as the accident alert. The proposed framework implements a layer with Fog computing capability, thus eliminating these problems since Fog is located in devices closer to end-users and able to use various communication technologies (e.g., Dedicated Short-Range Communications (DSRC), LTE, WiFi).

Wang et. al. [139], presented an architecture based on Fog computing for data synchronization. The proposed architecture has three layers Cloud, Fog, and devices where Fog is used to reduce the cost of the network traffic, latency, and share its storage and processing power with devices and Cloud, thus reducing the computational load. Following the same concept, the works [149, 117] propose using RSU-Cloudlet or micro-datacenter besides the use of Cloud. This RSU-cloudlet is defined as a *Cloud* on a smaller scale located in RSU. The proposed architecture has three layers: *Central Cloud* used for large-scale computing; *RSU-Cloudlet* used for short-scale computing of events occurring in the communication coverage region; and VCC used for short-scale computing in a cooperative manner with other vehicles (e.g., sensing). The drawback of this proposal is that the *RSU Cloudlet*, which uses DSRC, only provides resources for users who are in their radius of coverage with direct communication, therefore to provide full coverage it is necessary to deploy RSU throughout the city. Our framework allows the creation of regions of interest with an area larger than their coverage radius and is capable of using different communication technologies (e.g., LTE, 802.11p), allowing a greater penetration into the system and the possibility of load balancing and a better resource distribution, thus increasing the quality of ITS services.

Mario Gerla [55] exposed the concept of Vehicular Cloud Computing (VCC), where a set of oncoming vehicles traveling in convoy, for example, can offer their idle resources to other entities. The VCC is a good resource to help building the sensing regions and provide additional funding for infrastructure. However, this is a spontaneous resource, formed or undone as dynamically as the density of vehicles in the region. Whaiduzzaman et al. [142] present the state of the art and taxonomy of VCC.

Many papers envisage using other communication technologies such as WiMAX [99, 8, 48] and LTE [97, 27, 6, 143] to meet the limitations of the IEEE 802.11p standard. Some of the limitations are low communication range [99], low bandwidth, high latency when compared to other technologies such as LTE, and not providing Quality of Service (QoS) desirably for some ITS services (e.g. video streaming) [97]. Due to the diverse requirements of the wide range of ITS services, ranging from security services that require low latency to infotainment services that require more bandwidth. Choosing, dynamically, which communication technology best fits the service requirements, would avoid some problems caused by the limitation of the technology used and increase the general system performance. For example, Vinel [136] analytically demonstrates that when vehicle density is high, services as emergency/security that use periodic messaging (e.g., beacon) in a short period like $100ms$, the LTE technology would not be able to meet this requirement. Therefore, using multiple communication technologies, as is possible in the Fog framework proposed, simultaneously to exploit the strengths and address the weaknesses of the other would be a good option. Thinking about this integration of technologies, Mezghani et al. [96] propose an opportunistic communication method where vehicles that own only V2V communication, get access to Internet content through another vehicle that has LTE communication (called Seed). This bridge vehicle is chosen by criteria proposed by the authors.

3.2 Data dissemination solutions

Generally many ITS applications use Vehicular Ad Hoc Networks to send messages from a source (vehicle, RSU) to all vehicles located inside a geographic region [38]. Such activity is known as data dissemination. Two important challenges to develop a data dissemination solution are the broadcast storm and the intermittently connected network. The broadcast storm problem happens when many closer vehicles try to communicate at the same time, so increasing drastically the message collision [128, 121]. The intermittently connected network problem happens when the numbers of vehicles are reduced (e.g., daybreak, holidays, and rural areas), compromising the data messages dissemination in multi-hop communication model [137, 116]. Besides these challenges, each application has specific requirements, which demands different strategies to support the data dissemination. Many data dissemination solutions have been proposed to address these challenges [128, 121, 137, 116, 38, 135, 37, 94, 36].

Tonguz et al. [128] present the Distributed Veicular Broadcast (DV-CAST), a data dissemination solution that proposes to solve the broadcast storm and the network partitions problems. DV-CAST uses periodic beacon messages to build the local topology (one-hop neighbors) that is used to rebroadcast a message. DV-CAST performs data dissemination in both sparse and dense networks focused only on highway topologies. In a dense network, the receiver applies the broadcast suppression algorithm but in sparsely it uses the store-carry-forward algorithm. However, DV-CAST performance depends on the well-tuned beacon frequency.

UV-CAST (urban vehicular broad-cast protocol) [137] is proposed to perform data dissemination in different traffic conditions. UV-CAST uses broadcast suppression or store-carryforward depending on vehicle relative position with the sender. If the vehicle is on the edge of a connected component, the UV-CAST infers that it has a higher probability of meeting new neighbors. Thus, the received message is stored until rendezvous a new one neighbor. However, if the vehicle is not a border vehicle, the broadcast suppression algorithm to rebroadcast the message is used.

Villas et al. [135] propose the Data Dissemination Protocol in Vehicular Networks (DRIVE) a new solution to perform data dissemination in VANETs, considering dense and sparse network scenarios. The DRIVE relies exclusively on local one-hop neighbor information to deliver messages in these scenarios. It uses the predictability of vehicle mobility to create a preference zone, because in addition to the distance of the transmitter, it sets the delay on the message retransmission. Furthermore, the solution employs implicit acknowledgments to guarantee robustness in message delivery under sparse scenarios. However, the number of transmissions is increased by acknowledgment messages.

Meneguette et al. [94] proposed a data dissemination protocol called Autonomic Data Dissemination in Highway for VANETs (ADDHV) to provide greater coverage and a low delay in the data dissemination independently of the region traffic condition. The authors create two mechanisms to mitigate the broadcast storm and the network partitions problems. For the broadcast storm problem, a mechanism based on [135] was used which defined regions called sweet spot in which vehicles inside them have a higher priority to message rebroadcast and a lower delay. For the network partitions problem, an autonomic computing technique to decide whether the vehicle should disseminate a packet or not was implemented. This choice is based on a propagation efficiency concept and on the geographic localization of vehicles. However, ADDHV can introduce network overhead because of the store-carry-forward mechanism applied in the solution.

Akabane et al. [4] developed the Context-Aware Routing pROtocol (CARRO), which explores the geographic context knowledge for data dissemination in VANETs on urban and highway scenery. CARRO protocol, as DRIVE [135] solution, selects vehicles located in a high-priority geographic region to disseminate a message. To create a network topology knowledge about the neighboring vehicles at 1-hop, each vehicle, periodically, transmits beacons with its position, velocity, and direction. When CARRO detects the network partition problem, the store-carry-forward mechanism is used. However, this mechanism can increase the network delay. Moreover, the high number of beacons necessary for solution work can increase the overhead.

Cunha et al. [36] introduced the Clustering Coefficient and node DEGREE protocol (CC-DEGREE), which uses two social metrics to choose the most reliable vehicles to retransmit the message. The fist metric is the clustering coefficient, which is defined by the number of connections between the vehicle neighbour divided by the total number of possible connections between vehicle neighbours. The second metric is the node degree, which is defined by the number of one-hop neighbours. The CC-DEGREE protocol operates independently of the road traffic density. However, the CC-DEGREE computes the clustering coefficient metric based only on the position of an individual vehicle, thus a low variability for dense scenarios and impacting the retransmission nodes choose. This characteristic causes similar waiting time assignments to several vehicles, increasing the message collision probability.

Costa et al. [37] proposed a data dissemination protocol based on complex network metrics, named DDRX, for VANETs in an urban scenario. In DDRX, vehicles maintain a local knowledge of its 1 and 2-hops neighbors, which will be used to build a subgraph. Using complex metrics (i.e., betweenness centrality, and degree centrality), DDRX selects the best vehicles to retransmit the message. DDRX provides data dissemination with low overhead and delay, maximizing coverage, and minimizing the number of packet collisions.

3.3 Route management services

This section presents related works that address the problem of route management, i.e., traffic congestion management of vehicles in urban centers. In the last years, such a problem has been explored by several works [85, 86, 105, 46, 7, 42, 44]. However, we have not found works that address adequately this problem through the Fog computing paradigm to improve system performance, such as the load balancing (network, processing), the response time and the network load. For simplicity, these sections group the solutions accordingly with the route processing architecture (e.g., route decision, route computation, and traffic road classification): Centralized solutions 3.3.1: presents solutions where all computing is performed on a centralized infrastructure; Vehicle distributed solutions 3.3.2: presents solutions where computation is performed in a distributed way among vehicles; Hybrid distributed solutions 3.3.3: shows solutions where some steps are distributed between vehicles and others are performed on the infrastructure; and Infrastructure distributed solutions 3.3.4: presents solutions that use distributed infrastructure to perform processing.

3.3.1 Centralized solutions

The system proposed in [105] is responsible for traffic monitoring and vehicle re-routing to decrease the traffic congestion of vehicles. The goal is to reduce the driver's travel time, as well as the CO2 emissions and fuel consumption of vehicles. To this end, real-time data about vehicular traffic conditions, such as position, speed, and direction are gathered by a centralized system through vehicle-to-infrastructure (V2I) communication. Four steps are periodically executed in the traffic care system: (a) *Data collection and representation*, which describes the network using a directed graph, in which the weights are the average travel time; (b) *Congestion prediction* is the service that periodically checks all road segments to detect signs of congestion; (c) *Selection of Vehicles to Be Rerouted* selects candidate vehicles near the congested roads; and (d) *Choose alternative routes* for each previously selected vehicle. The authors employ three strategies to calculate new routes: (i) Dynamic Shortest Path, which calculates the route with the lowest travel time; (ii) Random k Shortest Paths, which selects the k lowest travel time path routes and assign, at random, one of them to the vehicle; and (iii) Entropy Balanced k Shortest Paths, which is an enhancement of Random k Shortest Paths, in which it is considered the impact of the selected road on the future density of the road. However, these strategies have the following drawbacks: (i) congestion in other places due to the suggestion of routes to the same area; (ii) long routes can be selected to reduce the traffic of vehicles in another area; and (iii) the use a central server requires a substantial computational resource and network communication, so the use of a central server is not salable.

Jeong et al. [69] proposed a cloud-based system for traffic optimization called the Self-Adaptive Interactive Navigation Tool (SAINT). In this system, vehicles report the road traffic conditions to the traffic control center hosted in the Cloud. RSU and eNodeB (from cellular network) require Internet connections to communicate with the Cloud, thus vehicles are equipped with 802.11p and 4G. To reroute vehicles, SAINT uses a modified Dijkstra's algorithm where the weight function takes into account the vehicle's delay to reach the roads of the route. Thus, the probability of a route becoming very popular and causing a new congestion is reduced. However, such a solution has some limitations. For instance, vehicles must continuously inform the conditions of the routes through the Internet connection to the Cloud. Another limitation is the DSRC communication used by vehicles since it does not have any mechanism to work correctly in high-density scenarios, such as urban centers.

The work [61] presents a vehicular traffic management service called Re-RouTE. Re-RouTE uses a traffic engineering theory to classify road congestion based on road density. Periodically, vehicles send their location information to a centralized management service to classify and calculate a new rote to vehicles when necessary. The vehicle can communicate through V2X using 802.11p protocol to vehicle/RSU communication or using the 5G cellular network infrastructure. The Re-RouTE service is divided into four modules/tasks: (i) Location Information, where is received the vehicle's information to classify the roads; (ii) Network Representation, where is created the weighted graph base on road density; (iii) Network Classification, where the road is classified into congested or not congested; and (iv) Route Suggestion, where the alternative route is computed. AS FOXS the load balance is made upon road graph updated with new road conditions. However, FOXS also uses the probabilistic method to provide a more effective load balance during the route calculation step.

The work [140] proposes a framework called CDRAM (Content Dissemination framework for Real-time trAffic Management) that enables real-time communication based on heterogeneous network access in IoV systems. CDRAM consists of three main components: RSU, that is used to receive traffic data gathered by vehicles and retransmit to a central server; cellular Base Station, that is used when RSU is not in the coverage area; and a traffic management system(TMS), that is responsible for detecting congestion spot and suggesting new routes to vehicles. The CDRAM framework applies a delay-sensitive routing algorithm for message dissemination who chose the next hop and communication interface to reduce the transmission delay. However, differently from FOXS, the CDRAM uses centralized traffic management and does not have any traffic balance.

3.3.2 Vehicle distributed solutions

Meneguette et al. [93] proposed a solution, named INCIDEnT (INtelligent protocol of CongestIon DETection), based on an Artificial Neural Network (ANN) to estimate congestion level and maximize the urban traffic flow. The ANN uses the average speed and the density of vehicles on the road as the input of the system to classify the traffic and suggest new routes for drivers. The congestion is classified into three levels: Free; Moderate, and Congested. Finally, the classification data are disseminated by all vehicles on the road through periodic beacon messages. When a vehicle receives a message about a road congestion level, the ANN can decide whether to keep its current route or calculate an alternative route. However, the solution does not have full knowledge of the map neither a method to avoid the overlapping routes, which can in turn generate a new traffic jam. Another problem detected is that it does not implement any broadcast suppression mechanism, thus decreasing its efficiency, especially in a high-density scenario.

A distributed traffic management system, called SGTD, was proposed by Gomides et al [57]. The system in a distributed way, uses only vehicles and their on-board sensors to collaboratively with neighboring vehicles classify the traffic conditions. The traffic conditions are inferred by each vehicle data (e.g., average speed traveled on the road) and disseminated data from neighbour vehicles. Hence, when the vehicle reaches a road intersection, it is checked if there is a faster route. SGTD also decreases the number of disseminated messages on the network as it limits messages sent to a limited region. However, SGTD has only knowledge of a specific region and as vehicles have similar road condition information, new congestions may occur.

Garip et al. [54] proposed a V2V distributed congestion avoidance mechanism. The mechanism disposes checkpoints to all vehicles between their initial position and the destination position. When vehicles approach the next checkpoint, vehicle information about the roads (average speed) is sent via V2V communication. This information is used to calculate the best route between the current checkpoint and the next vehicle checkpoint. One disadvantage of this solution is that the proposed system is not able to generate good traffic congestion information once it uses only near road information.

Sousa et al. [41], propose a V2V distributed traffic congestion service aim to reduce the traffic jam drawback with low communication overheads called DisTraC. DisTraC discretizes road congestion in 10 levels according to the vehicle speed. Periodically, the vehicles disseminate the current road level to other vehicles using a proposed broadcast suppression protocol. Therefore, periodically the vehicle re-calculates its route using the knowledge acquired. However, this solution has some drawbacks like the lack of traffic balance mechanism, the limited traffic knowledge, and need for a high number of messages to work.

Akabane et al. [3] propose an infrastructure-less system for traffic management called dEASY (distributed vEhicle trAffic management SYstem). dEASY implements three architecture layers: environment sensing and vehicle ranking, knowledge generation and distribution, and knowledge consumption. The environment sensing and vehicle ranking layer applies a vehicle ranking mechanism to choose the vehicle head to concentrate all nearby knowledge generated in the next layer, knowledge generation and distribution. In the last layer, knowledge consumption, each vehicle computes an altruistic alternative route based on congestion knowledge and the neighborhood route information disseminated by the previous layer. As FOXS and FOXS-GSC, dEASY has a traffic balance mechanism named as altruistic route. However, as other fully distributed solutions, knowledge of congestion in the city is more limited than a central service and although dEASY needs fewer messages than other fully distributed solutions, it still requires a large volume of messages to be exchanged for its operation.

3.3.3 Hybrid distributed solutions

Doolan et al. [47] proposed a VANET routing solution, named EcoTrec, aimed to reduce the CO2 emission without significantly affecting the travel time. For this, each vehicle periodically disseminates data about its fuel consumption, current route, and average road speed. Thus, EcoTrec determines the road conditions, and in a distributed way, each vehicle calculates a new route. To avoid the various vehicles always attributing the same best route, EcoTrec randomly assigns the second-best route to some vehicles. The EcoTrec architecture has three main parts: (i) VehicleModel with the vehicle and embedded sensors characteristics; (ii) a RoadModel with a road representation and characteristics that are allocated in the central server; and (iii) a TrafficModel with the traffic condition based on the VehicleModel and RoadModel characteristics. However, all vehicles in the system send messages to the neighboring vehicles and the central server to update the TrafficModel. Moreover, the vehicles compute their route based on the TrafficModel received from the server. Thus, the scalability of the system is compromised by a large number of messages exchanged.

Pan et *al.* [104] proposed DIVERT, a distributed congestion avoidance which offloads the rerouting computation in each vehicle providing privacy to the drivers and enabling a potentially real-time congestion avoidance. To this end, it relies on a central server (e.g., Cloud) to build global traffic knowledge which is further reported to the vehicles. Furthermore, vehicles report its position and receive global traffic knowledge through onboard devices (e.g., smartphones) using cellular network such as 3G/LTE. Thus, when the central server detects signs of congestion, it notifies vehicles that are near the congestion and they notify its neighbors through a limited hop flooding manner, enabling them to verify if they will pass through the congestion so cooperatively rerouting themselves to avoid it and provide a better traffic balance. Finally, DIVERT [104] differs from FOXS because it offloads the route computation in each vehicle, which improves the privacy of the drives, once that they do not have to report their route and destination to another entity. Furthermore, it reduces overheads to avoid congested areas, as the routes are computed locally in each vehicle. Thus, not depending on the density of vehicles to provide a suitable solution. However, to alert vehicles to compute an alternative route it uses a broadcast mechanism. Nonetheless, unlike FOXS, a suppression mechanism to reduce redundant transmissions is not implemented.

Lourenço et al. [90] proposed GRIFO, a distributed solution to traffic management that uses the vehicle's computational power to detect congestion and calculate new routes when necessary. To this end, each vehicle, through an auxiliary communication infrastructure (RSU), receives information about the traffic characteristics of the nearby roads. The RSUs are distributed in the scenario using the Voronoi diagram and each RSU is responsible for updating and sending the road information (e.g., the time it took the vehicle to travel on each road) within the defined region. Periodically, vehicles request road characteristics to the nearest RSU, and based on this information received, the vehicle application will perform the routing process. Although road processing is distributed among vehicles, the RSU still has a responsibility to collect and aggregate road characteristics to send this information to vehicles. Also, GRIFO can generate new congestion as each vehicle calculates its route using the same algorithm and road information shared with all vehicles on the network. In [91] Lourenço et al. proposed a solution named DESTINy (DEcentralized System for Traffic Management) that is an update of the GRIFO solution. Its major new features are: i) road assignment to RSU, now roads are assigned to closer RSU; ii) the V2V communication has its formula to calculate the retransmission delay based on destination distance modified; iii) and the vehicles forwarded massages (e.g., road condition) also update its local database to reduce the quantity of messages in the system.

3.3.4 Infrastructure distributed solutions

The works described in [42, 44] proposed services for real-time traffic management with route planning and congestion detection. CHIMERA [44] was based on SCORPION [42] and its main difference is in the route suggestion. In [44], an intelligent traffic system was proposed which improves the overall spatial utilization of the road network to reduce the average vehicle travel costs, named CHIMERA. In CHIMERA, vehicles provide their information (ID, current position, route, and destination) to an RSU entity through a single-hop long-range communication, such as 4G and LTE. For this, CHIMERA was modeled into three main parts: *(i)* congestion detection; *(ii)* traffic classification; and *(iii)* route suggestion. CHIMERA performs congestion detection and traffic classification using *K-NN* (k-nearest neighbors) according to the average speed and density of the path. As output, it informs the road classification based on the traffic condition (e.g., free-flow, slightly congested, moderately congested, and severely congested). Finally, CHIMERA uses the K-Shortest Path-based algorithm for the route choice. However, different from this article, these solutions [42, 44] did not propose a message scheduling mechanism to reduce problems in data transmission, such as packet collision. Another problem, solved in this article, is that communication between RSUs is not implemented. Thus, the RSU is not aware of the traffic conditions in other regions of the map, thereby limiting the efficiency of the routing system.

Wang et al. [138] proposed a solution called Next Road Rerouting (NRR) to alleviate urban traffic jams. To calculate a route, NRR applies a heuristic based on a cost function that uses information such as road occupancy, travel time, distance to destination and the congested roads. Vehicle routing is made in two steps. In the first step, the intelligent Traffic Light (iTL) module checks whether any of its intersection roads is congested. If congested, iTL sends beacons informing the vehicles about the congested road. Thus, vehicles that pass through this road request an alternative road to iTL. In the second step, when the vehicle receives the alternative road, this vehicle requests to the central server a new route from the road suggested by iTL to the final destination. NRR has a 3-tier architecture with (i) Central Manager located at the Traffic Operation Center, (ii) intelligent Traffic Light (iTL) with loop detectors disposed in each intersection, and (iii) local computers residing in the middle tier connected to iTL and Central Manager. However, this solution depends only on iTL to acquire road traffic information, thus iTL needs to be installed at all intersections. Moreover, NRR needs an Internet connection to work correctly, while this assumption is not necessary for FOXS.

Younes et *al.* [148, 146, 147] proposed ECODE (Efficient road Congestion Detection protocol) and the ICOD (intelligent path recommendation protocol), that uses V2V and V2I communications to detect traffic congestion on each road segment. ICOD [147] is based on ECODE [148, 146] having as the main increment the mechanism that enables users to choose which type of route the system will provide according to user concerns and priorities (e.g., fuel consumption, traveling time, road segment context). To do that, RSUs are placed in every intersection and, using V2V, vehicles send an advertisement message (ADV) containing its information (e.g., ID, Speed, location, direction, destination, and timestamp) to nearby vehicles. When a vehicle receives an ADV, the information received is aggregated to the neighbor report table (NR) to calculate a traffic monitoring report (TMR) that informs the average road speed, the density and the estimated travel time. Furthermore, the closest vehicle to any RSU sends the TMR to that RSU. When the RSU receives the TMR, they check its local information to determine the best direction for each destination, then disseminate a *RecomReport* message. Finally, when a vehicle receives a *RecomReport* message, it changes its route onto the destination and forwards the message to the one-hop neighbors. Differently, from FOXS, ECODE does not have full knowledge of the map causing the same problem as [93] and despite the forward control message mechanism, the large number of messages necessary to work cause scalability constraints as in [47].

3.4 Final Remarks

This Chapter presented a review of the contributions and limitations of related work to this thesis. Furthermore, the related works indicated that new solutions should be proposed to fill the gap in the design of traffic management services considering specific features of this kind of service.

In Section 3.1 the ITS architectures and communication were presented. In Section 3.2 the data dissemination solutions were shown. Finally, in Section 3.3 the route management services and the comparison with solution design in this thesis were presented.

Summarizing the comparison of route management services, Table 3.1 shows the features of the related works and highlights the main contributions of service presented in this thesis. It is observed that no related work uses the Fog computing paradigm to improve system performance. Because of these features, we developed a routing service, named FOXS, that overcomes the gaps of existing approaches. Therefore, FOXS uses a Fog computing paradigm allowing the cooperation of RSUs, distributing the network and processing capabilities.

Table 3.1: Comparison of related solutions.

Work	Communication Architecture	Route Processing Architecture	Track Shift Avoid	Route Size Check	Route Area	RSU Interaction	Message Management
PAN (i) [105]	V2I	Centralized	✗	✗	All map	✗	✗
PAN (ii) [105]	V2I	Centralized	Random	✗	All map	✗	✗
CHIMERA [44]	V2I	Distributed RSU	Probabilistic	✗	RSU coverage	✔	✗
EcoTrec [47]	V2V/V2I	Distributed Vehicles Server: traffic knowledge	Random	✗	All map	✗	✗
SGTD [57]	V2V	Distributed Vehicles	slightly different road weight	✗	All map	✗	✔
INCIDEnT [93]	V2V	Distributed Vehicles	✗	✗	Neighborhood	✗	✗
GRIFO [90]	V2V/V2I	Distributed Vehicles Server: traffic knowledge	✗	✗	RSU Voronoi region	✗	✗
DESTINy [91]	V2V/V2I	Distributed Vehicles	✗	✗	RSU nearest roads	✗	✗
DIVERT [104]	V2V/V2I	Distributed Vehicles Server: traffic knowledge	✗	✗	All map	✗	✗
ECODE [146]	V2V/V2I	Distributed RSU	✗	✗	Neighborhood	✔	✔
NRR [138]	V2I	step1: distributed iTL step2: centralized	✗	✗	All map	✗	✗

Table 3.1: Comparison of related solutions (continuation).

Work	**Communication Architecture**	**Route Processing Architecture**	**Track Shift Avoid**	**Route Size Check**	**Route Area**	**RSU Interaction**	**Message Management**
Re-RouTE [61]	V2V/V2I	Centralized	Traffic balancing	✗	All map	✗	✔
CDRAM [140]	V2I	Centralized	✗	✗	All map	✗	✔
Garip [54]	V2V	Distributed Vehicles	✗	✗	Near Roads	✗	✗
DisTraC [41]	V2V	Distributed Vehicles	✗	✗	All map	✗	✔
dEASY [3]	V2V	Distributed Vehicles	Traffic balancing	✗	All map	✗	✗
FOXS	V2V/V2I	Distributed Fog	Probabilistic	✔	Defined region	✔	✔

Chapter 4

Proposed Service I
FOXS – Fast Offset Xpath Service

Having discussed the main characteristic of VANETS, and also the related works, this chapter presents a traffic management service for route suggestion called FOXS - Fast Offset Xpath Service. For the development of FOXS, a set of algorithms and mechanisms are designed and integrated.

To reducing the collision problem in 802.11p protocol, we design a mechanism that analyzes and schedules the packet sent by FOXS (see Section 4.2). In section 4.2, an algorithm to cover the whole map using fewer Cloudlets is presented. A road traffic classification and the method to update its classification on each Cloudlet was presented in (see Section 4.3). Finally a probabilistic route suggestion that takes into account the traffic jam shift was presented in (see Section 4.4).

4.1 Overview

FOXS is based on the Fog computing paradigm, which distributes the computer and communication resources among ITS components using the various computational entities, as presented in Figure 4.1.

The FOXS consists of two main components, Vehicles and Cloudlets. Vehicles have communication capabilities and embedded sensors (e.g. GPS) that are responsible for collecting data about road conditions as well as receiving/requesting new routes. Cloudlets are implemented as RSU, and the Cloudlet set forms the Fog computing environment (see Figure 4.1, Label A). Cloudlets are spread in the scenario according to the RSU communication range to reach full coverage of the entire map. Without loss of generality, the RSU deposition follows the cellular antennas deployment (hexagonal areas). Each Cloudlet is responsible for collecting, storing and analyzing the data (vehicle position and velocity, road occupancy, level of congestion) of a specific region and computing routes for vehicles in its region (see Figure 4.1, Label B). By considering specific areas, the data is kept closer to end-users (vehicles) and road sensors, resulting in a more efficient processing, communication and quick response time. As seen in Figure 1.3, the Cloudlet is in Tier B and Vehicles are in Tier C.

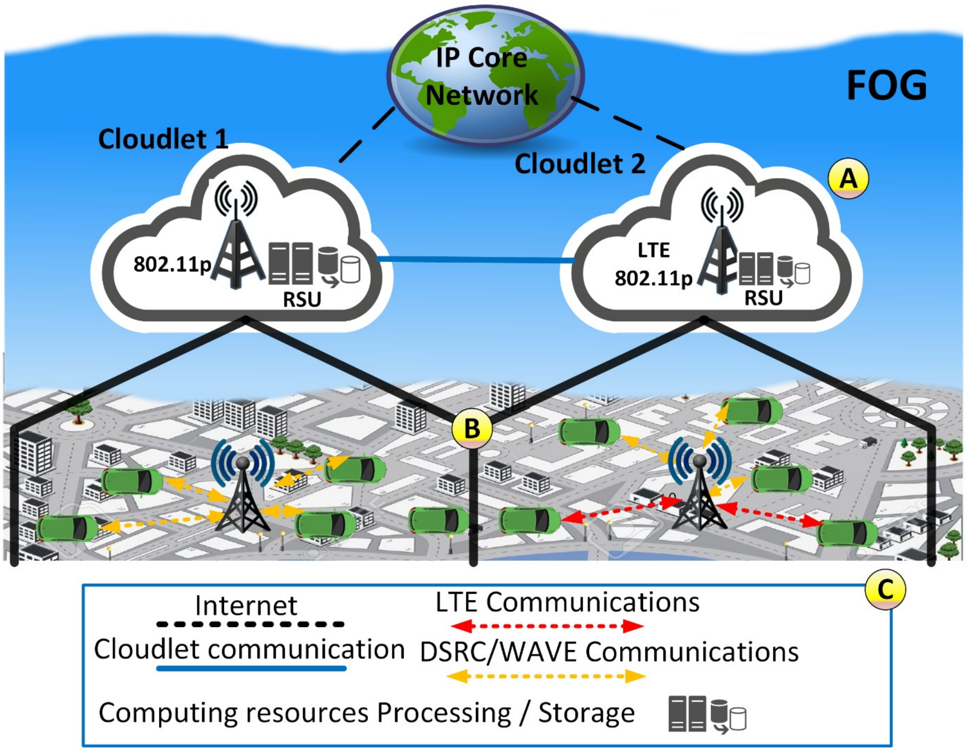


Figure 4.1: FOXS design.

It is important to point out that the spatial context-aware characteristics allows each Cloudlet to be independent from each other. To increase the flexibility and environment communication penetration, the Cloudlet can be equipped with various communication technologies (e.g., LTE, 802.11p) and multiple antennas (Figure 4.1, Label C).

The design of FOXS has three steps following the ITS workflow described in Section 2.3.1: (i) Data Gathering, Subsection 4.2 describes the data gathering process, the algorithms and mechanisms developed for this role; (ii) Data Processing, Subsection 4.3 presents the data transformations and road traffic classification; and (iii) Service Delivery, Subsection 4.4 describes the algorithms and methods used to compute the new route and delivery to the users.

4.2 Data Gathering and Communication

In order to perform data gathering, we need to deploy the RSU infrastructure, i.e., the Cloudlets. After the deployment process, the Cloudlet gathers road/vehicle data.For this implementation, we use a set of RSU with Cloudlet capability spread homogeneously in the environment to achieve full coverage of the entire map (see Figure 4.2). The services assume that does not any location obstructions (buildings, lakes, etc.) for deployment. Each Cloudlet comprises a single RSU with communication capability to other RSUs and with vehicles. The communication between RSU is made by wire and RSU-Vehicle communication uses 802.11p wireless communication. Vehicles are equipped with a GPS and an On-Board Unit (OBU) allowing the vehicle communication and user interaction.

Cloudlets are responsible for collecting all data generated inside its communication coverage (represented by hexagons in the Figure 4.2).

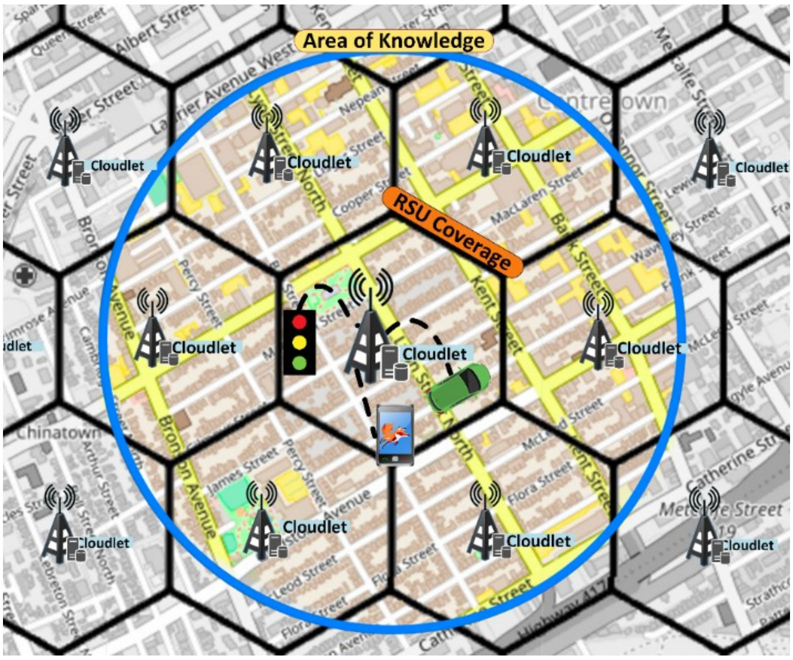


Figure 4.2: Cloudlet Division.

Cloudlets are distributed accordingly to their coverage by applying the Hexagonal Binning ([28]) algorithm to reach an efficient relationship between the number of RSUs versus the city map size. This algorithm is based on cellular base-station deployment models [62]. This strategy is consistent with the Fog paradigm once the map is partitioned, sharing the users between each Cloudlet, and the resources are brought closer to the users. Algorithm 1 shows the process of Cloudlet distribution. The inputs to the Algorithm are the dimensions of the map (width (*Width*), length (*Length*)) and the Cloudlet coverage (*Coverage*). Algorithm 1 based in hexagonal binning algorithm fills the Cloudlet set needed to cover the entire map based on the dimensions of the map and the Cloudlet coverage (Line 5). Next, their coordinates (x, y) are assigned accordingly to the map dimensions (Lines 6–7). If an RSU is assigned to a region that does not have any road, it will be removed.

For the service to work correctly, the components of the traffic management system, such as Cloudlets and vehicles, send control data periodically (beacons) informing about road traffic conditions in their region and other types of data that FOXS uses. The acquisition of this data is executed in a distributed way using the communication capability of the Cloudlets. Cloudlets send beacons informing their position, the route interval for that region and the list of roads congested inside its region. Vehicles use such information to find nearby Cloudlets to send traffic information and request a new route. The data sent are the vehicle speed, position, and time spent to move on each road. These data are sent periodically through beacons to the closest Cloudlet. Once the Cloudlet receives data about a specific region, it uses aiming to knowledge to execute the traffic service.

```
Algorithm 1: Cloudlet Distribution
Input  : Width // Width of the map
         Length // Length of the map
         Coverage // Cloudlet coverage
         Roads // Road list
         cloudlets // set of cloudlets
Output: Coordinates of each Cloudlet
// returns the set of Cloudlet to cover the entire map
cloudlets ← getNumberOfRSUs(Width, Length, Coverage);
foreach $c \in cloudlets$ do
    // returns the Cloudlet coordinates $(x, y)$ if its contains 1 or more roads in our area
    Cloudlets_coordinates.add(getCoordinates($c$), Roads);
end
```

The process of Cloudlet collecting data transmitted by vehicles begins when the beacon time step is reached (stage A in the Figure 4.3). The vehicle creates the beacon message (MSG) and sends it to the vehicular network (B). Whether Cloudlet receives the MSG or not, it verifies if there is not a duplicate MSG and if the origin is in its coverage (C). If so, the Cloudlet updates information, otherwise, it will drop the MSG. Whether other vehicles receive the MSG or not, vehicles will verify if there is not a duplicate MSG (D). If there is, it will drop the message, otherwise the vehicles will check if the origin of MSG is on the same road (E). If true, their beacons MSG are rescheduled for the next beacon time and the MSG data received are aggregated with data from the current vehicle (e.g, average between both speeds) and the MSG is scheduled for forwarding. After the data gathering, there is a data processing phase in each Cloudlet.

The data exchanged considering the proposed cloud-based architecture and the DSRC/WAVE communication protocol is illustrated in Table 4.1. Our proposal takes advantage of the DSRC/WAVE control and service channels to better use the wireless resources.

Table 4.1: Data exchange by DSRC/WAVE communication channel.

DSRC/WAVE Channel	**Vehicle Messages**	**Cloudlet Messages**
Control Channel(CCH) Beacon	–Speed –Position –Time spent moving on each road –Current route	–Position –Region route interval –List of congested roads
Service Channel(SCH) Data	Request route	Response with the new route

In order to improve the communication between vehicles and the RSU, a mechanism for message scheduling was developed to increase the packet delivery rate in the architecture using a DSRC/WAVE communication standard. The mechanism (presented in Algorithm 2) aims to schedule the time of sending the packets to the 802.11p MAC layer to avoid

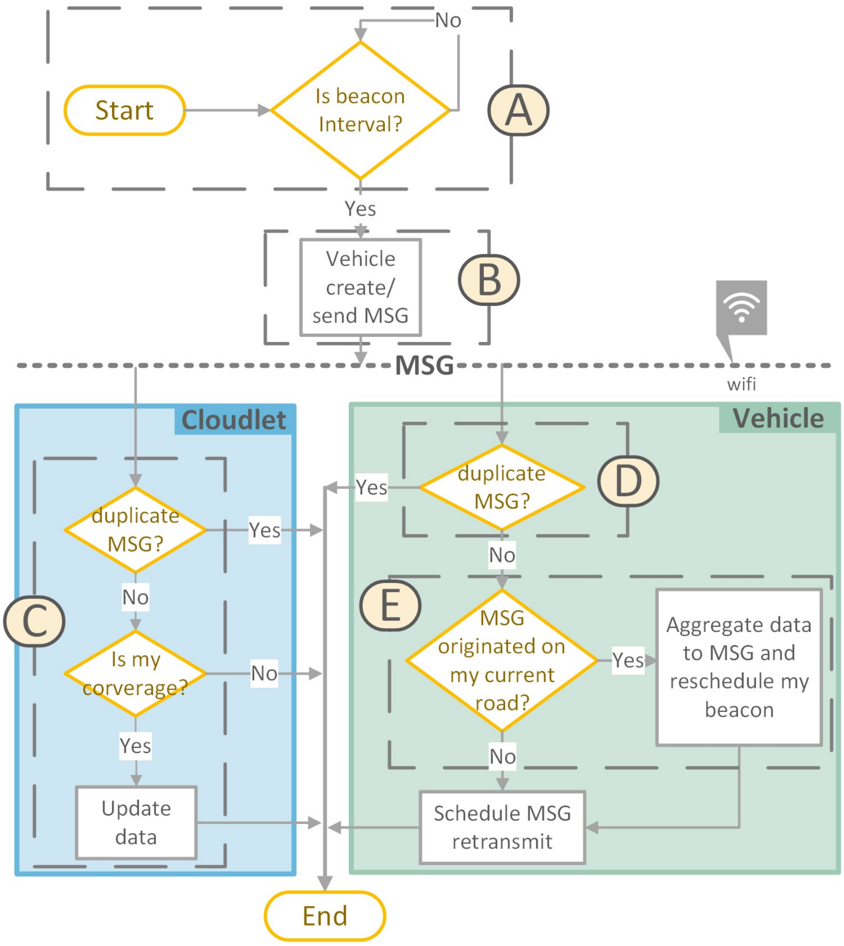


Figure 4.3: Information message flowchart

the resynchronization problem. The mechanism also schedules the messages to be sent according to their size and the node bandwidth (line 16, Algorithm 2). To do this, there are two queues for the messages (lines 4 and 6), one to the control channel (CCH) used to send control messages as beacons, and another to the service channels (SCH) used for all other messages in the service. Algorithm 2 verifies the channel type of the message (lines 8 to 11) then assigns the send delay time according to the last message transmitting time in the corresponding queue (line 16). If the queue is empty, a value of zero is assigned to the delay (line 13). Then an additional delay is calculated based on the active channel, the schedule time and size of the last message in the queue, and the network bandwidth (lines 18 to 28). Thus, as all messages are scheduled for transmission, this mechanism ensures that when the network layer sends a message to the MAC layer, the message will be sent promptly.

Algorithm 2: Message Schedule Mechanism

```
Input  : //
    SCH // service channels;
    Tc // channel active time (50 ms);
    Ts // remaining time for a channel switch;
    P // Message with data, delay, and channel type;
    Qs // queue of SCH message;
    Qc // queue of CCH message;
    Q // queue to sent message;
Output: updated sent message queue
if $P.channel == SCH$ then
    $Q \leftarrow \&Qs$;
else
    $Q \leftarrow \&Qc$;
end
if $Q$ is empty then
    $P.delay \leftarrow 0$;
else
    $P.delay \leftarrow Q[lastItem].delay + (Q[lastItem].size/bandwidth)$;
end
if $P.channel! = CurrentChannel$ then
    $rounds \leftarrow \lfloor P.delay/Tc \rfloor$;
    $Ta \leftarrow Ts + (rounds * Tc)$;
    $Td \leftarrow P.delay + Ta$;
else
    $Td \leftarrow P.delay$;
    $Ttmp \leftarrow P.delay - Ts$;
    if $Ttmp > 0$ then
        $rounds \leftarrow \lceil Ttmp/Tc \rceil$;
        $Ta \leftarrow rounds * Tc$;
        $Td \leftarrow P.delay + Ta$;
    end
end
$P.delay \leftarrow Td$;
$Q$ insert $P$;
```

4.3 Data Processing

In this phase, the system processes the collected information from the previous step. Once the map is all covered by Cloudlets, each Cloudlet has the responsibility of collecting and processing only road data within its coverage. Thereby, limiting the data gathering/processing to a smaller region, reducing the cost of communication and making a processing load balance between Cloudlets in the system.

However, as shown in Chapter 4.6 (Section 4.6.2), the size of the route region (amount of information) that will be used to compute the new route has an impact on the quality of the route. Therefore, a router region that is just the radio coverage of the Cloudlet may be too small for efficient routing. To solve this problem, the Cloudlet acquires information about roads that are in the coverage of other Cloudlets in order to improve the routing solution by increasing the amount of information (with roads and their features and current traffic situation) that the routing algorithm will use. This additional knowledge area contains information of roads that are under the responsibility of other Cloudlets and is called Area of Knowledge (AoK) (inside the blue circle in Figure 4.2). The Area of Knowledge is at least the size of the Cloudlet coverage area. The size of the AoK affects the performance of the service, since a larger AoK (e.g., more roads to route) results in a better result, but the computation time is increased. The best AoK is choosed by a set of experimentation presented in Section 4.6.2.

Each Cloudlet periodically updates the weight of each route based on information gathered by vehicles inside its coverage. A multi-weight directed graph $G = (V, E)$ is used to represent the AoK, where $V = \{v_1, v_2, ..., v_i\}$ is the set of intersections within a range of the AoK (representing the vertices) and $E(w) = \{e_{01}, e_{12}, ..., e_{ij}\}$ is the set of roads connecting the intersections $E \subseteq VxV$ (representing the edges). Each edge e_{ij} is defined by a pair of subsequent vertices $(v_i, v_j) \in V$. Give $C = \{c_1, c_2, ..., c_i\}$ as a set of vehicles (nodes) and $\langle e_{ij}, ..., e_{mn} \rangle \mid e_{ij}, ..., e_{mn} \in E$ is the route $\forall c \in C$.

The weight of each road is a tuple $W(k, l) = \{w_{01}, w_{12}, ..., w_{ij}\} \mid \forall w_{ij}$ is the weight of road e_{ij} $\forall$ $e \in E$. For weight $W(k, l)$, k is the relation of the maximum allowed speed in the road inversely proportional to the average speed at which vehicles travel on the road, represented by S_{ij} in equation 4.1. Therefore, if the vehicle speed is close to the maximum speed allowed on the road, the weight of the road is lower; l is the road occupancy that is inferred through the vehicle's positions sent by vehicle beacons.

$$S_{ij} = 1 - \frac{S_{E_{ij}}}{S_{E_{ij}}^{Max}}. \tag{4.1}$$

Where:

$S_{E_{ij}}$ = average road speed in a defined time period.

$S_{E_{ij}}^{Max}$ = max allowed speed in the road.

However, these weights are not applied to roads in real-time with new data arrival. The road weight is done periodically in the following way. The Cloudlet makes averages of the data sent (speed, occupancy) by vehicles that pass in a specific road for a time interval. Next, the Exponential Moving Average (EMA) is used to obtain the new road weight. The EMA is used to smooth out a large oscillation in the road weight that may occur toward recurrent events such as a vehicle parking or stopped for a short period of time. Thus, implying in an abrupt increase of the road weight implying in the road choice in the routing process. Equation 4.2 presents a generalization formula of EMA.

$$EMA_n = \begin{cases} Y_1, & n = 1 \\ (1-\alpha)^n \cdot EMA_0 + \alpha \sum_{i=1}^{n} (i-\alpha)^{n-i} \cdot Y_i, & n > 1. \end{cases} \tag{4.2}$$

Where:

Y_n = the time interval average (e.g., S_i, $occupancy_i$).

EMA_n = the value of the EMA at any time period (e.g.,$w(k)_{ij}$,$w(l)_{ij}$).

α = the degree of smoothing/weighting decrease factor between 0 and 1.

Note that a higher α reduces the value of old observations more quickly. α is defined in equation 4.3.

$$\alpha = \frac{2}{1+P}. \tag{4.3}$$

where, P is the smoothing period. For our solution we use a $P = 15$ for a smooth transition chosen by a set of experimentation.

The road weight K is used by the routing algorithm (described in the Section 4.4) and the weight L is used with the K to road classification. The road classification is used to inform vehicles which roads have congestion, so the request for a new route is made based on this information. The road classification is based on Level-Of-Services (LOS) present in the Highway Capacity Manual (HCM) [18]. The HCM uses the speed and density of vehicles on the roads to measure the capacity and quality of traffic. The HCM classifies the congestion into six levels as follows:

- LOS A: free flow;
- LOS B: reasonable free flow;
- LOS C: restricted freedom of maneuvering, changing lanes carefully, secondary incidents are easily absorbed, queues can be expected behind an obstruction;
- LOS D: velocity begins to decline as flow increases; rapidly growing density; very limited freedom to maneuver; Secondary incidents create queues;
- LOS E: operation at maximum capacity, vehicles next to each other, incidents produce queues and congestion;
- LOS F: unstable and interrupted flow; congestion, demand exceeds the capacity.

Each LOS level defines the minimum and maximum speeds for each level based on the maximum speed allowed on the road, and by maximum occupancy capacity. We consider a road congested when the relative road speed (calculated as Eq. 4.1) is classified as LOS C, or the road occupation is classified as LOS D. These threshold levels were chosen because that is when the road presents signs of emerging congestion. Thus, FOXS takes action to prevent a vehicle taking a road/street with sing of congestion so reducing the congestion formation/increase. Table 4.2 and Table 4.3 present the LOS of relative road speed and occupancy capacity respectively.

Table 4.2: LOS criteria of relative road speed (based on [18]).

LOS	Relative Speed
A	$> 85[$
B	$]67$ - $85]$
C	$]50$ - $67]$
D	$]40$ - $50]$
E	$]30$ - $40]$
F	$\leq 30]$

Table 4.3: LOS criteria of occupancy (based on [18]).

LOS	Occupancy
A	$< 26[$
B	$[26$ - $42[$
C	$[42$ - $60[$
D	$[60$ - $80[$
E	$[80$ - $100[$
F	> 100

Note that each Cloudlet only classifies and updates the weight of the roads in its coverage area. To update roads on its AoK outside its coverage, Cloudlets share the road knowledge between them using publish/subscribe paradigm [50]. Subscribers register in events and asynchronously they are notified of events generated by publishers. The publish/subscribe protocol developed for FOXS is based on Message Queuing Telemetry Transport (MQTT) [66]. Regarding this, all Cloudlets in the system are publishers, subscribers, and the Cloud server is the broker (see Figure 4.4). In the beginning, each Cloudlet subscribes in the road update process by sending a road list including all roads inside its AoK except for roads inside its radio coverage (represented by hexagons). This list is sent to the Cloud server that has global knowledge of the map division (represented in Figure 4.4, Label A). Afterwards, the Cloud server notifies each Cloudlet (publisher) that is responsible for each road from the received list (see Figure 4.4, Label B). Cloudlets (publisher) notify all Cloudlets that are subscribers about road updates (see again Figure 4.4, Label C). Note that several Cloudlets may be intersected on the same path. The publish-subscribe method is an interesting choice for ITS because of its asynchronous nature. The method also has the ability to work with context-aware applications as interests flow considering a specific geographic region [80].

After all these update processes, the Cloudlets disseminate the list of congested roads inside its AoK in beacon messages. Thus, FOXS takes action in order to avoid the formation of congestion.

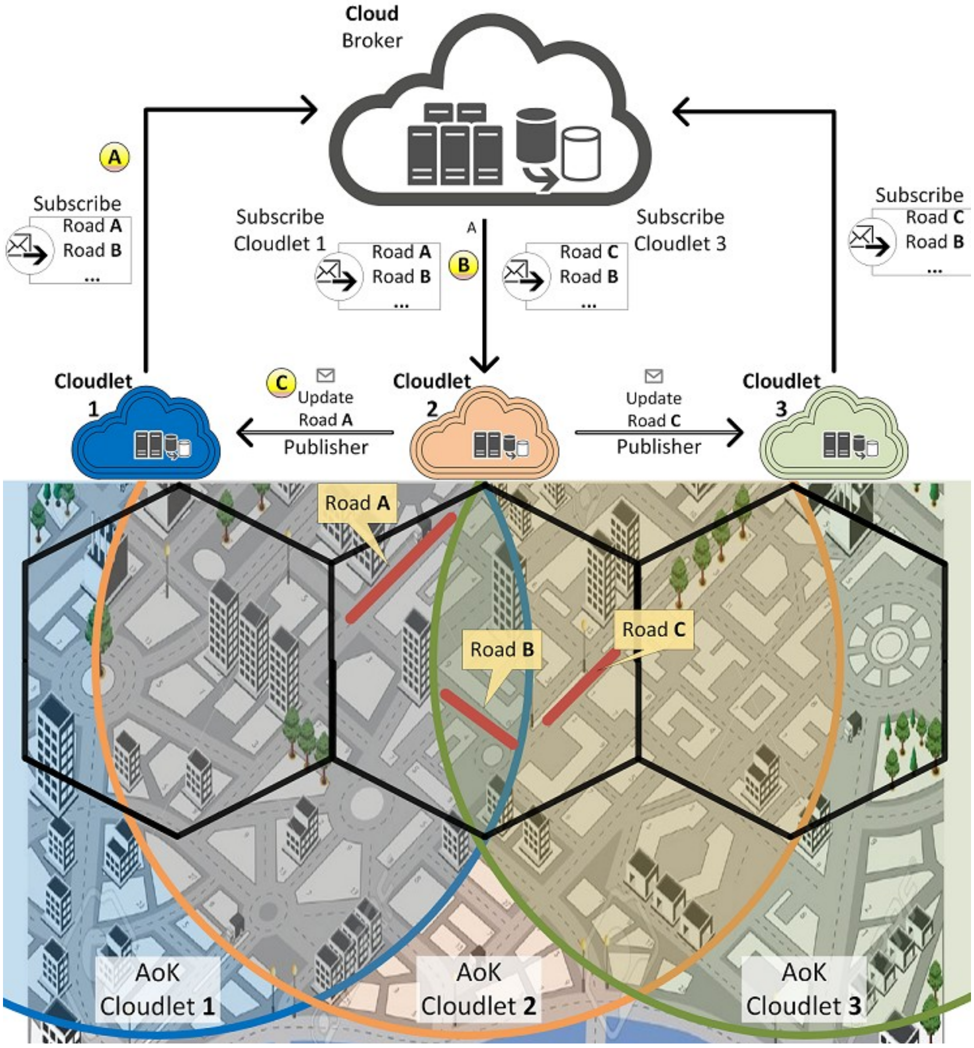


Figure 4.4: FOXS publish-subscribe protocol.

4.4 Service Delivery

In this phase, each Cloudlet performs the detection and control of congestion by calculating alternative routes to the vehicles. Thus, decreasing the load on the congested roads. Each vehicle periodically checks if it will pass through a congestion road. To do that, the vehicles receive a beacon message sent by the Cloudlet with a list of congested roads. This list only contains roads belonging to the Cloudlet's AoK.

Hence, at each route interval, the vehicle checks if its route passes through a congested road. The service delivery mechanism is illustrated in Figure 4.5. If it does not pass in congested road (Figure 4.5 stage A), the router interval is restarted. Otherwise, a message is sent to the closest Cloudlet requesting a new route and recovery time is started (Figure 4.5 stage B). The recovery time is a fault-tolerance mechanism that checks if the vehicle has received the requested route within a specific time interval. A new request message is sent in case of failure to receive (Figure 4.5 stage C). When the vehicle receives the new route, it is assigned and the route interval time is started (Figure 4.5 stage D).

Therefore, the Cloudlet computes a new route for the requesting vehicle in the scope

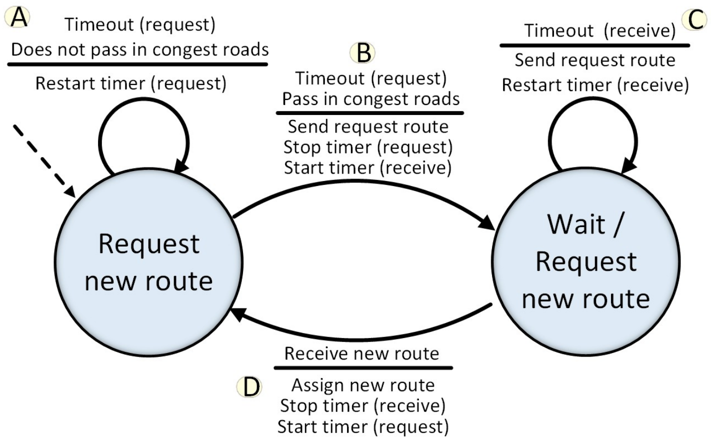


Figure 4.5: State diagram: Request new route.

of its AoK, i.e., only using roads within the blue circle. Thus, the rerouting process does not change the part of the route outside of the AoK in purple (standard route). As can be observed in Figure 4.6, the routing of vehicles (e.g., green car) is performed considering its current position (point **A**) until the last road in its current route that is within the AoK of the Cloudlet (point **B**).

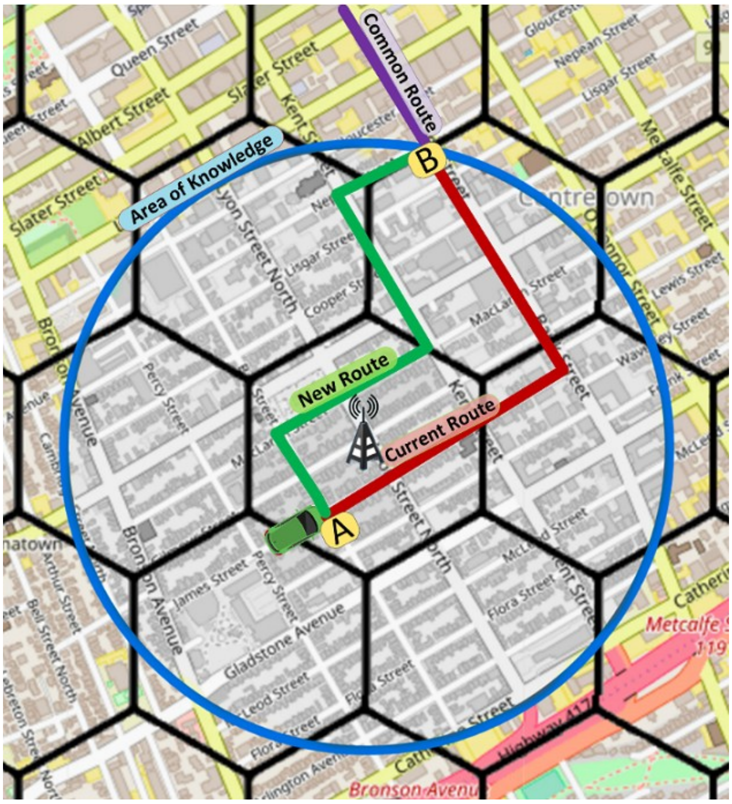


Figure 4.6: FOXS: Fast Offset Xpath Service.

As can be seen in Algorithm 3, the routing process begins when the RSU receives a new route request from vehicles with their information (e.g., current position, route) (Line

Algorithm 3: Route Computing

```
Input : //
        $v$ // request vehicle information
        $G$ // Graph created by each Cloudlet (AoK)
        $K$ // number of alternate routes
$route \leftarrow$ v.getRoute();
// returns the current edge of vehicle v contained in graph G
$source \leftarrow$ v.getPosition();
// returns the last edge inside of the AoK of the Cloudlet responsible
$lastEdge \leftarrow$ G.getLastEdge($route$);
// returns remaining edges of the route
$remainingEdges \leftarrow$ getRemainingEdges($lastEdge, route$);
// returns k shortest paths between current and last edge for vehicle v
$alternativeRoutes \leftarrow$ G.getKShortestPaths($source, lastEdge, K$);
// selects a path from the set of alternate routes with Boltzmann
$newRoute \leftarrow$ boltzmann($alternativeRoutes, G$);
if $\underline{lastEdge! = \text{route.getDestination()}}$ then
    // concatenates the remaining route of the old route to the new route
       to vehicle v
    newRoute.add($remainingEdges$);
    // returns the new route
    sendRoute($v.getId(), newRoute$);
end
// sends the new route to vehicle v
sendRoute($v.getId(), newRoute$);
```

4, Algorithm 3). To do this, Algorithm 3 uses the graph G of AoK with its congestion characteristics (in Algorithm 3 as G) and the variable K describing the maximum number of alternative routes that must be calculated.

The route weight is calculated as the sum of the weights of all the roads contained in the path. Thus, the routes with lower weights are the most requested, possibly moving the congestion from one point to another. Aiming to avoid this problem, the service computes a set *alternativeRoutes* of K alternative shortest paths as possible routes the vehicle can take (Line 8, Algorithm 3).

A route of this set is selected probabilistically based on the sum of the weights (w) of its roads applying the Boltzmann probability distribution [75] (Line 9, Algorithm 3).

Boltzmann's probability was chosen because it fits well with the vehicle route problem. Boltzmann probability uses the concept of temperature to make a probabilistic choice of a route preventing the algorithm to choose the same route multiple times. When temperature $T \to \infty$, all alternatives routes have the same probability of being chosen, i.e., the process approaches a uniform random distribution. When temperature $T \to 0$, the lightweight route has a high probability of being chosen. Thus, using the set of R_j, the vehicle traffic is balanced between roads and the general performance of FOXS is maintained. The decision rules to choose the new route are presented in the equations as follows:

J = set of vehicles on the scenario
R_j = set of alternative routes of the vehicle j ($j \in J$)
r_j^i = route i of vehicle j ($j \in J$) and ($r_j^i \in R_j$)
w_j^i = weight of route r_j^i
T = Boltzmann temperature; $T \in [0, \infty]$
$N(w_j^i)$ = normalized value of w_j^i ($w_j^i \in [0, 1]$) defined by Eq. 4.4:

$$N(w_j^i) = \frac{W(r_j^i)}{max\{W(r_j^i) \quad | \quad \forall r_j^i \in R_j\}}. \tag{4.4}$$

The K_T^j is the Boltzmann constant of vehicle j for temperature T, according to Eq. 4.5:

$$K_T^j = \sum_{i \in R_j} e^{-\left(N(w_j^i)/T\right)}. \tag{4.5}$$

The $P_T^j(r_j^i)$ is the probability of choosing route i of vehicle j with the parameter of temperature T, according to Eq. 4.6:

$$P_T^j(r_j^i) = \frac{1}{K_T^j} e^{-\left(N(w_j^i)/T\right)}. \tag{4.6}$$

The $E(R_j)$ is the route chosen ($E(R_j \in R_j)$), the choice is made according to Eq. 4.7:

$$E(R_j) = \max\{X \times P_T^j(r_j^i) | \forall r_j^i \in R_j, X \sim \cup([0, 1])\}. \tag{4.7}$$

Once the route is selected, the system checks whether the last edge of the calculated alternative route is the destination of vehicle v (Line 10, Algorithm 3). If this condition is not satisfied, the new alternative route is concatenated to the remaining of the original route that lies outside the AoK of the Cloudlet that made the routing of v (Lines 11–12, Algorithm 3). Afterwards, the system sends a message to the vehicle with the new route.

When the requesting vehicle receives this new route, its navigation system verifies the satisfaction of the variable *route size factor*, which determines how much longer (in percent) the new route may be in comparison with the current route. If this *route size factor* does not satisfy it, the vehicle maintains the current route. Therefore, the system can limit the maximum size of the route, thus avoiding the increase of other traffic problems such as CO2 emission.

Figure 4.7 presents a flowchart of the process presented above. Point A in Figure 4.7) shows when the vehicle creates a request route message (MSG) addressed to the closest Cloudlet and transmits through VANET. Whenever the Cloudlet receives a message, it verifies if it is addressed for it (point B). If it is not, the route message is dropped, otherwise, the Cloudlet calculates a set of K shortest paths (point C1). After this stage, one path is chosen in a probabilistic way using the Boltzmann algorithm. After the route is chosen, the Cloudlet creates a message (MSG) with the new route (point C2) and sends it to the VANET (point C3). When the vehicle receives the MSG (point D), it checks if it is addressed to it. If so, it drops the MSG, otherwise, it verifies whether the route size factor is satisfactory. In this case, the new route is assigned to the vehicle, otherwise, the request routine is finalized.

4.5 FOXS Final Remarks

This Section presented FOXS, a Fog-based ITS service to detect, classify, and control traffic congestion. The use of the Fog computing paradigm introduces several benefits to the FOXS such as computational distribution, providing the processing load balancing and proximity of computational resources to the end-users, decreasing the response time and the network bandwidth usage.

FOXS design was divided into three stages: (i) Data Gathering, where all data are collected; (ii) Data Processing, where the gathered data is transformed in relevant information to the service; and (iii) Service Delivery, where the route suggestion is computed and forwarded to users.

To ensure FOXS works properly, a set-off algorithms and mechanisms are designed. In (i) Data Gathering, a stage related to data collection, has proposed a package scheduling mechanism for the 802.11p protocol. An algorithm was also proposed to distribute the Cloudets in the scenario in order to obtain full coverage of the map. In (ii) Data Processing was a road classification method based on LOS that uses information such as speed and road occupancy. For the smooth change in road classification, the EMA process was implemented to avoid a drastic change in the weight of the road, such as a vehicle maneuvering on the road to park. This stage also presented a method based on MQTT protocol to update road classification on all Cloudlets. Finally, in (iii) Service Delivery, a

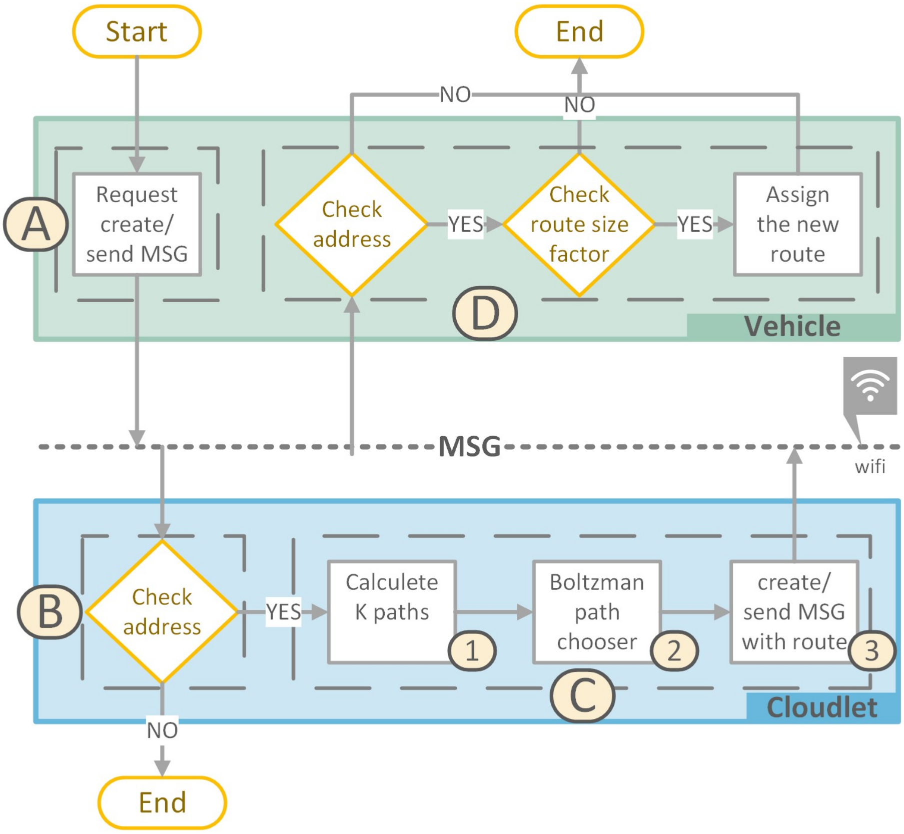


Figure 4.7: FOXS flowchart

probabilistic route suggestion that uses Fog computing capabilities was presented. This route suggestion calculates only routes for vehicles inside its AoK, thus reducing the network delay and making the service more scalable. Several experiments comparing FOXS with various literary services in different scenarios are presented in Section 4.6.

4.6 FOXS Performance Evaluation

This section presents the experiments conducted to validate the described routing service as well as the framework and its components.
Simulations to analyze service behavior according to its parameters' configuration were performed. These evaluations help us to understand the service behavior as well as to choose the set of parameters that generated the best results.

The experiments are divided into three subsections: i) Experiment 1 (Section 4.6.2) presents a study related to the choice of the number of alternative K routes calculated by the FOXS service and the size of AoK, as well as the relationship between these variables; ii) Experiment 2 (Section 4.6.3) presents the impact of the number of vehicles on the network that accepts the route suggested by the service. Moreover, an analysis related to the routing interval and AoK size was performed; iii) Experiment 3 (Section 4.6.4) presents a comparative evaluation between the FOXS and other literature solutions. The configuration parameters for these experiments are chosen accordingly with configuration results obtained in Experiment 1 4.6.2 and Experiment 2 4.6.3. In addition to evaluating traffic conditions and their impacts, a communication network evaluation was performed.

These experiments were performed in different stages of the development of the presented service, thus containing some differences. However, the exploratory analysis of configurations presented in Experiment 1 (Section 4.6.2) and Experiment 2 (Section 4.6.3) are accurate to apply to services presented in this thesis. The detailed description of the service evaluated in Experiment 1 can be found in paper [25] and the service of Experiment 2 in paper [24]. The service presented in Experiment 3 is described in this thesis.

4.6.1 Tools and General Methodology

This subsection presents the tools used during the development and evaluation of this thesis.

The solutions presented in this thesis were validated through simulations using the Veins framework, which allows the integration of OMNet++ to simulate the communication environment and SUMO to simulate vehicle traffic on urban and highways roads. When running simulations, both simulators (OMNet++ and SUMO) are executed in parallel and connected via Transmission Control Protocol (TCP) (Transmission Control Protocol). The evaluation was based on realistic scenarios as the SUMO simulator can import real city maps through an Open Street Map [1]. The tools are described below:

- OMNet++ (Objective Modular Network Testbed) [103] is an event simulation environment. This tool is generally used in telecommunications network traffic and queue modeling, multiprocessing modeling, and other distributed hardware systems. This simulator is open-source and was not designed to be a specific simulator, so it has been used in many different domains, from simulating queuing networks to the wireless network. The project OMNet++ uses the Eclipse Integrated Development Environment (IDE) [2] as a desktop environment, enabling textual and graphical editing of the modules used in the project;

- MiXiM (Mixed Simulator) [98] is a framework that has many algorithms and protocols for wireless networks. This framework provides the developer with a base node, which has three sub-modules of the Open Systems Interconnect (OSI) networking reference model namely: appl, net, and nic. In addition to these submodules, there are also *mobility* submodule to move the nodes, the *Address Resolution Protocol (ARP)* function to discover the MAC address of the computational element to which a given IP package must be delivered and the *utility* to provide communication between these submodules;

- SUMO (Simulation of Urban Mobility) [15] consists of two main components: the map editor and the motion editor. The map editor allows map generation automatically, manually or by importing from real environments. This editor allows the user to trace routes on which vehicles will travel and also creates specific routes for each vehicle. The motion editor allows you to define all vehicle mobility parameters such

---

[1] Open Street Maps is a free collaborative mapping project (`http://www.openstreetmap.org/`).

[2] `http://www.eclipse.org/`

as accelerations and decelerations, minimum and maximum speeds, stop times and the route of each vehicle. In addition to the mobility parameters it is possible to evaluate pollutant emissions, fuel consumption, and noise pollution measurement. To manage the traffic provided by SUMO, TraCI (Traffic Control Interface) [141] was used, which is an interface that allows traffic management of the simulations at runtime. Thus, we were able to access and control the values of simulated objects, such as modifying vehicles' routes;

- Veins (Vehicles in Network Simulation) [125] is the open-source inter-vehicle communication (Inter-Vehicular Communication (IVC)) simulation framework as well as middleware between OMNet ++ and SUMO.

Figure 4.8 presents the Veins architecture connected in parallel to SUMO and OMNet++ via a TCP socket. The communication protocol has been standardized as the Traffic Control Interface (TraCI).

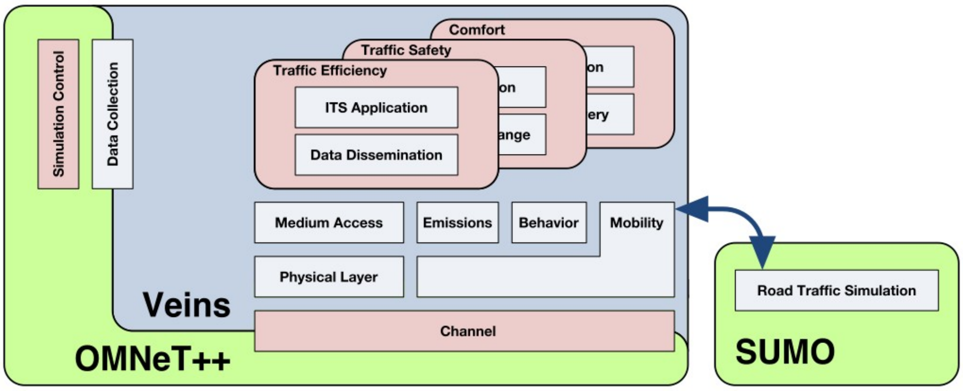


Figure 4.8: Veins architecture (Source [125]).

For the simulations and evaluations of the proposed service, 33 simulation repetitions were carried out and the results present the values with a confidence interval of 95%. To measure the CO_2 emissions and fuel consumption the EMIT model was used integrated into SUMO. EMIT is a simple statistical model for instant emissions of CO_2 and fuel consumption based on the acceleration and speed of vehicles. The EMIT formulation is described in the HBEFA [3]—Handbook Emission Factors for Road Transport. Specific metrics and features are described in their respective evaluation subsection.

4.6.2 Experiment 1

The following experiments evaluate the impact of the AoK size, and the number of k shortest paths in the route suggestion service.

Methodology

For the simulations, we used the network simulator OMNeT++ 4.3, SUMO 0.19 (Simulation of Urban MObility), and Veins 2.1.

---

[3] `http://www.hbefa.net`

For the simulation and evaluation of our system, we used the TAPAS Cologne scenario [131]. The TAPAS Cologne scenario (Figure 4.9) is a realistic scenario that contains information about vehicle mobility in the city of Cologne, Germany. The scenario includes the complete map of the city and the mobility trace of the vehicles, obtained through a real monitoring of the city traffic. The trace defines the times of entry of vehicles on the network during the period between 6:00am and 8:00am. During this period, several points of congestion are formed in the city. The trace has a duration of approximately 6 hours, since the trace ends only when the last entered vehicle completes its route and leaves the simulation. Table 4.4 presents some features of the evaluated scenario.

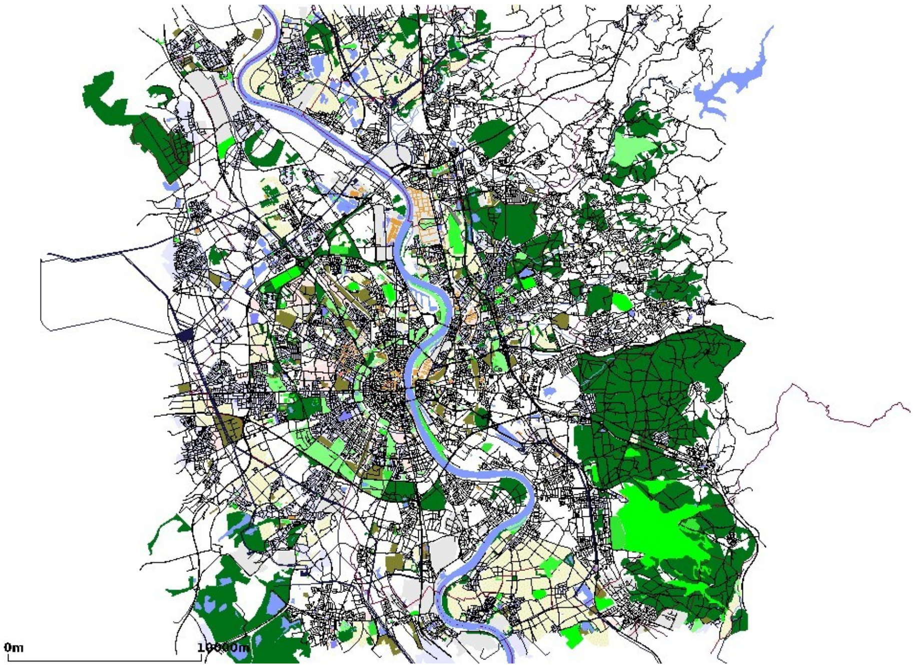


Figure 4.9: Topology of the Tapas Cologne Scenario.

Characteristics	**Values**
Size	$400\,\text{km}^2$
Amount of roads	4500 km
Peak of vehicles	136.000
Total vehicles	252.000
Duration	6 hours

Table 4.4: Features of the TAPAS Cologne scenario

The distribution of the RSUs happens homogeneously based on the communication radius of the RSUs and on the dimensions of the map. In this evaluation, different configurations of the AoK were tested in the RSU, and these areas were $2, 4$ and 6 kilometers to assess the impact of RSUs scenario knowledge in the proposed system. The frequency of re-routing was fixed at 180 seconds.

When the simulation reaches a steady-state, the re-routing of the vehicles starts minimizing the traffic jam that already exists in the actual trace scenario. Thus, different k

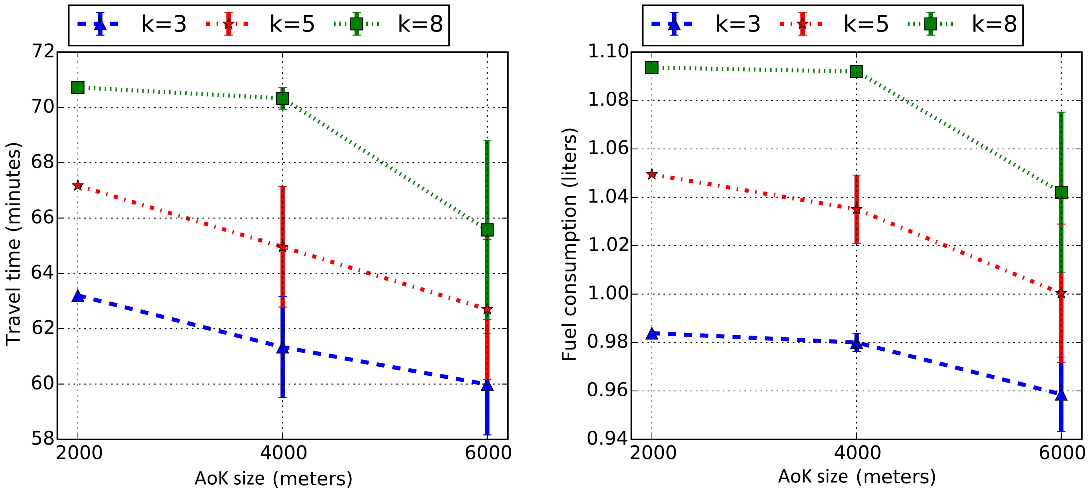
shortest paths were tested, which were $3, 5$ and 8 values. Finally, we use the following metrics to validate the service:

- *Travel time*: the average travel time from the starting point to the destination of all vehicles;
- *Fuel consumption*: average fuel consumption of all vehicles to traverse the entire path;
- *CO2 emission*: average CO2 emission of all vehicles from the entire path.

Evaluation of the proposed system with different configurations

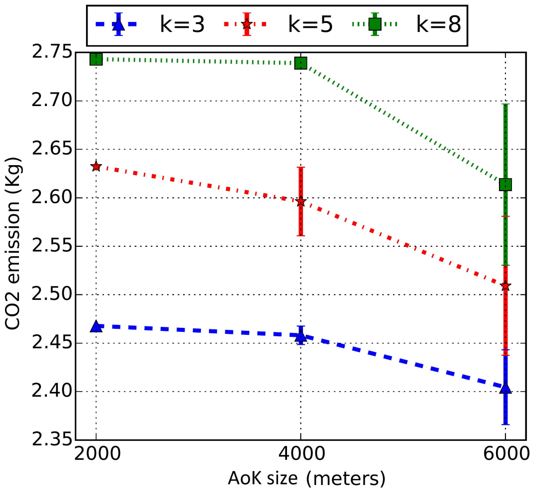
Figure 4.10 presents the average travel time, fuel consumption, and CO2 emission of the service by varying the AoK of the RSUs ($2, 4$ and 6 km), and the k shortest paths ($3, 5$ and 8).

Figure 4.10(a) shows the results of the average travel time. As can be observed, the higher the AoK of the RSUs, the greater the reduction in travel time. This is because RSUs have a greater knowledge of the area of the map and the calculated paths are better in terms of travel time when computed for a smaller area of the map. Moreover, we can observe that the amount of k shortest paths has an impact on the results obtained by the proposed service. For the variations of k, we can note that $k = 3$ has a better performance when compared to other values for k, thus resulting in a decrease of up to 9% concerning $k = 8$ and a reduction of up to 4% in relation to $k = 5$.



(**a**) Travel time.

(**b**) Fuel consumption.



(**c**) CO2 emission.

Figure 4.10: K shortest paths *vs* AoK size evaluation.

Figure 4.10(b) presents the fuel consumption of the service. As we can see, the higher the AoK of the RSUs, the greater the reduction in fuel consumption. This occurs because the RSUs have a greater knowledge of the area of the map and the calculated paths are better when computed for a smaller area of the map. Moreover, we can observe that the amount of the k shortest paths has an impact on the results obtained. The results show a difference of up to 7% on the evaluated configurations.

The results obtained with the proposed service for CO2 emission are presented in Figure 4.10(c) which related to fuel consumption following its behavior (Figures 4.10(b)).

Concluding, the results presented in Figure 4.10, show that the higher the AoK of the RSUs, the greater the reduction in travel time, fuel consumption and CO2 emission. Likewise, the results also show that the smaller the amount of the k shortest paths, the greater the reduction of the effects caused by congestion. Observe that greater values of k can lead to longer (worse) paths, thus directly affecting travel time, fuel consumption and CO2 emission. Another fact is that even if a longer route has more unconstrained traffic than the shortest route, the longer route has a greater number of crossings and traffic lights. Therefore, implying in a greater number of unnecessary vehicle braking and acceleration, which can decrease efficiency and continuous traffic.

4.6.3 Experiment 2

This set of experiments presents the impact in the system when only a subset of vehicles accept the new route suggested by the traffic management service. Furthermore, an analysis related to the routing interval and AoK size was performed. These experiments were made in FOXS early variation named FOX presented in [24].

Methodology

The map chosen corresponds to the Manhattan area in New York City, United States, with an area of $5km^2$. Vehicles travel with randomly chosen routes and the vehicle density in the area ranges from 1000 to 1500 vehicles/km^2 during the simulation.

As described earlier, the distribution of the RSU is homogeneous and it is based on the size of the communication radius and map dimensions, so that the larger the communication radius is, the smaller the amount of RSU used to accomplish full coverage of the map is. In our assessment, different configurations of the radius size of the AoK were tested in the FOX to assess the impact of these changes.

When the simulation reaches a steady state, the FOX starts the re-routing of vehicles in order to minimize a imminent congestion or congestion already in place. Thus, different parameter values are varied to analyze the best combination among them.

The *re-routing intervals* were tested, which were 150, 300 and 600 seconds. For the re-routing of vehicles, our algorithm has a parameter $K-routes$ that determines the number of alternative paths provided, hence we evaluated values 3 and 5. For the *route size factor*, which determines how much bigger (in percent) the new route can be when compared to the old one, we evaluated the values 25%, 50%, 75% and 100% (always accepted). The amount of vehicles was also varied that accepts the alternative route (ACC) by considering 25%, 50%, 75% and 100% for simulate users that do not accept the route for any other reason. In the radius of the AoK for RSU, we evaluated $1000m$, $2000m$, $4000m$. Table 4.5 shows the simulation parameters and values used in our evaluation.

Table 4.5: Simulation parameters Experiment 2.

Parameters	**Values**
Map Size	5 km^2
Re-routing interval	150, 300, 600 seg
Alternative routes (k)	3, 5
AoK (interest area)	1000m, 2000m, 3000m
Route size factor	25%, 50%, 75%, 100%
Percentage of Vehicles that accept the route (ACC)	25%, 50%, 75%, 100%

Finally, for the validation of our system, the following metrics were evaluated:

- *Traveled Time:* the average travel time from the starting point to the destination of all vehicles;
- *Stopped time:* average time spent stuck in traffic jams for all vehicles;
- *Average speed:* average speed of all vehicles;
- *Traveled distance:* average distance that all vehicles traveled;
- *Fuel consumption:* average fuel consumption of all vehicles to traverse the whole route;
- *CO2 emission:* average CO2 emissions for all vehicles during their journey.

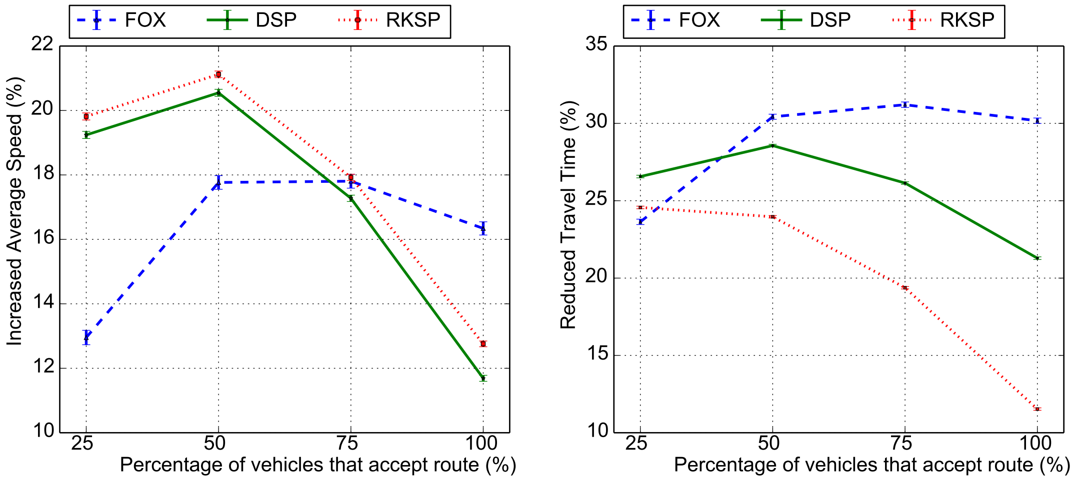
Evaluation of percentage of vehicles that accept route suggestion

The routing solutions FOX, PAN(i) [105], aka DSP, and PAN(ii) [105], aka RkSP need to pre-configure the routing interval. Tests with different re-routing intervals were performed and the best routing interval obtained was 150 seconds. Parameter k was also studied and $k = 3$ produced the best results for the algorithms FOX and RKSP. The value $k = 5$ presented the worst results since it led to longer (worse) paths, thus directly affecting the solutions. Results only show the density of 1500 vehicles/Km^2, because such density causes severe congestion, demanding more work from all the solutions.

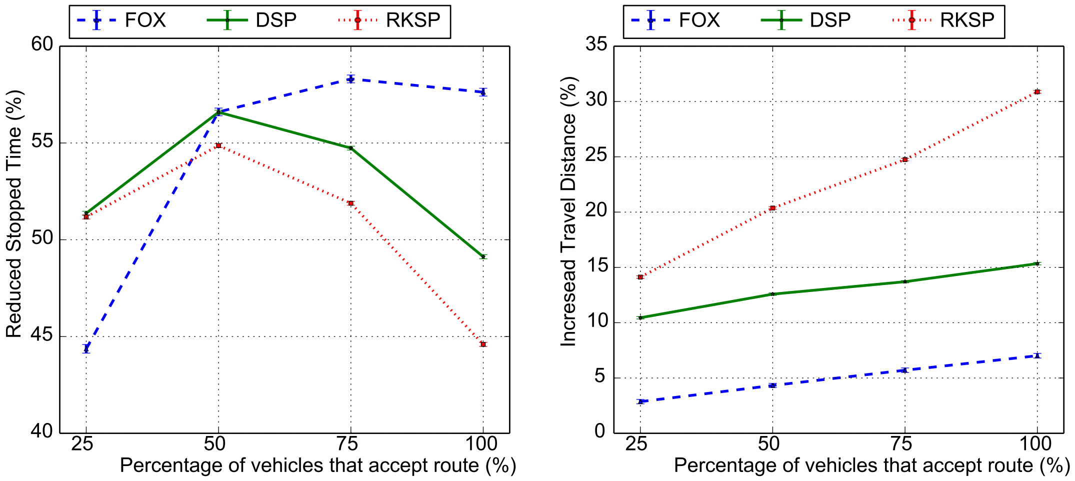
Figure 4.11 shows the percentage increase or decrease of all metrics for all protocols in relation to the baseline solution (no re-routing) when varying the *percentage of vehicles that accepts to be re-routed* (25%, 50%, 75%, 100%). As can be seen in Figure 4.11(a), when 25% of the vehicles are re-routed, DSP and RKSP present an increase of about 20% in the average speed when compared to the baseline solution, while for FOX this increase is about 13%. This means that DSP and RKSP are handling traffic jams more efficiently than FOX under this configuration. However, when at least 50% of vehicles are re-routed, the increase of the average speed for both DSP and RKSP when compared to the baseline solution starts to decay (12% when 100% of vehicles are re-routed), while for FOX it is rather constant, at about 16%. This result shows that FOX is more efficient in handling traffic jams when a high number of vehicles are re-routed.

A similar behavior can be observed when we analyze the travel time (Figure 4.11(b)). When 25% of vehicles are re-routed, all protocols reduce the travel time by about 25% when compared to the baseline solution. However, when more vehicles are re-routed (at least 25%), FOX is the solution that induces the highest decrease in the travel time. This result shows that in FOX, vehicles are being re-routed through alternative routes that are faster than the routes selected by DSP and RKSP. This fact is corroborated by the result shown in Figure 4.11(c), which shows the reduction in the time that vehicles remain stopped at traffic jams. As can be seen, when 100% of the vehicles are re-routed, FOX reduces the stopped time by about 58%, while for DSP and RKSP this reduction is about 50% and 45%, respectively.

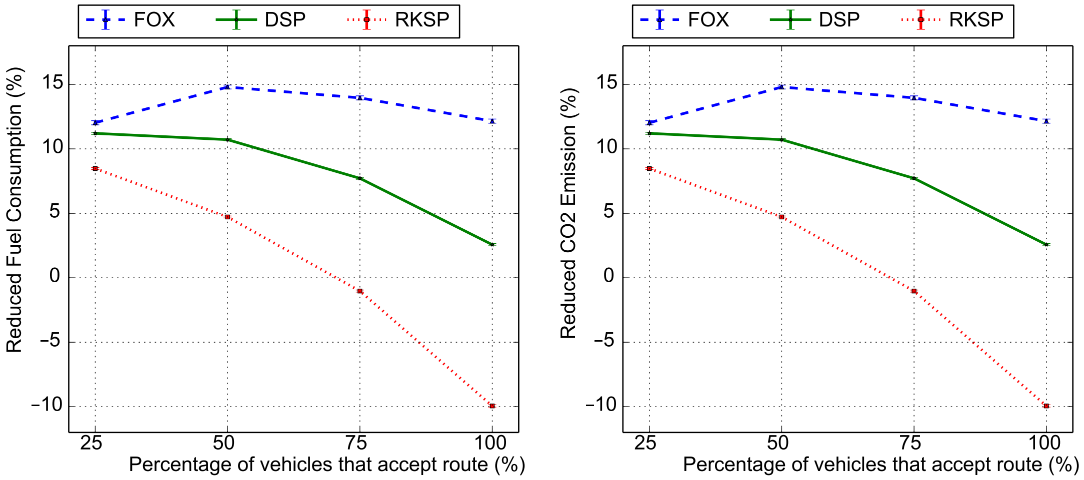
When considering the length of the alternative routes provided by the solutions (Figure 4.11(d)), we can see that all solutions suggest to vehicles alternative routes that are



(**a**) Average speed (**b**) Traveled time



(**c**) Stopped time (**d**) Traveled distance



(**e**) Fuel consumption (**f**) CO2 emission

Figure 4.11: Simulation results.

longer than the original ones. Notice, however, that in FOX, alternative routes are not 10% longer than the original ones, while in DSP and RKSP, alternative routes can be 15% and 30% longer, respectively. These results can be explained by the fact that FOX employs a parameter that controls how much longer alternative routes can be.

Finally, figures 4.11(e) and 4.11(f) show the reduction in fuel consumption and CO_2 emissions when compared to the baseline solution. As can be seen, independently of the number of vehicles re-routed, FOX reduces fuel consumption and CO_2 emissions by about 10%. An interesting fact in these results is that when at least 75% of vehicles are re-routed, the fuel consumption and CO_2 emissions for DSP and RKSP actually increase. This is strictly related to the longer alternative routes provided by these solutions, as shown in the previous result. In summary, these results show that when at least 75% of vehicles are re-routed, the FOX re-routing mechanism is better suited to handle traffic jams when compared to the other solutions. When 100% of vehicles are routed, services have a reduction in their efficiency, as many cars are sent on a route that is also congested due to routing. However, FOX has the lowest reduction among the evaluated solutions.

Evaluation with Different Configurations

We now vary the parameters *AoK* and *re-routing interval* for the FOX service, as shown in Figure 4.12. As can be seen, the performance of the FOX service improves with the growth of the radius of the *AoK*. This occurs because a great *AoK* implies more knowledge about the map (e.g. more roads and its traffic conditions), thus it is possible to calculate a more effective alternate route.

Moreover, a shorter *routing interval* is more efficient when the *AoK* is greater. When the *AoK* is small, the use of a constant re-routing interval implies in many best local routes being chosen, however they can be far from the best global solution, as shown in Figure 4.12(a). Using 150 seconds for the *re-routing interval* and the *AoK* with a radius of 1000 m, there was a 1% of improvement. For an *AoK* with radius 4000 m, FOX presented an improvement of about 18%. A long re-routing interval time is not effective under larger *AoK*, because the route calculated may have a road that will become congested before the next re-routing cycle. This fact can be observed in Figure 4.12(d), where the fuel consumption has the best results when considering an interval of 150 seconds, leading to a reduction of about 14%.

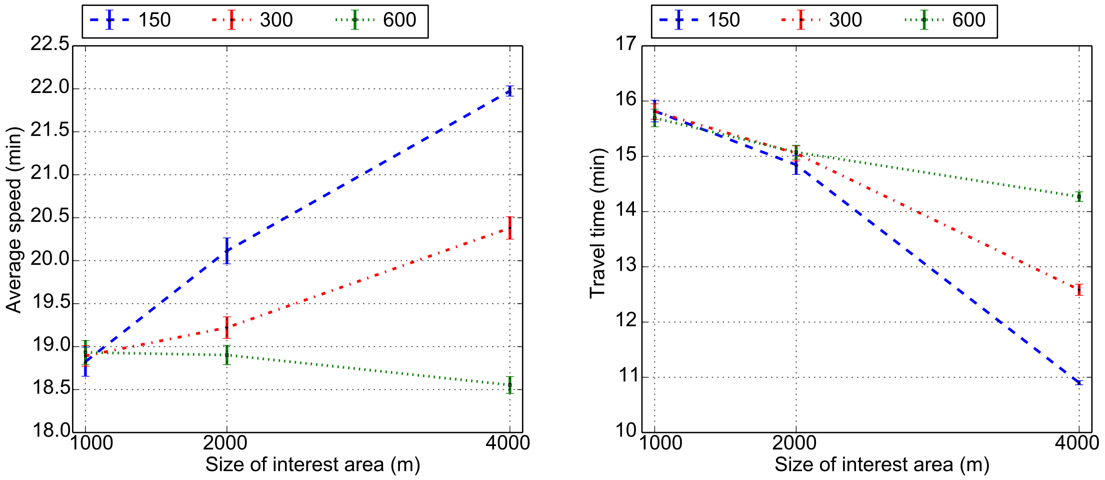
4.6.4 Experiment 3

A comparative evaluation between the FOXS and other literature solutions as presented in this experiment. The traffic conditions and the network communication were evaluated.

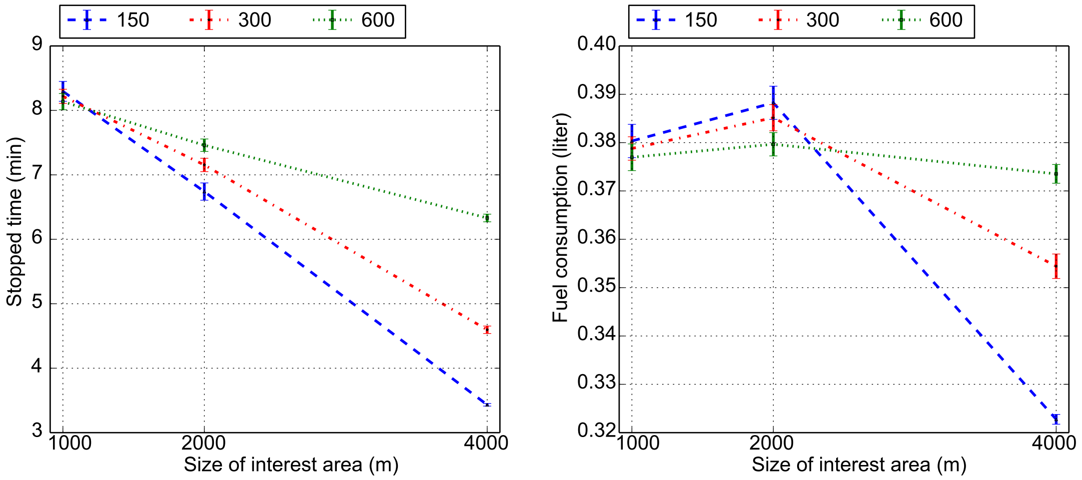
Methodology

The simulations were conducted using the network simulator OMNeT++ 5. For the simulation of traffic and mobility of vehicles, we used the SUMO 0.25.0 simulator.

We used realistic scenario from an urban region, Cologne-Germany city, that experiences traffic jam problems. The Cologne scenario was chosen to stress the system due



(**a**) Avarege speed (**b**) Traveled time



(**c**) Stopped time (**d**) Fuel consumption

Figure 4.12: Evaluation of the area of interest by routing on FOX.

to its more complex structure with fewer alternative routes and a considerably higher vehicle density. Furthermore, the network parameters for all simulations were set to 18 Mbit/s at the MAC layer and the transmission power to 2.2 mW, resulting in a coverage of approximately 300 m under a two-ray ground propagation model [126]. Table 4.6 shows the simulation parameters and values used in our evaluation.

Table 4.6: Simulation parameters Experiment 3.

Parameters	**Values**
Transmission power	2.2 mW
Communication range	300 m
Bit rate	18 Mbit/s
Beacons	4s
Alternative routes (k)	3
Confidence interval	95%
AoK	3000 m
Route size factor	25%
Interval to request new route	120 s

All evaluated solutions follow the same operation flow (see again Figure 2.12). Each Cloudlets collect traffic data and classify only the roads within their coverage area. Vehicles in DSP do not disseminate any information about the traffic and it uses the road length as road classification. The routing interval is set to 120 s for all solutions and three alternatives routes are used for FOXS, CHIMERA and PAN3 to obtain a fair evaluation. This number of alternative routes was chosen because it has the best results for all solutions in the evaluated scenarios. For the solutions mentioned above, we evaluated the traffic efficiency and the impact on the network, the computational resource, and the scenarios' comparative analysis. To do this, the evaluation was divided into two stages: Traffic Efficiency and Network and Resource Cost. For *Traffic Efficiency* evaluation, we consider the following metrics:

- *Traveled time:* the average travel time from the starting point to the destination of all vehicles;
- *Stopped time:* the average time spent stuck in traffic jams for all vehicles;
- *Average speed:* the average speed of all vehicles;
- *Traveled distance:* the average distance that all vehicles traveled;
- *Fuel consumption:* the average fuel consumption of all vehicles to traverse the whole route;
- *CO_2 emission:* the average CO_2 emissions for all vehicles during their trip.
- *Planning Time Index (PTI):* measures the reliability of the ratio of the 95% travel time to the ideal flow on the same path (e.g., a PTI of 2 means that for a 25-min trip in free flow traffic, a time of 50 min should be planned).

- *Route size Histogram:* the histogram of the number of routes by its size grouped into intervals of 500 m.
- *Empirical Cumulative Distribution Function (ECDF) of the routes size:* the ECDF of routes size.
- *PTI by route size:* PTI of a group of routes with similar sizes in 500 m interval.
- *PTI Utility metric:* the percentage of influence of the *PTI by route size* in the *PTI* result.

To verify the behavior of the solutions according to the route size, we present the metrics: Route size Histogram with its ECDF and PTI by route size presenting the route size distribution and the relation between the PTI and the route size range. The PTI utility metric shows the route size influence on the quality of the result in a specific scenario. The *Network and Resource Cost* were evaluated in the following metrics:

- *Transmitted messages:* the total number of messages transmitted (excluding beacon messages, which are used in all solutions);
- *Collisions per packets sent:* the percentage of collided packet per all packet sent;
- *Network delay:* the average time to spread messages to all vehicles (in milliseconds);
- *Application delay:* the average time for the application to receive the new route when requested, with the service response time and retransmission time when necessary (in milliseconds);
- *New route accepted:* the average of new routes accepted per vehicle in simulation;
- *Cloudlet routes computed:* the average of routes computed per Cloudlet;
- *Cloudlet computation heat-map:* representing the amount of routing executed by regions.

Cologne Scenario

Cologne scenario is a sub-part of a TAPAS Cologne scenario [132] (presented in Figure 4.13) that includes the 24 h of the mobility trace of the vehicles, obtained through real monitoring of the city traffic. Due to the large scale of the TAPAS scenario, we used a more critical traffic time (7 am to 8 am) and a sub-area of 4.5 km^2 of the Cologne downtown. This area possesses the greatest density and flow of vehicles, thus maintaining the representation of the scenario for our analysis. This scenario has approximately 14.000 vehicles inserted during the simulation and roads with 70 km of total length.

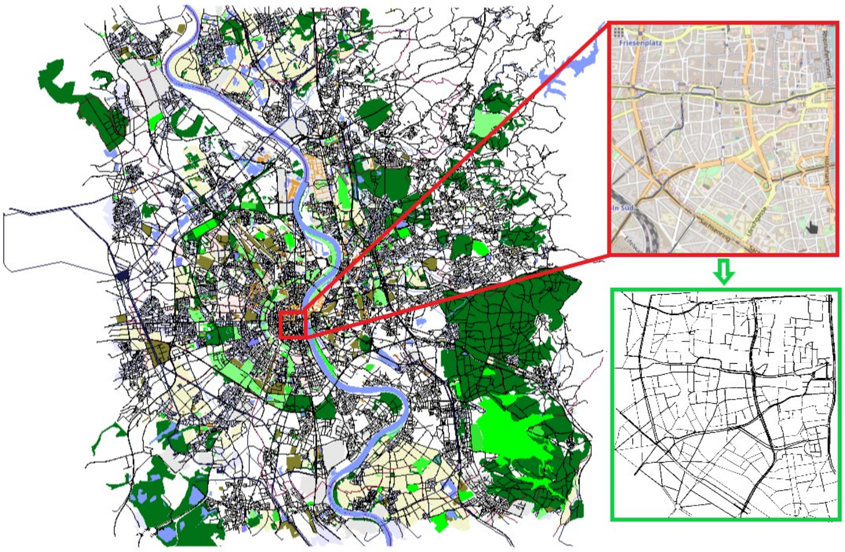


Figure 4.13: Topology of the Cologne Scenario.

Traffic Efficiency Evaluation

Figure 4.14 presents values and *BASE* related percentage of all metrics for the implemented solutions concerning traffic efficiency. The *BASE* is the results obtained in a scenario without the intervention of any routing service.

DSP reduced the distance traveled in 2.32 % since the shortest route is always chosen. PAN1 and FOXS had the lowest route increase with 1.35 % and 1.53 %, respectively (Figure 4.14d). Note that CHIMERA increases the route size by 3.78 % related to *BASE* and 2 times related to FOXS. This is because CHIMERA does not have a mechanism to evaluate the size of the route as FOXS.

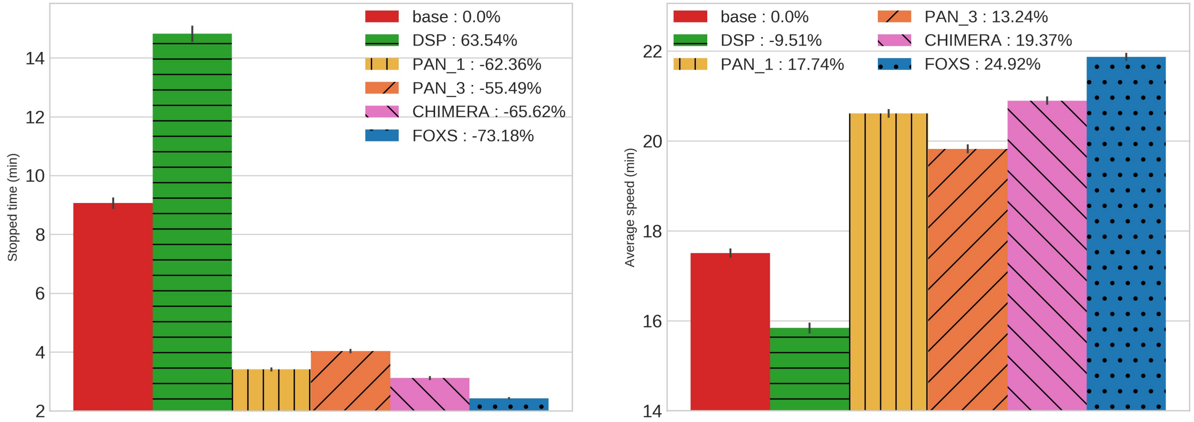
The Stopped time (Figure 4.14a) was reduced for PAN1 (62.36 %), PAN3 (55.49 %), CHIMERA (65.62 %), FOXS (73.18 %) and the Average speed (Figure 4.14b) was increased PAN1 (17.74 %), PAN3 (13.24 %), CHIMERA (19.37 %) and FOXS (24.92 %) in relation to *BASE*. Note there is a high difference between FOXS, PAN1, PAN3 and CHIMERA to *BASE*. This is caused by the large volume of vehicles in the scenario and by the great impact generated during the choice of alternative routes since the roads have quite a different size. However, FOXS chooses better routes than all solutions because the alternative route is chosen in a probability way reducing the chance of choosing a wrong route and it considers the size of the new route beyond the route classification.

The profit reached by FOXS in previous metrics is reflected in the improvement of the fuel consumption (reducing 28.25 %, Figure 4.14e), of the CO2 emission (reducing 28.25 %, Figure 4.14f) and of the travel time (reducing 53.53 %, Figure 4.14c). Concerning the PAN1 and PAN3, which differs only by the alternative routes, it was observed that for a real scenario, the random choice of a set of best routes does not provide a good result as presented in Figure 4.14.

The quality of the congestion control of the solutions evaluated using the PTI is presented in Figure 4.15.

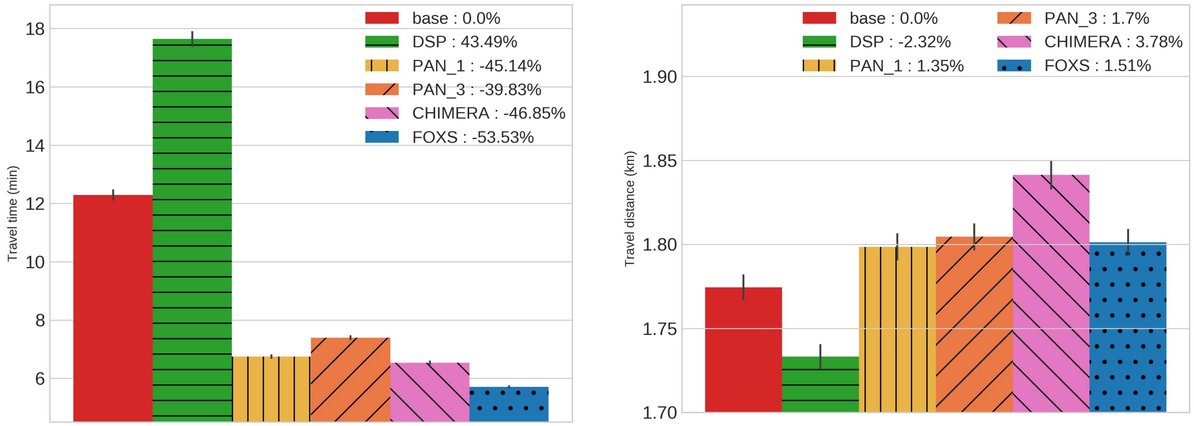
The metric PTI was reduced in 49 % by FOXS, in 44 % by CHIMERA and in 39 % by PAN1 compared to *BASE* showing the efficiency of these solutions. These results show that FOXS is able to better handle the city traffic, corroborating with the results obtained in Figure 4.14. In Figure 4.16 the route size histogram and route size ECDF is presented.

As can be seen, the Cologne scenario has short routes where 80 % of routes are shorter than 2.500 m (see Figure 4.16b) and approximately 60 % of the routes are between 1.500 m and 2.500 m (see Figure 4.16a). Short routes hamper a good response of routing solutions since the set of alternative routes will be smaller. To verify which solutions work better according to the route size, Figure 4.17 shows the PTI for 500 m intervals. We found that FOXS shows a reduction in PTI when the route size increases. These two graphs show that FOXS is better on larger routes in this specific scenario. The analysis of the PTI Utility metric (see Figure 4.18) presents that 75 % of more congested routes have a size between 1.000 m and 2.500 m.



(**a**) Stopped time.

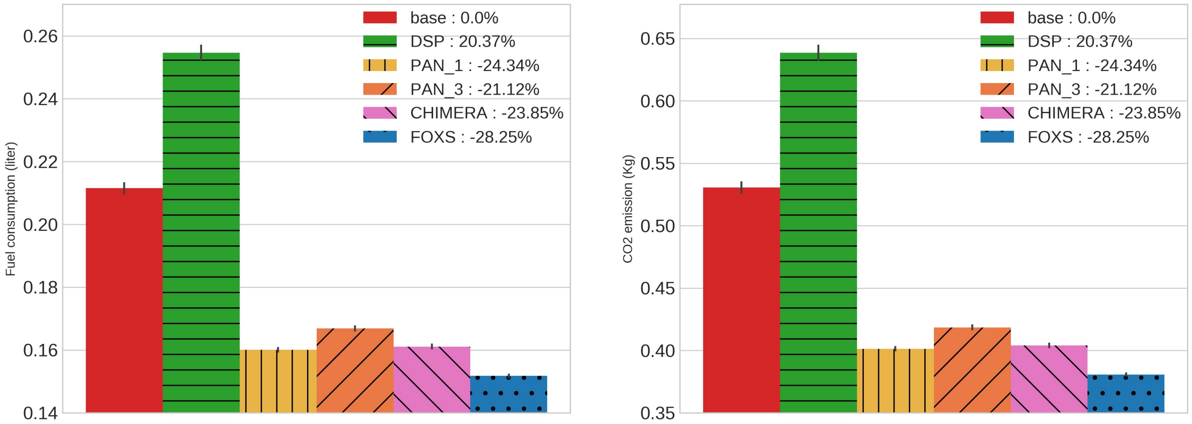
(**b**) Average speed.



(**c**) Traveled time.

(**d**) Traveled distance.

Figure 4.14: Traffic Efficiency Results—Cologne Scenario. *Cont.*



(**c**) Fuel consumption. (**d**) CO2 emission.

Figure 4.14: Traffic Efficiency Results—Cologne Scenario.

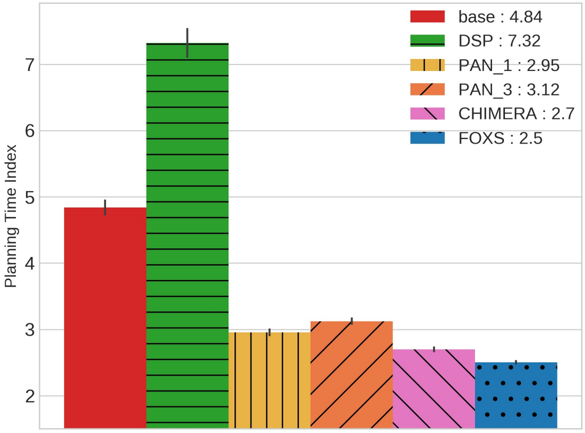
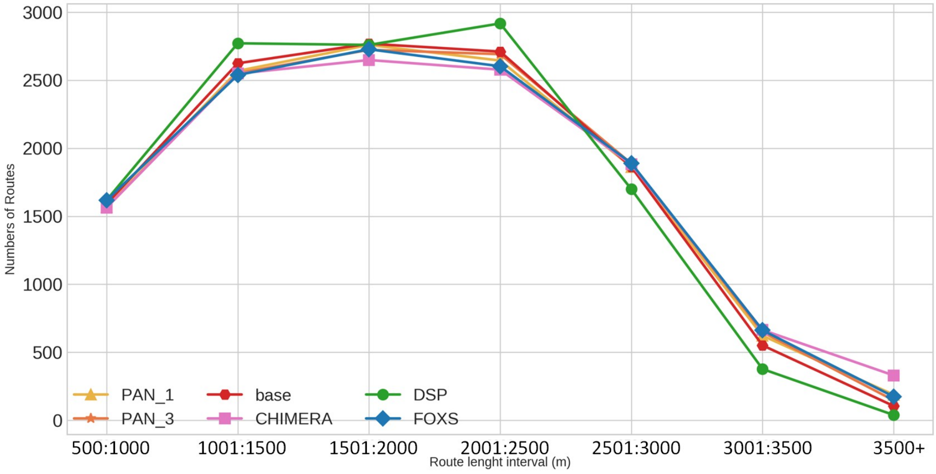
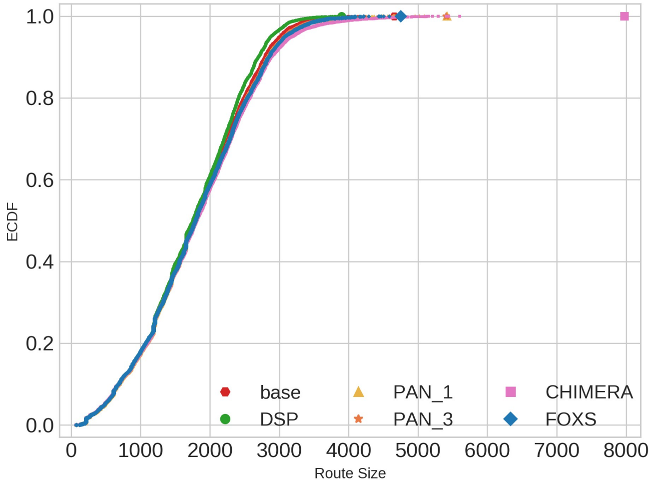


Figure 4.15: Planning Time Index (PTI)—Cologne Scenario.



(**a**) Route size Histogram.



(**b**) ECDF - Route Size.

Figure 4.16: Route Size Analysis—Cologne Scenario.

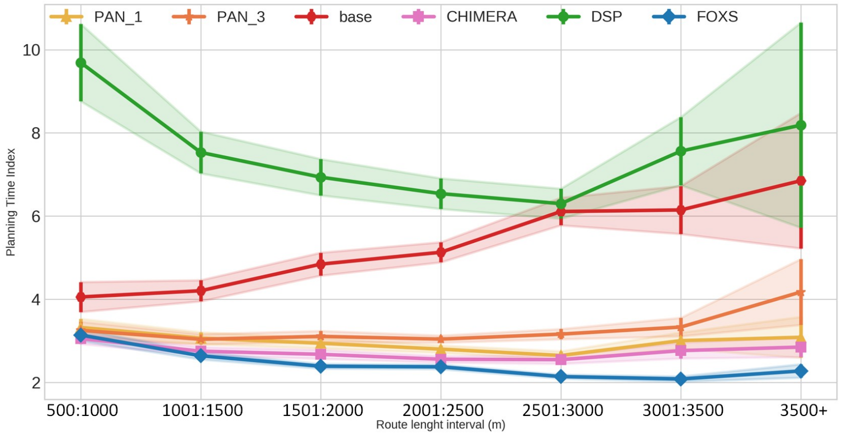


Figure 4.17: PTI by route size—Cologne Scenario.

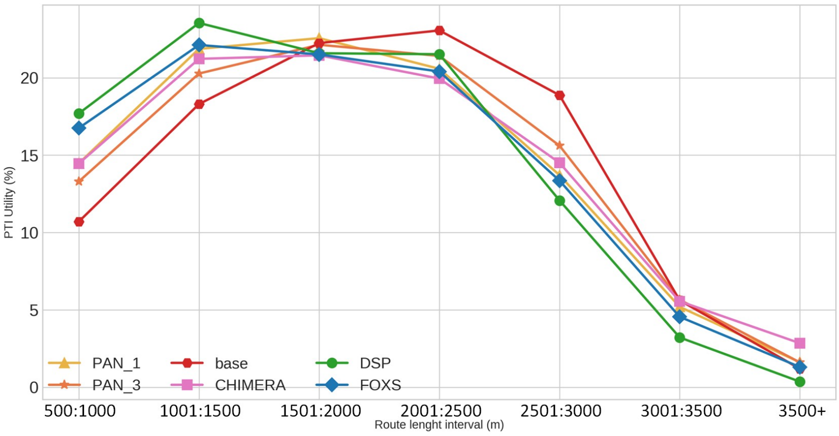


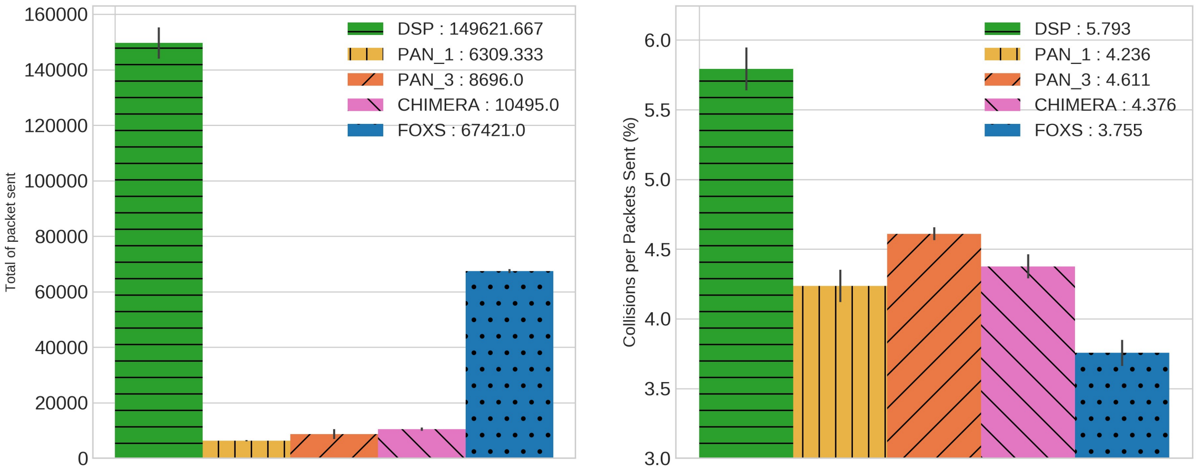
Figure 4.18: PTI Utility metric—Cologne Scenario.

Network and Resource Cost Evaluation

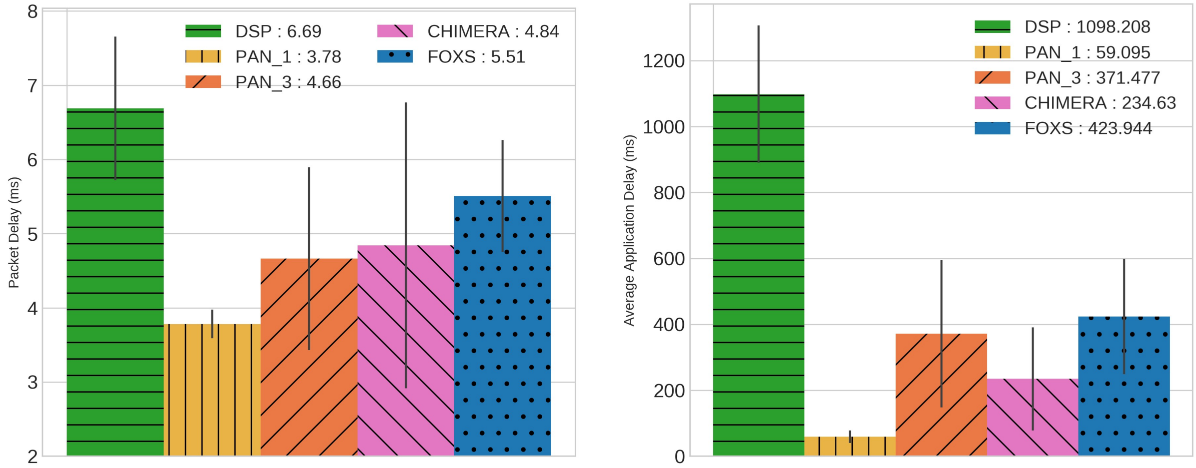
Analyzing the network metrics, in Figure 4.19 we can see the results of the defined metrics.

Figure 4.19a shows the total number of messages during the simulation. Note that FOXS sends more messages compared to the CHIMERA, PAN1 and PAN3. However, FOXS has a better average of the routes attributed to vehicles (see Figure 4.19e) compared to PAN1 (0.117 reroute per vehicle), PAN3 (0.233 reroute per vehicle) and CHIMERA (0.225 reroute per vehicle) (Figure 4.19e). Thus, the greater number of routed vehicles in the right way increase the city's traffic quality. Note that the high number of routes accepted does not always produce a good result as seen in DSP (2.121 reroute per vehicle).

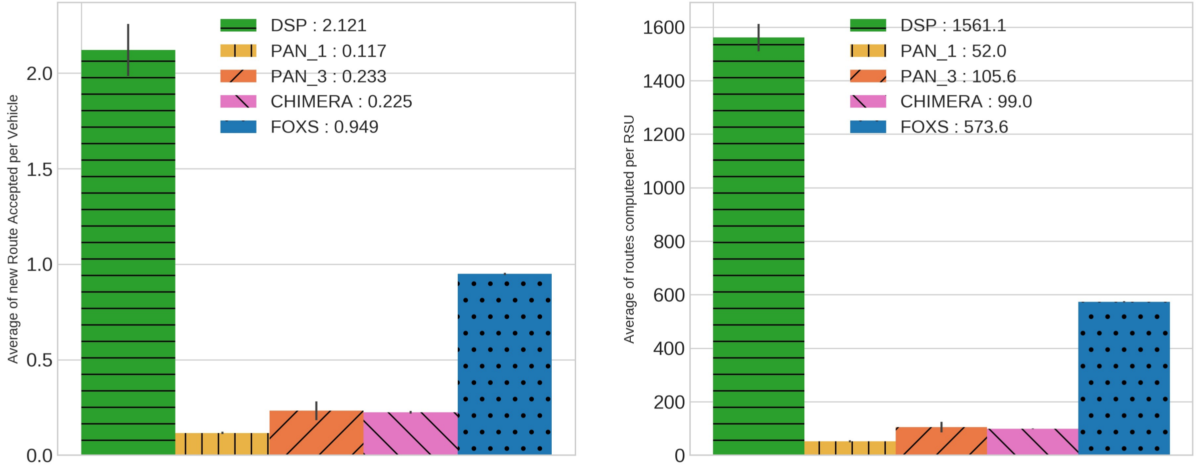
The average of routes computed by each RSU-Cloudlet is presented in Figure 4.19f. FOXS had the average of 573.6 routes computed and PAN1, PAN3 and CHIMERA had the average of 52, 105.6 and 99 routes computed, respectively. The high number of routes computed by FOXS is not a problem because of the use of the Fog paradigm, which distributes the calculation of the routes by regions in several Cloudlets. The behavior of the number of packet collisions (Figure 4.19b) shows that FOXS has a result of approximately 11.5 % better than PAN1. Despite the large volume of packets generated by FOXS, the network delay (see Figure 4.19c) did not have a significant difference to CHIMERA and PAN3. This was due to the messaging scheduling mechanisms and effective routing by distributing the vehicles in the scenario. Thus, avoiding the concentration of vehicles in a region competing with the network channel. The application delay (Figure 4.19d) is influenced by the number of routes computed and the others network metrics. We can see that although FOXS has a large number of computed routes and packet sent, the number of packets collisions was the smallest among the solutions. Thus, its application delay had a value similar to all evaluated solutions.



(**a**) Total of packet sent. (**b**) Collisions per packets sent (%).



(**c**) Network delay. (**d**) Application delay.



(**e**) Average of new route accepted per vehicle. (**f**) Average of routes computed per RSU.

Figure 4.19: Network Cost Results—Cologne Scenario.

Figure 4.20 geographically shows this route calculation distribution of FOXS on the map exposing the areas with the most demand for Cloudlets. Using this analysis, Cloudlets can be implemented in each region with computing and communication power based on user demand, thus saving equipment costs. Moreover, a load-balance can be made sharing resources to more occupied Cloudlets. Specifically for FOXS, this information can be used to adjust its settings.

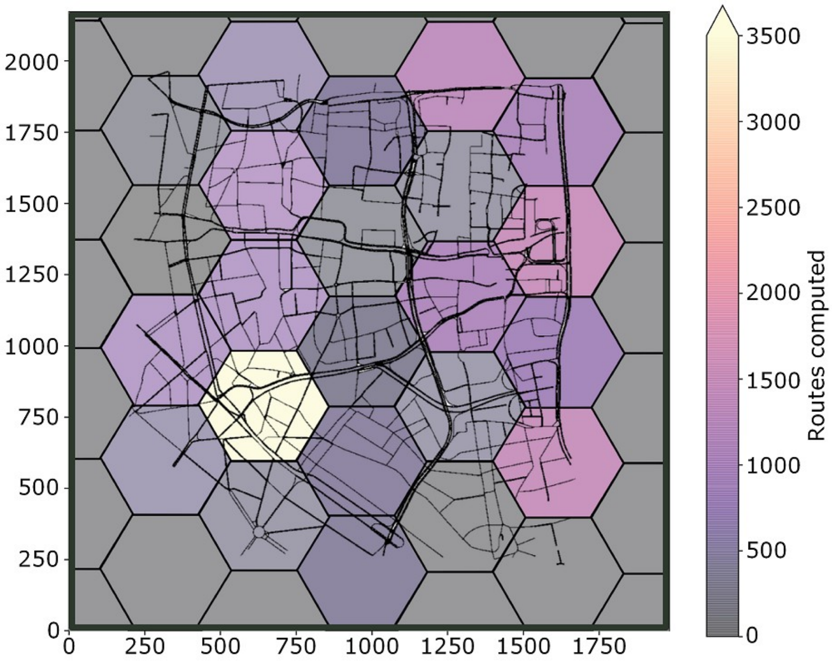


Figure 4.20: Routes computed per region—Cologne Scenario.

Finally, such results show that FOXS is better suited to handle traffic jams in the evaluated scenario.

4.6.5 Results Final Remarks

This results section presented a set of experiments made to validate the FOXS and to make explanatory research about the behavior of routing services in relation to different configuration parameters and scenario characteristics. The evaluations were divided into three experiments.

Experiments 1 present the influence and relationship between a number of alternative K routes calculated by the FOXS, and the size of AoK. The results show that greater values of k alternative routes can lead to worse results, where $k = 3$ alternative routes provide the best result to our service. This happens because the higher the number of alternative routes is, the greater the chances of the number of worse routes calculated and placed on the k route set, and can be probably chosen by the service.

Experiment 2 presented an analysis of the number of vehicles that accepts the route suggested, the routing interval, and the AoK size. The results show that the proposed service reduces the travel time by approximately 32 % and the stop time by 59 % in relation with other literature solutions. Besides this, FOXS has better performance on congestion control when the number of network vehicles that accept the route is high (75 % to 100 %). However, this is not the case of the other solutions evaluated, showing that a solution that is not well developed, the congestion can get worse when routing all vehicles on the network sending the vehicles to new congestion. Another system configuration characteristic noticed is that large AoK is more efficient with short routing intervals. This happens because when a long route is calculated, this route may have roads that may be congested in the next routing cycle. Otherwise, when the AoK is small, using a constant re-routing interval implies in many best local routes being chosen, however, they can be far from the best global solution.

Finally, Experiment 3 presents an extensive evaluation comparing FOXS with other literature solutions. Several metrics for evaluating the traffic efficiency and network and resource cost were used to evaluate these services in two realistic urban scenarios with different characteristics. We also included in the new evaluation metric created by the authors, the PTI Utility metric, that measures the influence of the PTI by route size on the route suggestion service. When compared with literature solutions, FOXS shows a reduction in stop time by up to 70 %, the CO_2 emissions by up to 29 % and, the PTI by up to 49 %. When considering communication evaluation metrics, FOXS reaches a better result than other solutions in the packet collisions metric (up to 11.5 %) and on the application delay metric (up to 30 %).

Chapter 5

Proposed Service II FOXS-GSC - Fast Offset Xpath Service with HexagonS Communication

This chapter describes the distributed communication protocol called FOXS-GSC. It selects relay nodes based on message retransmission to disseminate data over urban VANET scenarios. FOXS-GSC considers the geographic position of the destination node and the position of neighboring nodes concerning the (re)transmitting node. To do this, vehicles must maintain local knowledge of their neighbors obtained from the beacons sent in the ITS.

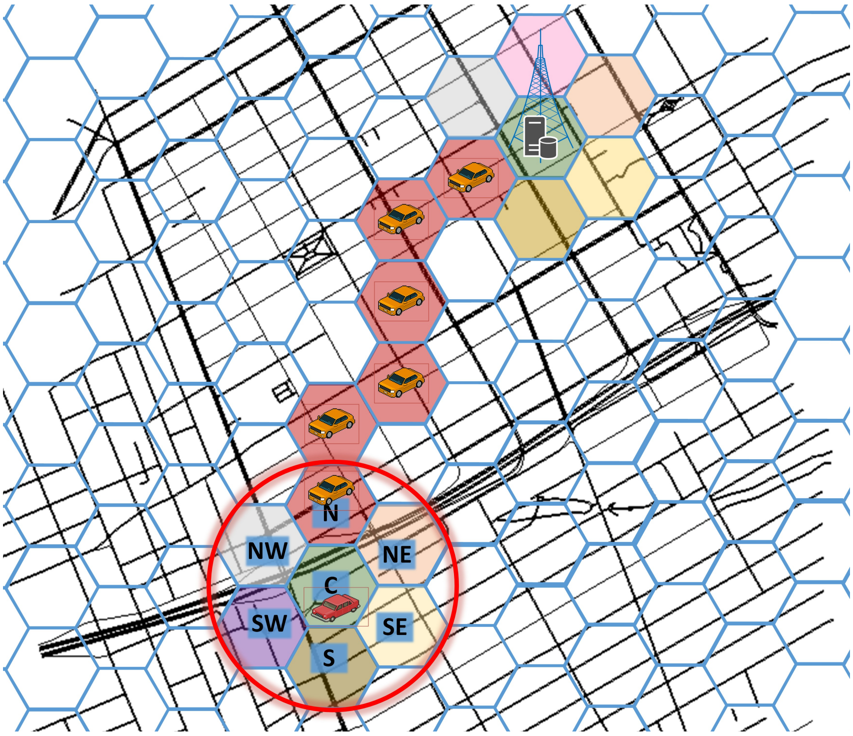


Figure 5.1: Hexagonal multihop communication.

The protocol discretizes the scenario in sectors with a hexagonal shape (Figure 5.1).

Using this division, each vehicle populates its adjacent sectors with vehicles in the neighborhood. Thus, the messages are carried out in multiple hops between adjacent sectors that contain at least one vehicle (relay) until reaching the destination (red hexagons in Figure 5.1). Note that there can be several paths between the origin and the destination, but the algorithm will choose the path with the fewest hops. However, another metric for the graph weight can be easily implemented in the presented solution, such as network or sector occupation.

5.1 Overview

The FOXS-GSC – Fast Offset Xpath Service with hexaGonS Communication is a congestion control service based FOXS. The major upgrading in FOXS-GSC is the proposed communication protocol allowing multihop communication.

FOXS-GSC uses the Fog computing paradigm that brings resources closer to devices to acquire advantages when using this paradigm, among them: low latency, wide geographical distribution, real-time interaction, mobility, scalability, and extensibility. To do this, Cloudlets are used which are Fog computational entities with processing and communication capabilities. The Cloudlets are integrated into network infrastructure Roadside Units (RSUs) spread in the scenario. Cloudlets are capable of direct communication between other Cloudlets using a direct link or via the Internet.

The proposed communication protocol permits the use of less Cloudlets in the scenario, thus the full map coverage is not necessary since the multihop communication is available between vehicles and Cloudlets. Using the proposed communication protocol it is possible to place Cloudlets only at strategic points to serve the largest number of vehicles with the least number of hops between the vehicles and the Cloudlets (e.g., high dense road or crossroad). The distribution of Cloudlets can also be based on the computing load to load balance.

The proposed protocol aims to send a message using multihop communication between two specific nodes. In the case of the FOXS-GSC solution, the vehicle requests a route to the Cloudlet and it responds to the vehicle requesting the new route. As a premise, all vehicles know the Cloudlet's geographic location on the map, so a request message from the routing service will be sent to the nearest Cloudlet.

5.2 Sectorization of the Scenario

The regular hexagon is a polygon that has desirable properties for the proposed communication protocol. Hexagons have 6 sides and corners. In a regular hexagon, the size between the corner to the center is the same as the hexagon side. Hexagons also have 6 adjacent neighbors with equal distance between its center and the center of all its neighbors (see Figure 5.2).

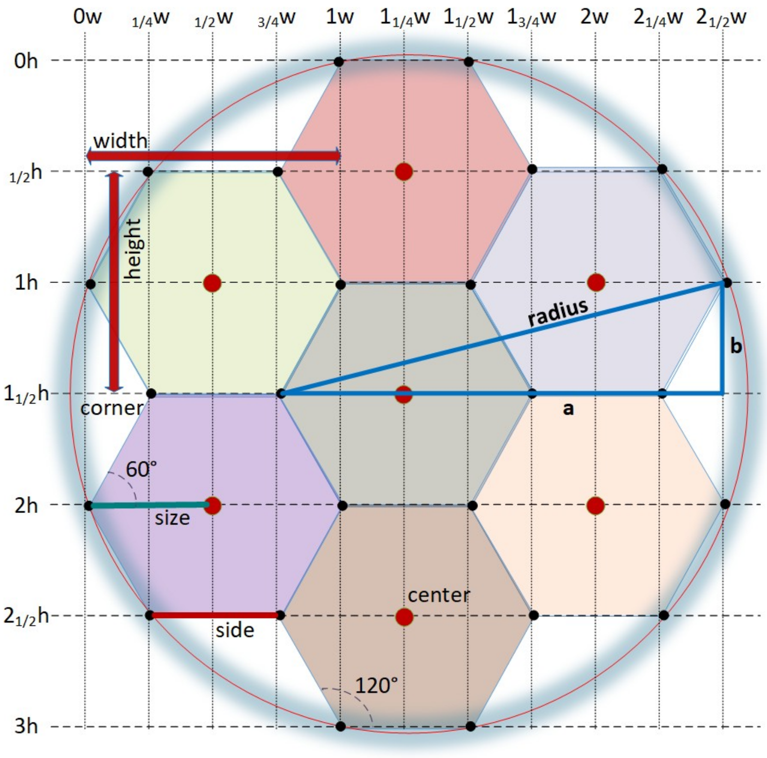


Figure 5.2: Hexagon properties.

With these characteristics, hexagons can uniformly cover the plane efficiently without overlapping and it is possible to circumscribe the hexagon and its adjacent neighbors regularly [11, 82]. The hexagon center is shared with the circle center.

As a definition, the radius of the circle that circumscribes the hexagon and its 6 adjacent neighbors is calculated in such a way that the vehicle located in the central sector (Figure 5.2) can communicate with any vehicle located in the neighboring sectors.

To calculate the hexagon size to permit these communication characteristics an equation was deduced. To do this, the hexagon representation in a cartesian plane is made (see Figure 5.2). The *size* units are the distance between any corner to the hexagon center. The angle of 2 adjacent sides is 120 grades. Hexagon width (w) is

$$w = 2 * size, \tag{5.1}$$

and height (h) is

$$h = \sqrt{3} * size, \tag{5.2}$$

where $\sqrt{3}$ comes from sin(60).

The relations between h and w are defined by Eq. 5.3,5.4:

$$w = (2 * h)/\sqrt{3}. \tag{5.3}$$

$$h = (w * \sqrt{3})/2. \tag{5.4}$$

The triangle was drawn on the plane with the line connecting the furthest corners of 2 neighboring hexagons as the hypotenuse. The hypotenuse, represented by the *radius* is the node's communication coverage. The largest cathetus is represented by a and the smallest by b. Equations 5.5 and 5.6 present the cathetus size inferred concerning the Cartesian plane:

$$a = \frac{4}{7}w. \tag{5.5}$$

$$b = \frac{\sqrt{3}}{4}w. \tag{5.6}$$

Finally, the Pythagorean theorem is applied as follows (Eq. 5.7):

$$radius^2 = \left(\frac{4}{7}w\right)^2 + \left(\frac{\sqrt{3}}{4}w\right)^2. \tag{5.7}$$

Thus the formula for calculating the size of the hexagon is obtained (Eq.5.8):

$$w \approx 0.5547 * radius. \tag{5.8}$$

For sectorization of the map, the Hexagonal Binning algorithm [28] is used, having as input the dimensions of the map and the hexagon height (h).

5.3 Neighborhood knowledge discovery

The proposed communication protocol uses a vehicle's on-Board Unit (OBU), that is a device mounted on vehicles that has processing power and allows DSRC/802.11p communications with other OBUs or RSUs, to acquire knowledge of neighboring vehicles that are at one hop. Regarding this, vehicle periodic beacons contain information such as vehicle ID and current location. ITS vehicles already send beacons, thus not increasing network overheads.

Upon receiving a beacon, the protocol stores vehicle IDs, location and timestamp of receiving. The protocol stores this data in one of the 6 lists, representing the neighboring sectors (N, NE, SE, S, SW, NW) of the receiving vehicle, according to the current position of the vehicle that sent the beacon. Figure 5.3 shows the vehicle positions and sectors and the neighbor list filled with the vehicle's id and timestamp of beacon received.

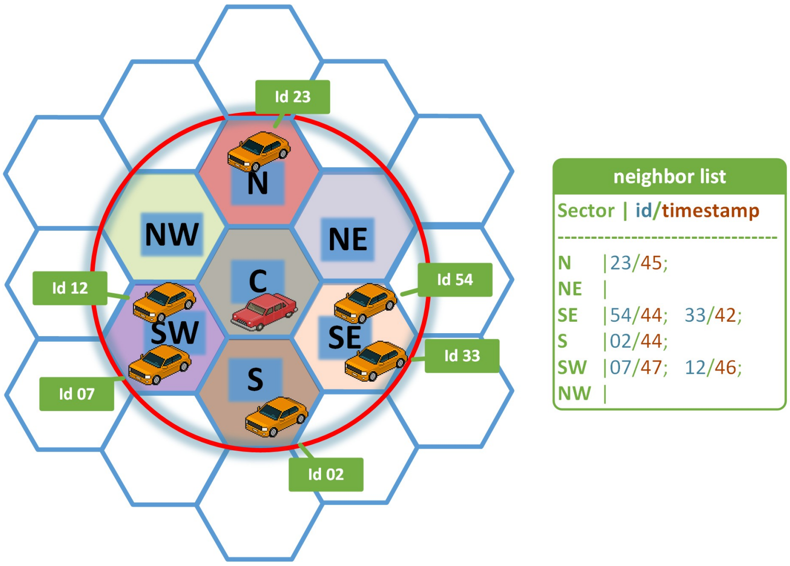


Figure 5.3: Vehicle Sectors and Neighbor List.

Those lists with information are used to choose the next hop in the message transmission. However, information from neighbors quickly becomes out of date due to vehicle mobility. Thus, the list must be constantly updated within a periodic update window. The update period was estimated according to the average time required for the vehicle to change sectors at an average speed of the scenario. For example, a scenario with an average speed of $54km/h$ $(15m/s)$ the time for the vehicle to leave the sector with a size of 160 meters in which it is located in the center will be approximately 5 seconds. Several works estimate that a similar update window that generated satisfactory results [128, 135, 37].

5.4 Dissemination Process

The message forwarding process is performed by choosing the most suitable vehicle to forward the message (called relay) to the destination node. FOXS-GSC has two wireless communicating components, Cloudlets, and vehicles. Cloudlets receive messages directed to it that originated from a vehicle. Taking this into account, Cloudlets process the message and respond to the originating vehicle when it is a route request. Cloudlet also exchanges messages between them to update traffic conditions. Vehicles send periodic messages containing information about road conditions, and when necessary, send a request route to Cloudlets. Vehicles can also work as a relay node to retransmit messages according to the proposed protocol rules.

The proposed protocol has two routing methodologies. These methodologies are related to the destination of the message sent. If the message is directed to Cloudlet (e.g., route request, road conditions) the only information about the destination is the Cloudlet geographic position. As Cloudlets in FOXS-GSC only send a message to a specific vehicle when is requested by it, the response message (e.g., new route) to vehicle has the list of all relay nodes used to forward the message to the destination.

The proposed protocol adds the following information to the original message presented in Table 5.1.

Field	**Description**
origin_sector	sector that the initial node is within
origin_node	initial node id
destination_sector	sector that the destination node is within
destination_node	destination node id
nexthop_sector	next sector
nexthop_vehicle	next hop (relay)
visited_sectors	list of visited sectors
visited_vehicles	list of vehicle used as relay
recovery_tag	number of backtrack steps

Table 5.1: Protocol message format (MSG)

5.4.1 Message routing addressed to Cloudlet

This routing method occurs when only the geographical location of the destination is known, that is, a message sent to a Cloudlet. When a vehicle prepares a message addressed to Cloudlet, it checks if the Cloudlet is in its communication coverage. If so, the message is sent directly. Otherwise, it will be necessary to use the proposed protocol for multihop communication.

The proposed multihop protocol uses the prior knowledge of the geographical location of the Cloudlets and also the knowledge of neighboring nodes acquired through the received beacons. Figure 5.1 presents the discretized scenario as well as the behavior of the proposed protocol. As can be seen, the vehicle in the sector C wants to send a message to the nearest RSU. When the routing process starts, it is necessary to check which neighbor sectors have at least one vehicle and which sectors have been visited to avoid looping. Therefore, to find the best path to the destination, the protocol does a high-level routing using the shortest path algorithm (e.g., Dijkstra) in the graph representing the sectors discretized in the scenario. When the next sector is chosen, one vehicle located in this sector is selected and the message will be addressed to it. This method is executed at each hop by the respective relay until reaching the destination. The Algorithm 4 formalizes this protocol.

Algorithm 4 has as its initial node the sector in which the vehicle is located and as the final node the destination Cloudlet sector. The other input parameters is the neighbor list, the list of hexagon and its positions, the list of Cloudlets id and positions, vehicle id, position, and current sector (lines 1 to 5).

To avoid the protocol choosing a sector that does not have any vehicles or creating a loop when the message is routed to the destination, sector nodes can be removed from the hexagon sector graph in some cases. The fist case is presented in Algorithm 4, line 7, where it is checked which neighboring sectors (*neighbors_list*) contain at least one active vehicle. Neighboring sectors that do not have any vehicles, are temporarily removed from the sector graph. Second case (Algorithm 4, line 12), all sectors contained in the *visited_sectors* list received in the message to be retransmitted, are temporarily removed from the graph (*Gtmp*).

With this, the next-hop is chosen based on the shortest path algorithm between the current sector and destination sector (see Algorithm 4, line 15). After choosing the sector, it is verified if there is more than one vehicle that is a relay candidate in this sector. If so, the vehicle with the most recent beacon is chosen as a relay (see Algorithm 4, line 16). Finally, the message fields is updated (see Algorithm 4, lines 17 to 20). During the message retransmission process, note that the ID of vehicles relays and visited sectors are stored in the lists *visited_vehicles* and *visited_sectors*, respectively (see Algorithm 4, lines 19, 20). These lists in addition to being used in this method, will be used in the service response message process which will be presented in the next section (5.4.2).

This process is repeated until reaching the destination.

Algorithm 4: Next Sector Chooser

```
Input  : //
       1 neighbors_list[] // neighbors list [N, NE, SE, S, SW, NW] containing vehicle ID and
     timestamp of the received beacon;
       2 GH_sector // hexagon graph sectors;
       3 Cloudlet_list // Cloudlet positions list;
       4 MSG // message to be (re)transmitted;
       5 vehicle // current vehicle characteristics (id, position, current_sector);
Output: the MSG updated to (re)transmitted;
    // make a temporary copy of hexagon graph (GH_sector)
 6  $Gtmp \leftarrow copy(GH\_sector)$;
 7  foreach $s \in neighbors\_list[N, NE, SE, S, SW, NW]$ do
        // removes (s) sector from temporary hexagon graph if it is empty
 8      if s is empty then
 9          $Gtmp.remove(s)$;
10      end
11  end
12  foreach $v \in MSG.visited\_sectors[]$ do
        // removes all visited sectors from temporary hexagon graph
13      Gtmp.remove(v);
14  end
    // get next hop sector between current sector and destination sector using temporary sector graph
15  $nexthopSector \leftarrow getNexthopSector(Gtmp, vehicle.current\_sector, MSG.destination\_sector)$;
    // get a vehicle inside of next sector with more recent activity
16  $nexthopVehicle \leftarrow getNexthopVehicle(neighborslist[nexthopSector])$;
    // update message fields
17  $MSG.nexthop\_sector \leftarrow nexthopSector)$;
18  $MSG.nexthop\_vehicle \leftarrow nexthopVehicle)$;
19  $MSG.visited\_sector.append(vehicle.current\_sector)$;
20  $MSG.visited\_vehicle.append(vehicle.id)$;
```

In case of retransmission errors, the proposed protocol has a recovery mode. The recovery technique is similar to the backtracking technique in which if a relay node does not have neighboring sectors with vehicles suitable for retransmission, the previous relay will have to try another sector closer if it exists. For this, a timer is triggered for each message sent. If the vehicle does not listen to the retransmission in this period, the message is updated (the *recovery_tag* is added 1, the sector previously used for transmission is placed in the *visited_sectors* list) and the process of choosing a new next-hop sector to retransmission is initiated (Algorithm 4). The *recovery_tag* is used to limit the number of backtrack steps.

Figure 5.4 shows a Flowchart describing the behavior of sending the message from a vehicle to a Cloudlet. The protocol receives a message (MSG) that can originate from the vehicle itself (e.g., route request) or from a retransmission message from a neighboring vehicle (stage A in Figure 5.4). It is checked whether MSG is destined for an adjacent sector (B). If it is, the MSG is sent directly (G). Otherwise, Algorithm 4 is executed to choose a retransmission node (C) then the message is sent (D). After sending the message in stage (D), a timer is activated to wait for retransmission of the message by relay node chosen by Algorithm 4 (E). If the message has listened, the protocol goes to start again (H). If not, recovery mode is activated (F) and it checks if $MSG.recoverytag$ is less than defined N value. If yes, the $MSG.recoverytag$ is added 1 and the Algorithm 4 is executed again (C). If not, its goes to (H). When stage (H) is reached, the protocol goes to stage (A) to wait for a new message.

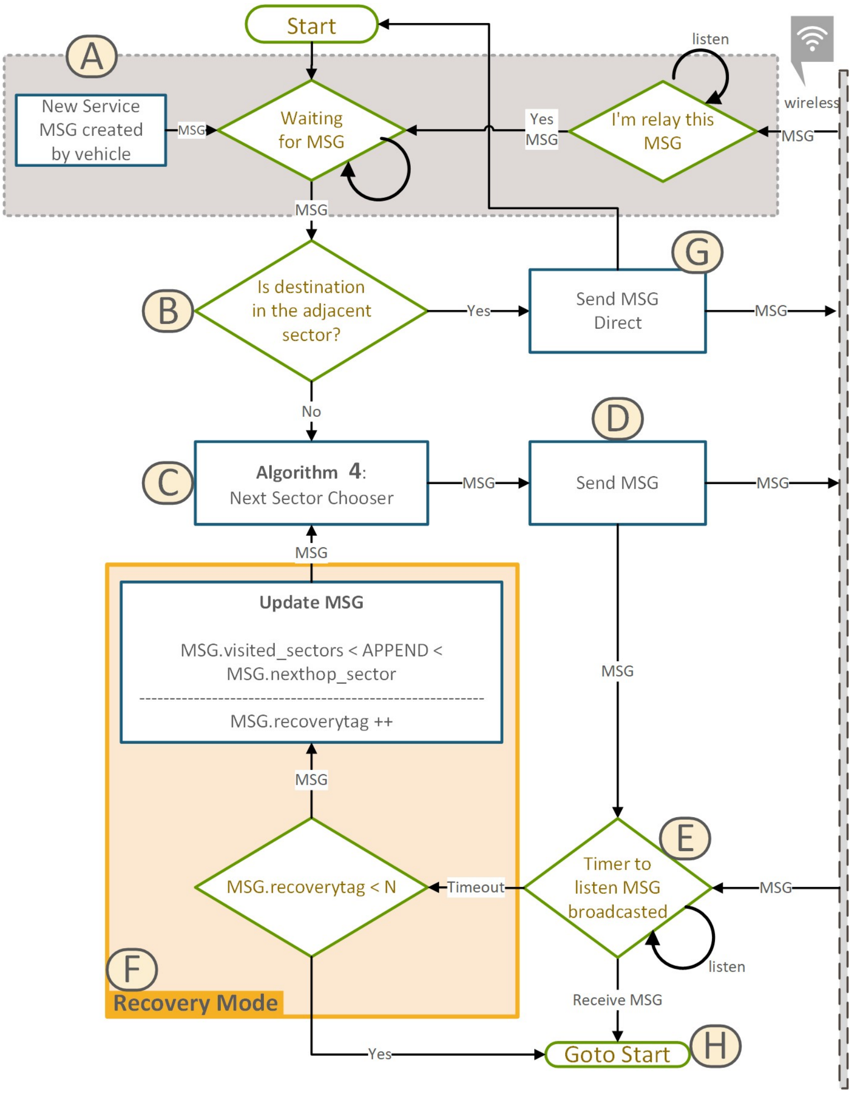


Figure 5.4: Flowchart: Message routing addressed to Cloudlet.

5.4.2 Response message routing

This routing method is used when the hops (relays) necessary to reach the destination node are known. For the FOXS-GSC case, the method is used when a new route for the requesting vehicle is answered. As the service response time is fast enough compared to network topology changes (vehicles moving), the response message for the requesting vehicle uses the list of relay nodes (MSG.*visited_vehicles*) contained in the request message.

Figure 5.5 shows a Flowchart describing the protocol behavior of sending the message from a Cloudlet to a vehicle. In stage (A), the protocol receives a message (MSG) created by the Cloudlet service (e.g., route response) or a retransmission message, where the vehicle that received the message is a relay. At each retransmission of the response message, the protocol removes the last ID from the list and copies its value to the next hop field (MSG.*nexthop_vehicle*) (step B). Afterwards, it checks if the next hop is the destination (B). If so, the message is transited and the protocol is finalized. Otherwise, the message is sent (D) and a timer to wait for the retransmission of the message on the next hop is activated (E).

This methodology is applied due to the likelihood of a transmission failure (e.g., topology change). Thus, the recovery mode is used. The timer is set to 100ms which is the time for two-channel changes in 802.11p protocol. If the message is listened, the protocol ends. Otherwise, the recovery mode is activated (F). Thus, the message is updated removing all visited sectors and vehicles from the MSG.*visited_sectors* and MSG.*visited_vehicles* consecutively. After this, the retransmission process is switched to the protocol phase presented in the previous step 5.4.1. Once the stage (H) is reached, the flow goes to start again. This is possible because the relayed message contains the sector of the destination vehicle and the fact that the response time is also fast enough to ensure that the vehicle is still located in the same sector when it sent the original requisition message.

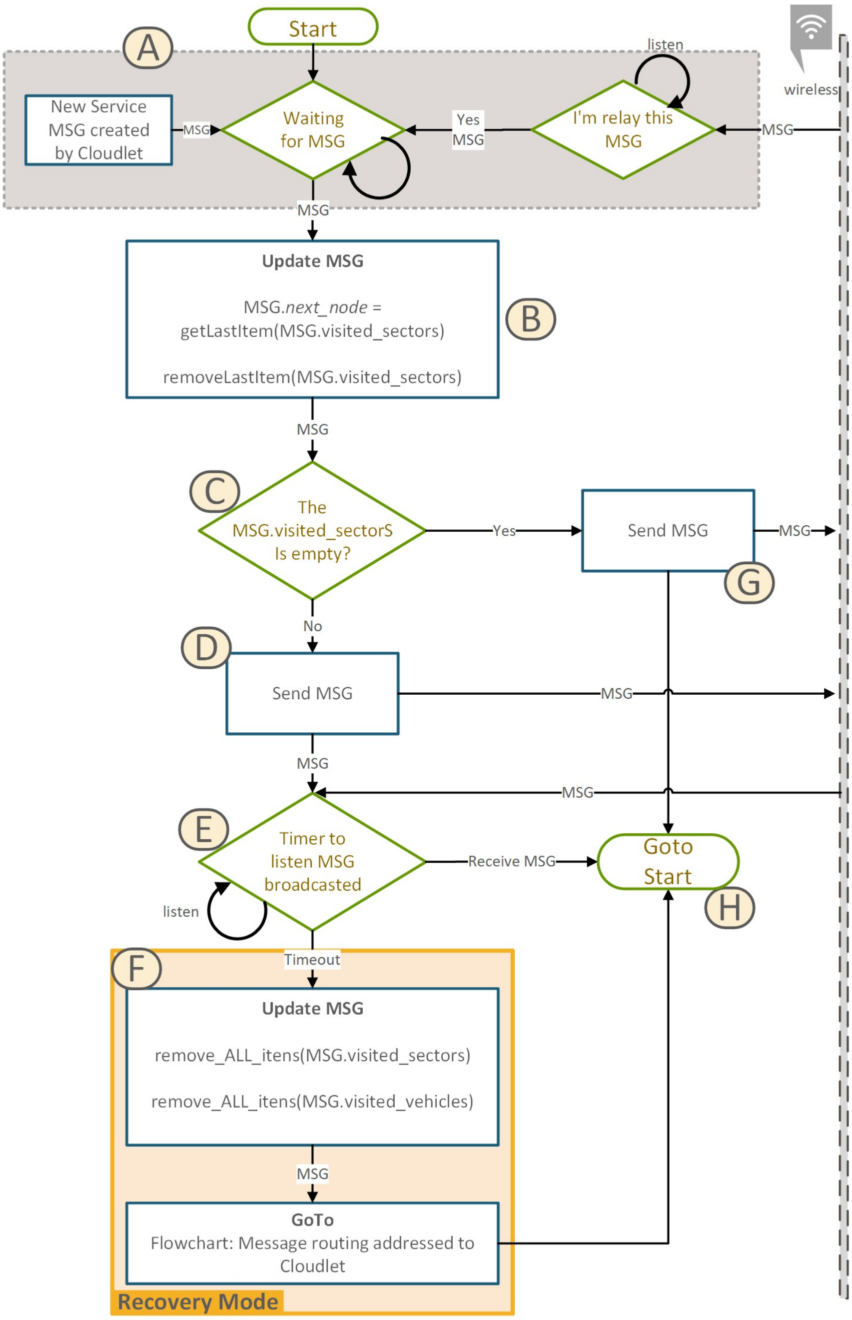


Figure 5.5: Flowchart: Response message routing.

5.5 FOXS-GSC Operation

The FOXS-GSC operation is divided into two stages, which are: (i) Data Gathering process and Data Processing, where the data transmission and road traffic classification occurs; (ii) Service Delivery, where the algorithms and methods are used to compute the new route and deliver it to the users.

5.5.1 Data Gathering process and Data Processing

The information acquisition process is made as follows. Vehicles periodically send a message ($Agg\_info\_message$) with aggregating data (average speed and occupation) of roads in its sector to a near Cloudlet to acquire knowledge of road conditions. For vehicle's aggregate knowledge of roads into the current sector, vehicles sent beacon messages containing the vehicle ID, position, and the road data that is traveling. Therefore, the vehicles that receive this beacon will store this data and aggregate it with the data acquired by other beacon messages. The data aggregation is performed using the Exponential Moving Average (EMA) technique to obtain the new road weight as described in 4.3.

Vehicles periodically send information about the roads contained in their current sector to the nearest Cloudlet. In order to prevent several vehicles in the same sector sending similar information in a short period, the following procedure is illustrated in Figure 5.6. All vehicles in the scenario schedule to send in N seconds (2 vehicle beacons time of 4s each was used) a message containing the data of the roads in their sector. When this time is reached, the message is sent and the counter restarts (Figure 5.6 stage A). When vehicles inside the same sector listen to an aggregation information message, it aggregates the new information and reschedules the message dissemination for the next N seconds interval (Figure 5.6 stage B).

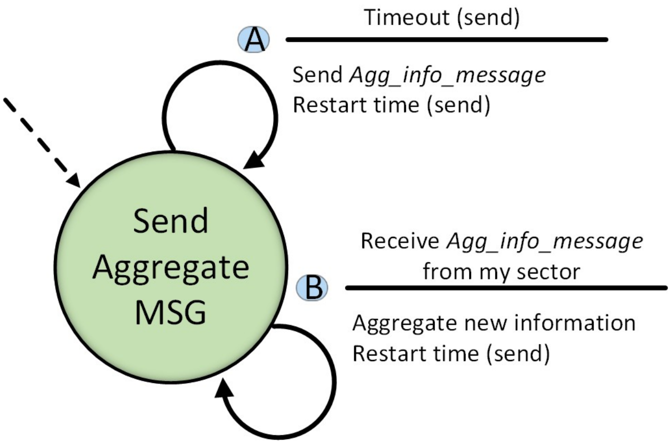


Figure 5.6: State diagram: Send aggregate information message.

During the sending of the aggregated messages, the relay vehicles and all vehicles that listen to the message during the multihop transmission from the starting sector to the Cloudlet, collects the information contained in the message about the conditions of the roads in the originating sector of the message. This is made for the congestion control process to be presented in the next step (5.5.2 - Service Delivery).

As Cloudlets receive only knowledge of the sectors closest to their geographical position, Cloudlets share road conditions among all Cloudlets to have full knowledge of the scenario. To do this, a FOXS publish/subscribe protocol is used that is based on MQTT (Message Queuing Telemetry Transport) (described in 4.3).

5.5.2 Service Delivery

At this stage, the FOXS-GSC congestion control process is carried out. The request route decision is made by vehicles in two possible situations: first, a periodic route request occurs, acting with congestion prevention and balancing traffic since Cloudlets use a congestion control algorithm that balances traffic; the second situation is when it is on the information of the roads acquired by the vehicles in the previous step, a sign of congestion is verified on a road to which it will travel. In both situations, the vehicle in question requests a new route to the nearest Cloudlet. The Cloudlet that receives this request will execute a Boltzmann probabilistic congestion control algorithm (presented in 4.4) and a new route is answered to the requesting vehicle using the proposed communication protocol.

However, the requested vehicle route message may not reach the nearest Cloudlet for several reasons, such as network congestion causing packet loss or not having enough vehicles on the network to allow communication between the source and the destination. Predicting this problem, when requesting a new route, the vehicle starts a timer (time of a beacon). Reaching the time limit, the vehicle understands that the message was lost. Thus, if the Cloudlet is in coverage, the message is sent again, otherwise it performs the routing on its own using the local knowledge obtained in the step presented in 5.5.1. The local vehicle routing is less effective compared to that performed by Cloudlet as the vehicles only have restricted knowledge of the traffic conditions of the roads in the scenario.

5.6 FOXS-GSC Performance Evaluation

This section presents the performance evaluation conducted to validate the FOXS-GSC solution. Simulations varying the number of RSUs and vehicle density are made to analyze the efficiency of proposed multihop communication protocol and the route suggestion.

The experiments are divided into: i) Network Communication; ii) Scenario Coverage Evaluation; and iii) Traffic Efficiency Evaluation.

i) Network Communication experiment: evaluate the impact of different vehicle densities on FOXS-GSC and other literature solutions (Flooding, DESTINy [91]);

ii) Scenario Coverage Evaluation: analyze FOXS-GSC related to the impact of scenario coverage varying the number of RSU and its relation with scenario density;

iii) Traffic Efficiency Evaluation: assert the traffic efficiency of FOXS-GSC varying the scenario coverage.

For all experiments, the simulations were conducted using the network simulator OMNeT++ 5. For the simulation of traffic and mobility of vehicles, we employed the SUMO 0.25.0 simulator. The tools and general methodology that conducted these evaluations are the same as those presented in 4.6.1.

5.6.1 Network Evaluation

Methodology

This evaluation considered an area of $1km^2$ from New York City, USA. The scenario configuration ranges the density of the vehicles from 50, 150, and 500 $vehicles/km^2$. We used 1 RSU to obtain partial scenario coverage, therefore evaluating the multihop efficiency of the proposed solution.

The network parameters for all simulations was set to 18Mbit/s at the MAC layer and the transmission power to 2.2mW, resulting in a coverage of approximately 300m under a two-ray ground propagation model [126]. Once the simulation remains stable (i.e., all vehicles were in the scenario), periodically all vehicles sent a message to the nearest Cloudlet. When the Cloudlet receives a message, the Cloudlet sends a reply to the vehicle that sent it. To prevent all vehicles from sending messages at the same time, the period is added with a random variable. This experiment simulates the behavior of the proposed service where a route is requested and the Cloudlet responds with the suggested route.

Table 5.2 shows the simulation parameters and values used in our evaluation.

Table 5.2: Simulation parameters.

Parameters	**Values**
Map	Manhattan downtown
Map Size	1 km^2
Transmission power	2.2 mW
Communication range	300 m
Bit rate	18 Mbit/s
Beacons	4s
Number of RSUs*	1, 4, 8
Confidence interval	95%
Message sending period	3 s

*Only 1 RSU to experiments on Section 5.6.1.

The following metrics were considered to evaluate the performance of data multihop communication protocols:

- *Total of packets re-transmitted*: displays the number of retransmissions required to reach the destination;

- *Number of collisions per vehicle*: the total number of collisions per vehicle in the system;

- *Receive coverage*: the percentage of vehicles that receive an answer at least once during the simulation time; and

- *Percentage of messages received*: the percentage of messages sent that was answered.

Results

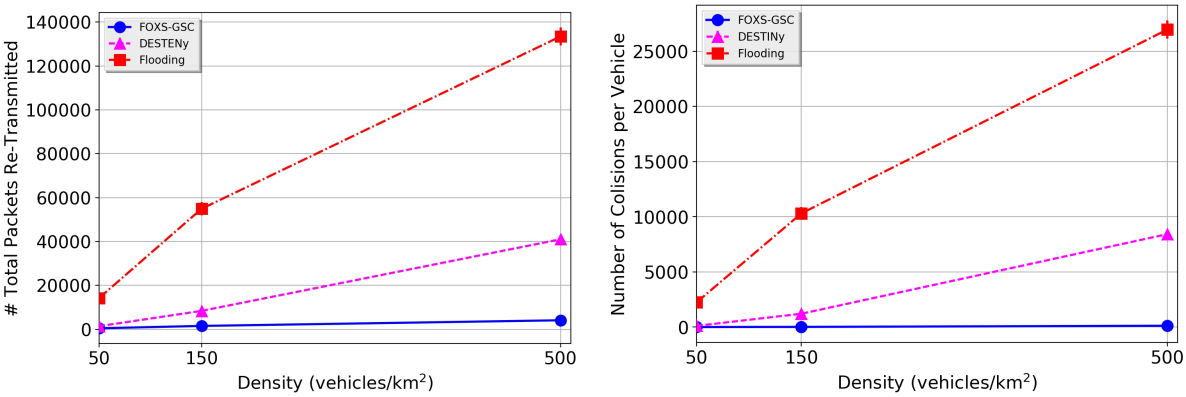
Figure 5.7(a) shows the number of messages re-transmitted. Flooding had the largest number of messages re-transmitted with more than $135,000$ messages in the scenario with $500 vehicles/Km^2$. This occurs because Flooding does not have a broadcast suppression mechanism.

The suppression mechanism of DESTINy, which is based on distance to the destination, reduces the number of messages by 70% in relation to Flooding. However many messages are still generated until the message reaches the destination, which is 90% higher than the FOXS-GSC with the same vehicle density. FOXS-GSC has an average number of re-transmissions per message sent equally to the number of sectors used on the path to the destination. Thus, keeping the number of messages in the system reduced.

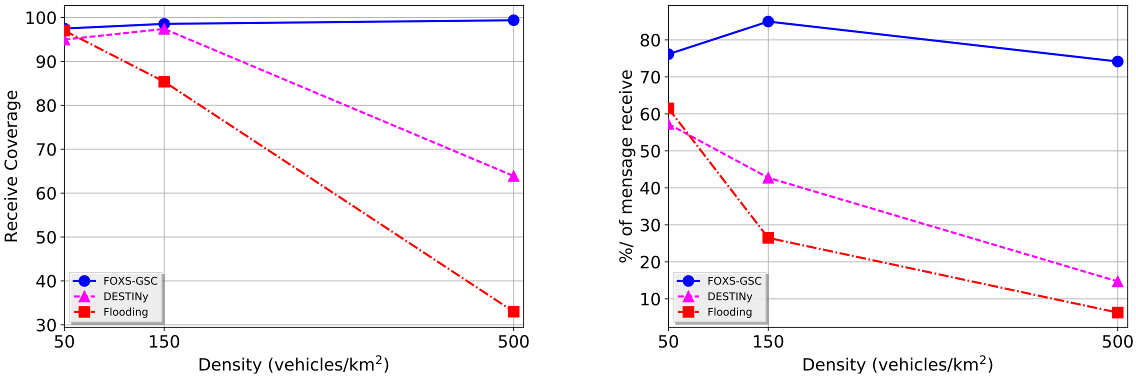
Figure 5.7(b) shows the packet collision related to the number of vehicles. FOXS-GSC keeps the collision number stable in relation to the number of vehicles with 120.4 messages for the density of 500 $vehicles/km^2$. Note that the Flooding solution number of collisions (26,000 for the density of 500 $vehicles/km^2$) grows much faster than that of DESTINy (8,000 on 500 $vehicles/Km^2$ density) and FOXS-GSC. This is related to the large number of messages re-transmitted (see Figure 5.7(a)).

Figure 5.7(c) shows the percentage of vehicles that sent and received the response of at least one message during the simulation. FOXS-GSC reached an average of 98% coverage. Flooding and DESTINy had a significant reduction in coverage with an increase in vehicle density covering only 33% and 64% of vehicles in the scenario with a density of $500 vehicles/Km^2$. These values are corroborated by the high number of collisions of these solutions. Thus, showing that FOXS-GSC can reach the entire monitored environment.

Figure 5.7(d) shows the percentages of messages sent and properly answered during the simulation. In the low-density scenario (50 $vehicles/km^2$) the percentage of messages answered was 79% for FOXS-GSC, 65% for Flooding, and 60% for DESTINy. With density increase, FOXS-GSC kept the average percentage of delivery at 74% whereas Flooding and DESTINy had a great reduction, reaching below 20% in the density of 500 $vehicles/km^2$. This was due to the large number of message collisions generated by these solutions. Thus, the proposed solution proved to be efficient both in a sparse scenario and in a more dense scenario.



(**a**) Total of packets re-transmitted. (**b**) Number of collisions per vehicles.



(**c**) Receive coverage (%). (**d**) Percentage of messages returned.

Figure 5.7: Network Results of literature evaluation.

5.6.2 Scenario Coverage Evaluation

Methodology

To assess the impact of the number of RSU on the proposed protocol, these experiments were performed varying both the number of RSU and the density of the scenario. The scenario used was the same as in the previous assessment (Manhattan $1km^2$) with densities of 50, 150, 500 vehicles per km^2. The variation in the number of RSUs was 1, 4, and 8. The scenario with 8 RSUs provides full coverage of the scenario. The metrics evaluated were: Total of packets re-transmitted, Number of collisions per vehicle, Receive coverage, and Percentage of message answered. The experiments work similarly to the previous experiment (Network Evaluation), where periodically all vehicles send messages to the nearest Cloudlet and it responds to the vehicle. The simulation parameters and values used in our evaluation were presented in Table 5.2.

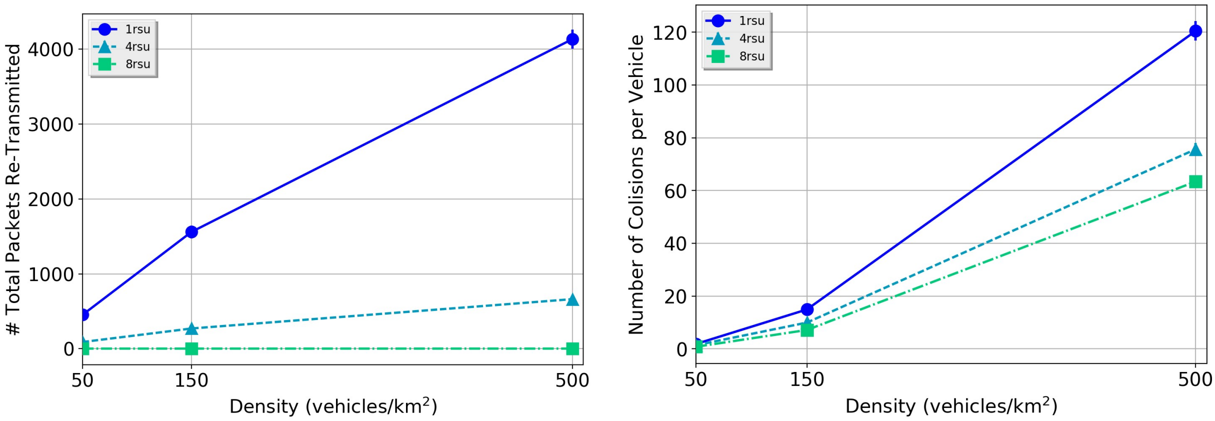
Results

In Figures 5.8(b) we note that the number of packet collisions increases with the need to use multihop communication since more messages are needed to reach the destination that will be the most distant. Note that the greater the RSU coverage, the fewer messages are needed to reach the goal.

Figure 5.8(a) shows the number of packets retransmitted. It can be seen that, as expected, the lower the number of RSU and the greater the density of vehicles is, the greater the number of messages re-transmitted. The method applied by FOXS-GSC that makes the intersection routing in an interactive way manages to keep the number of messages in the system low, even with a large number of vehicles and with reduced RSU coverage.

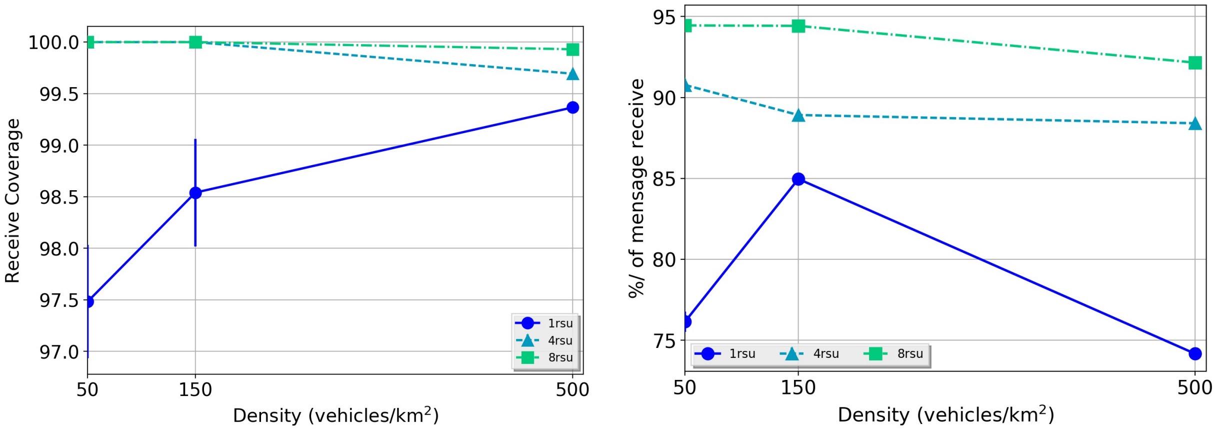
FOXS-GSC has good coverage as shown in Figure 5.8(c). When vehicle density is low, problems such as network disconnection reduce the coverage area. This problem can be reduced by applying transmission methodology such as store-carry-forward. The FOXS-GSC is the protocol used by the congestion control service, the proposed protocol does not implement this methodology since regions with low density have a low probability of having traffic congestion.

The percentage of responses for messages sent (see Figure 5.8(d)), as expected, is higher when the number of RSU is higher. Regarding the use of 1 RSU, when the vehicle density is 50 $vehicles/km^2$, the number of collisions (76%) is influenced by the network partition problem. As for the density of 500 $vehicles/km^2$, the result is influenced by the packet collision as shown in Figure 5.8(b). Note the peak for the density of 150 $vehicles/km^2$ in the scenario with 1 RSU. This is due to the fact that the scenario has a good number of vehicles to avoid network partition, but not many vehicles that produce several packet collisions, thus reducing the delivery rate.



(**a**) Total of packets re-transmitted.

(**b**) Number of collisions per vehicles.



(**c**) Receive coverage (%).

(**d**) Percentage of messages returned.

Figure 5.8: Network Results of RSU coverage.

5.6.3 Traffic Efficiency Evaluation

Methodology

For this evaluation, we chose the city of Ottawa, Canada. Ottawa scenario (Figure 5.9) represents a downtown region and was obtained from OpenStreetMap [1], and has an area of 8 km². The scenario has 409.42 km of total road length and 2,200 vehicles inserted during the simulation representing congested traffic. The mobility model used was random origin-destination.

---

[1] `http://www.openstreetmap.org/`

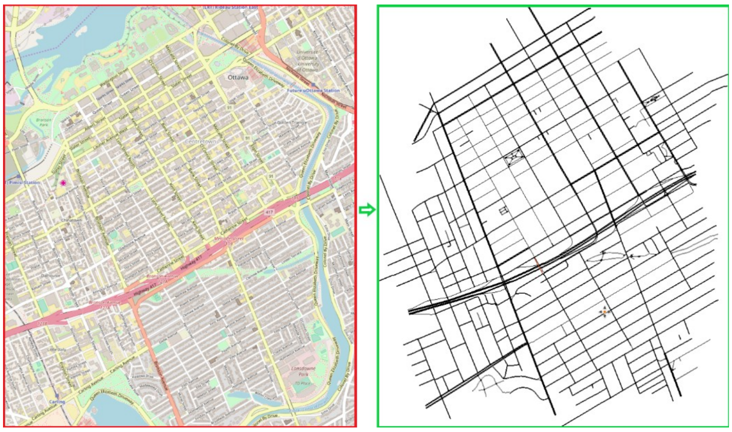


Figure 5.9: Topology of the Ottawa Scenario.

The network parameters were set to 18 Mbit/s at the MAC layer and the transmission power to 2.2mW. Table 5.3 shows the simulation parameters and values used in our evaluation.

Table 5.3: Simulation parameters.

Parameters	**Values**
Transmission power	2.2 mW
Communication range	300 m
Bit rate	18 Mbit/s
Beacons	4s
Alternative routes (k)	3
Confidence interval	95%
AoK	3000 m
Route size factor	25%
Number of RSUs*	1, 23, 50
Interval to request new route	120 s

*Only FOXS-GSC vary 1, 23, and 50 RSU. The other solutions use 50 RSUs providing full coverage.

The routing interval is set to 120s for all solutions and 3 alternative routes are used for FOXS-GSC, CHIMERA, and PAN3, therefore obtaining a fair evaluation. This number of alternative routes was chosen because it has the best results for all solutions in the evaluated scenarios.

For evaluating the traffic congestion service, the following metrics are used:

- *Traveled time*: the average travel time from the starting point to the destination of all vehicles;
- *Stopped time*: the average time spent stuck in traffic jams for all vehicles;
- *Average speed*: the average speed of all vehicles;

- *Traveled distance*: the average distance that all vehicles traveled;
- *Fuel consumption*: the average fuel consumption of all vehicles to traverse the whole route;
- *CO_2 emission*: the average CO_2 emissions for all vehicles during their trip;
- *PTI*: measures the reliability of the ratio of the 95% travel time to the ideal flow on the same path;
- *Route compute location*: the percentage of where the route was computed, Vehicle or Cloudlet.

Results

Figure 5.10 presents values and *BASE* related percentage of all metrics for the implemented solutions concerning traffic efficiency. For that evaluation, the FOXS-GSC was compared with DSP, PAN1, PAN3, and CHIMERA solutions. To evaluate the FOXS-GSC in different scenario coverages, 1, 23 and 50 RSUs were used. Where 50 RSUs provide full scenario coverage.

The solutions PAN1, PAN3, CHIMERA and, FOXS-GSC (with 1, 23, 50 RSUs) reduced the stopped time (Figure 5.10a) in 16.17 %, 13.68 %, 21.36 % and, (22.01 %, 30.46 %, 39.14 %) respectively in relation to the *BASE* solution. These results reflect a higher average speed, as presented in Figure 5.10b. FOXS-GSC has a better performance than all evaluated solutions, also in 1 RSU scenario. FOXS-GSC (with 1, 50, 23 RSUs) increasing the average speed by (9.04 %, 13.44 %, 17.47 %) because the FOXS-GSC uses a better road classification and the probabilistic mechanism to choose one of K alternative routes. Note also that the DSP had its stopped time increased by 90 % and the average speed reduced in 18 % compared to the *BASE*. This happens because DSP only calculates the shortest path for all vehicles, moving the vehicles to the same road. Consequently, creating a new congestion point unlike the other evaluated solutions calculates a new route when necessary and based on the road conditions.

Still considering the average speed metric, PAN1 (6.5 %) has better results than PAN3 (5.1 %). This happens because PAN3 calculates three alternatives routes and chooses one at random. Thus, the second and third bests routes may have different sizes compared to the best route due to the geography of the city map. This does not happen in FOXS-GSC, because the alternative route is chosen in a probability way reducing the chance of choosing a wrong route and the use of the route size factor. Similar behavior can be observed when we look at the travel time presented in Figure 5.10c. FOXS-GSC (1, 23, 50) manages better the urban traffic reducing (12.14 %, 17.34 %, 22.63 %) the travel time. FOX-GSC 50 had twice as good results as CHIMERA, which reduced 11.73 % compared to *BASE*.

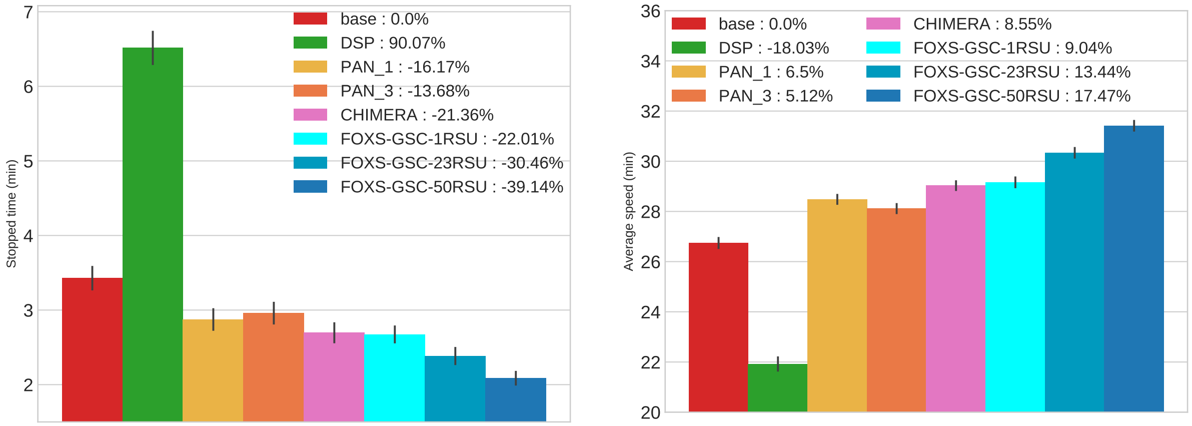
Analyzing the traveled distance (Figure 5.10d), only the DSP decreases the distance compared to the *BASE* (2.39 %), because it always chooses the shortest route to destination besides the other solutions that calculate the new route based on the road condition. Considering FOXS-GSC (1, 23, 50), alternative routes, on average, are not (0.99 %, 1.28 %, 0.39 %) longer than the original ones. These results can be explained by the fact that FOXS-GSC uses a parameter that controls how much longer alternative routes can be.

Figure 5.10f,e shows the CO_2 emission and fuel consumption, respectively. The traveled time, the stopped time, and the traveled distance have a direct impact on these metrics. The results present that FOXS-GSC 50 reduces three times the CO_2 emission and fuel consumption (10.27 %) compared with the CHIMERA that reduces 2.68 %.

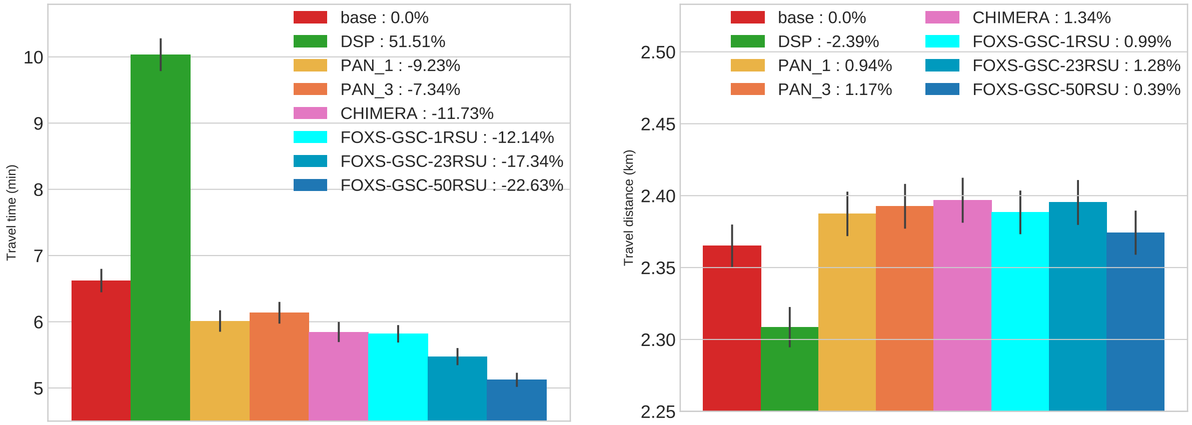
The quality of the congestion control of the solutions evaluated using the PTI is presented in Figure 5.11. Among the evaluated solutions, FOXS-GSC (1, 23, 50 RSUs) obtained PTI indices (2.68, 2.5, 2.32) lower than the compared solutions. Note that FOXS-GSC 50 is 19 % lower than the CHIMERA.

For all metrics evaluated except the distance traveled, FOX-GSC had its performance increased with the large scenario coverage. This happens because when coverage increases, more vehicles are able to communicate with the Cloudlet. Thus, receiving a more accurate route suggestion since Cloudlets have a global knowledge of the scenario. In the case of the metric distance traveled, the scenario with 50 RSUs, the Cloudlets with full knowledge, suggest better and shortest routes. For scenarios with 23 and 1 RSUs, more vehicles compute the new route using their local knowledge. Considering this, to avoid nearby congestion, FOX-GSC takes a longer route. However, as shown, the mechanism that limits the size of the route keeps the distance traveled statistically similar to BASE, PAN1, PAN3, and CHIMERA solutions.

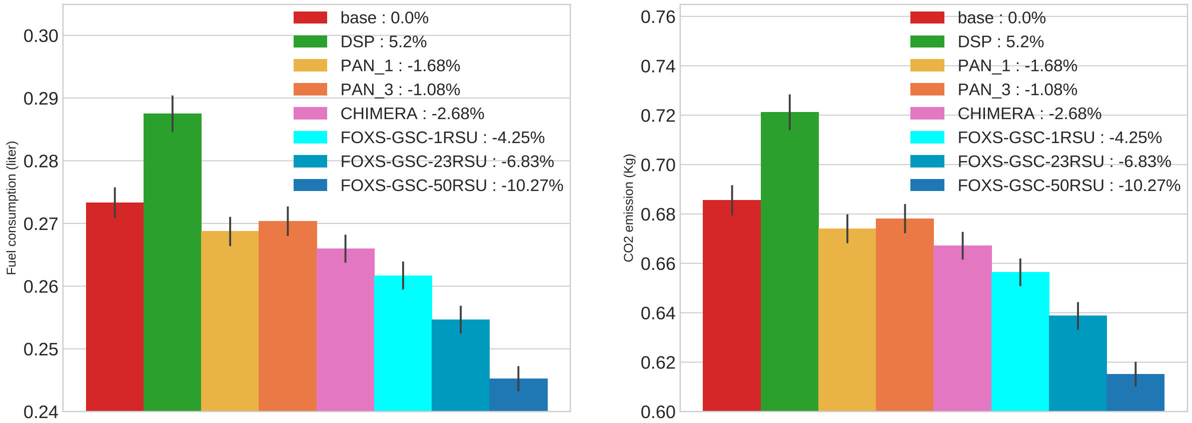
Figure 5.12 presents the percentages where the route was computed (e.g., vehicle or Cloudlet) for FOXS-GSC varying the scenario coverage in 1, 23, and 50 RSUs. Using 1 Cloudlet 80% of routes are computed by vehicles with their local knowledge. This number drops as the number of RSU increases, 28% for 23 RSU, and 0% full coverage. As expected, as there are many RSUs covering the scenario, fewer vehicles will compute their route, thus providing better traffic management as shown in Figures 5.10 and 5.11. However, with these results, a traffic engineer can make a trade-off between the cost of implementing Cloudlets and the effective gain in improving traffic.



(**a**) Stopped time. (**b**) Average speed.



(**c**) Traveled time. (**d**) Traveled distance.



(**e**) Fuel consumption. (**f**) CO2 emission.

Figure 5.10: Traffic Efficiency Results.

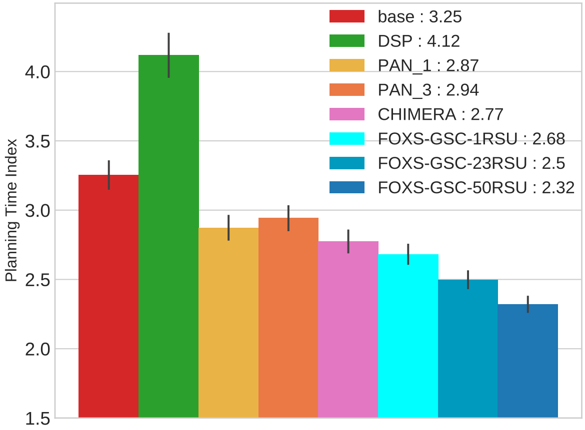


Figure 5.11: Planning Time Index (PTI).

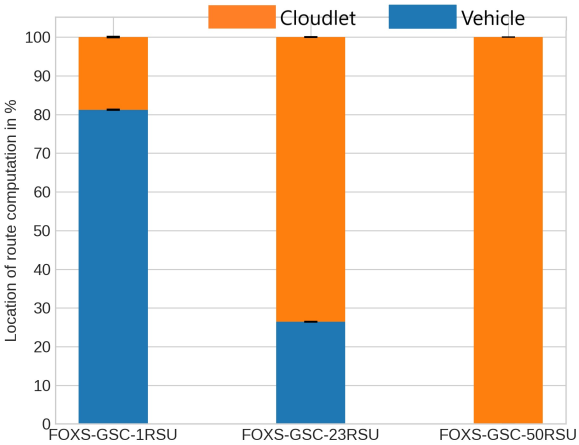


Figure 5.12: Percentage of where the route was computed (Vehicle/Cloudlet).

5.7 Final Remarks

This Section presented FOXS-GSC, a distributed communication protocol. The protocol sectorizes the scenario using a hexagonal shape. Thereby, the message is carried out through a path between origin and destination using the sectors occupied by vehicles. FOXS-GSC permits the proposed ITS traffic management service to use multihop communication, thus the full map coverage is not necessary. Several experiments to evaluate the protocol with various scenario coverage and the efficiency of traffic congestion were presented in Section 5.6.3.

Chapter 6

Conclusion and Future Work

This Chapter is organized as follows. Section 6.1 presents the summary of this thesis and Section 6.1 presents the future work.

6.1 Summary of this Work

The numerous problems caused by traffic jams in large urban cities cause a significant increase in monetary and social loss for people, companies, and the government [34, 129]. The advancement and emergence of new technologies have led to the creation of ITS. However, implementing effective ITS services to traffic management that reduces traffic jam drawbacks poses various challenges, such as the need to interact and integrate heterogeneous components and data, identify and classify the congestion, and circumvent the characteristics and limitations of VANET communication.

This manuscript presented the dissertation entitled Fog Computing-based Traffic Management Support for Intelligent Transportation Systems. In this work, we proposed a traffic service to route suggestions that use FOG computing called Fast Offset Xpath Service (FOXS). The use of the Fog computing paradigm introduces several benefits to the FOXS such as approximation of computer resources to end-users, which decreases the system response time and bandwidth usage, computational distribution, and providing the processing load balancing permitting a more scalable system.

We started our work by introducing the background (Chapter 2) and key concepts needed to understand the content of this thesis. First, we present an overview of the Fog computing characteristics and compassion with Cloud computing. Then, vehicular networks were introduced presenting the main characteristics, scenarios, type of applications, and communications standard. We also discussed Intelligent Transportation Systems (ITS) presenting its architecture, workflow, and describing the application and service requirements.

After that, we presented a vast related work in Chapter 3 pointing to the main contribution and limitations of related works. This survey also indicated the gap in the design of traffic management services considering specific features of this kind of service.

The focus of our proposed ITS service is to reduce the traffic jam and its harmful effects such as travel time, fuel consumption, and CO2 emissions. FOXS development followed the ITS workflow presented in Section 2.3.1. To address the development challenges of ITS congestion control services, various algorithms, mechanisms, and methods have been designed and enhanced. In particular, we highlight the ability of FOXS to provide a congestion control service in a distributed way with low response time and network bandwidth usage. Simulation results show that the FOXS can handle traffic jams drawbacks significantly reducing the travel time, the fuel consumption, PTI, and the CO2 emissions without compromising the network infrastructure and computational resources as presented in Section 4.6.

The main limitation of FOXS is the necessity of full scenery coverage, thus increasing the monetary cost of implantation. To solve this, the FOXS-GSC presented in Chapter 5 is developed. FOXS-GSC improves FOXS congestion control in addition to implementing a V2V communication protocol. Thus, allowing the operation of the route suggestion service even when there is no direct or indirect communication with the infrastructure (Cloudlet). These additions resulted in a service that is less dependent on infrastructure, enabling a cheaper service deployment that maintains good efficiency compared to solutions in the literature.

6.2 Limitations

The proposed solutions have some limitations, and these limitations serve as a guide for future directions. First, FOXS and FOXS-GSC solutions use a congestion classifier based on LOS. However, it has variable precision according to the characteristics of the scenario such as the average size of roads, number of crossings, number of traffic lights and time of day. Another limitation is the RSU distribution algorithm where It takes into account only the communication radius and if there are roads in its coverage. Thus, the algorithm does not provide a realistic distribution since there may be buildings, lakes, and other structures that make RSU distribution impossible.

The routing mechanisms of the proposed solutions only detect existing congestion or the verge of forming. Therefore, the vehicle route suggestion is limited as it could use a more detailed prediction based on the history and patterns of traffic behavior. Another limitation of the FOXS solution is that it needs an infrastructure that provides full coverage of the scenario. To solve this problem, the FOXS was reformulated and the FOXS-GSC multihop communication protocol was created. However, depending on the disposition of the RSU scenario, FOXS-GSC has considerable variations in the efficiency of the routing service and in the processing/network load balancing. Finally, as the FOXS-GSC multihop communication protocol was developed based on the router service requirements, the protocol has the limitation of not addressing the network partition problem.

6.3 Future Work

This work presents several possibilities for future research. For instance, as a future work, we plan to:

(i) Execute experiments considering different scenarios and the use of other communication technologies such as LTE or 5G;

(ii) Update the congestion classifier to use other traffic metrics to make the classification more accurate and independent of the characteristics of the city and roads;

(iii) Propose a service that could be implemented considering a Cloud computing interaction to increase the route suggestion efficiency using historical information about the traffic jam and driver characteristics;

(iv) Extend the service to have mobile Fog compute nodes (e.g., buses) removing the need for full coverage of the scenario. A plausible choice to be a mobile node is a bus since it has a predefined route and it is possible to schedule services according to their route;

(v) Update the proposed V2V dissemination system to allow more effective use of Cloudlets/RSU infrastructure as tunneling messages between far communication vehicles;

(vi) Create a general ITS-Fog compute framework allowing other ITS services (e.g., Video-on-demand) to receive the Fog computing advantages as FOXS does.

6.4 Award and Publications

We list the awards and all publications obtained during the doctorate below.

6.4.1 Award

Best paper awards – In 2016 IEEE Symposium on Computers and Communication

Brennand, C. A., Cunha, F., Maia, G., Cerqueira, E., Loureiro, A. A., and Villas, L. A. (2016). Fox: A traffic management system of computer-based vehicles fog. In 2016 IEEE Symposium on Computers and Communication (ISCC), pages 1–6. IEEE.

6.4.2 List of Publications

(i) Brennand, C. A., de Souza, A. M., Maia, G., Boukerche, A., Ramos, H., Loureiro, A. A., and Villas, L. A. (2015). An intelligent transportation system for detection and control of congested roads in urban centers. In 2015 IEEE Symposium on Computers and Communication (ISCC '15), pages 663–668. IEEE.

(ii) Brennand, C. A., Cunha, F., Maia, G., Cerqueira, E., Loureiro, A. A., and Villas, L. A. (2016). Fox: A traffic management system of computer-based vehicles fog.

In 2016 IEEE Symposium on Computers and Communication (ISCC), pages 1–6. IEEE. – **Best paper awards – ISCC 2016**.

(iii) Brennand, C. A., Boukerche, A., Meneguette, R., and Villas, L. A. (2017, July). A novel urban traffic management mechanism based on FOG. In 2017 IEEE Symposium on Computers and Communications (ISCC) (pp. 377-382). IEEE.

(iv) Brennand, C. A., and Pereira, G. (2019, September). Um Novo Serviço de Gerenciamento de Tráfego para ITS baseado em Computação em Névoa. In Anais do III Workshop de Computação Urbana (pp. 15-28). SBC.

(v) Brennand, C. A., Geraldo Filho, P. R., Maia, G., Cunha, F., Guidoni, D. L., and Villas, L. A. (2019). Towards a fog-enabled intelligent transportation system to reduce traffic jam. Sensors, 19(18), 3916.

(vi) Brennand, C. A., Meneguette, R., Geraldo Filho, P. R., Guidoni, D. L., and Villas, L. A. (2020, in the submission process). FOXS-GSC - Fast Offset Xpath Service with HexagonS Communication.

In addition, this research direct contributed to the following work:

(i) De Souza, A. M., Brennand, C. A., Yokoyama, R. S., Donato, E. A., Madeira, E. R., and Villas, L. A. (2017). Traffic management systems: A classification, review, challenges, and future perspectives. International Journal of Distributed Sensor Networks, 13(4), 1550147716683612.

Bibliography

[1] IEEE Std 802.15.7-2011. Ieee standard for local and metropolitan area networks-part 15.7: Short-range wireless optical communication using visible light. page 1 –309, 2011.

[2] Yuan Ai, Mugen Peng, and Kecheng Zhang. Edge computing technologies for internet of things: a primer. Digital Communications and Networks, 4(2):77–86, 2018.

[3] Ademar T Akabane, Roger Immich, Luiz F Bittencourt, Edmundo RM Madeira, and Leandro A Villas. Towards a distributed and infrastructure-less vehicular traffic management system. Computer Communications, 2020.

[4] Ademar T Akabane, Richard W Pazzi, Edmundo RM Madeira, and Leandro A Villas. Carro: A context-awareness protocol for data dissemination in urban and highway scenarios. In 2016 8th IEEE Latin-American Conference on Communications (LATINCOM), pages 1–6. IEEE, 2016.

[5] Saif Al-Sultan, Moath M Al-Doori, Ali H Al-Bayatti, and Hussien Zedan. A comprehensive survey on vehicular ad hoc network. Journal of network and computer applications, 37:380–392, 2014.

[6] Giuseppe Araniti, Claudia Campolo, Massimo Condoluci, Antonio Iera, and Antonella Molinaro. Lte for vehicular networking: a survey. IEEE Communications Magazine, 51(5):148–157, 2013.

[7] Guilherme B. Araujo, Fatima de L.P. Duarte-Figueiredo, Anna I.J. Tostes, and Antonio A.F. Loureiro. A protocol for identification and minimization of traffic congestion in vehicular networks. In Computer Networks and Distributed Systems (SBRC), 2014 Brazilian Symposium on, pages 103–112, May 2014.

[8] Muhammad Rizwan Arshad, Shahid Mehmood, Salman Afsar, Muhammad Azam Zia, and Umar Farooq. An evaluation study of wimax and wifi on vehicular adhoc networks (vanets). International Journal of Scientific & Engineering Research, 3(12):1, 2012.

[9] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. Computer networks, 54(15):2787–2805, 2010.

[10] Fan Bai and Bhaskar Krishnamachari. Exploiting the wisdom of the crowd: localized, distributed information-centric vanets [topics in automotive networking]. IEEE Communications Magazine, 48(5):138–146, 2010.

[11] Philip Ball. How honeycombs can build themselves. Nature, 10, 2013.

[12] Carolina Tripp Barba, Miguel Angel Mateos, Pablo Reganas Soto, Ahmad Mohamad Mezher, and Mónica Aguilar Igartua. Smart city for vanets using warning messages, traffic statistics and intelligent traffic lights. In Intelligent Vehicles Symposium (IV), 2012 IEEE, pages 902–907. IEEE, 2012.

[13] Javier Barrachina, Piedad Garrido, Manuel Fogue, Francisco J Martinez, Juan-Carlos Cano, Carlos T Calafate, and Pietro Manzoni. Road side unit deployment: A density-based approach. Intelligent Transportation Systems Magazine, IEEE, 5(3):30–39, 2013.

[14] Ramon Bauza and Javier Gozálvez. Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications. Journal of Network and Computer Applications, 36(5):1295–1307, 2013.

[15] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. SUMO - Simulation of Urban MObility: An Overview. In International Conference on Advances in System Simulation (SIMUL '11), pages 63–68, 2011.

[16] Mohamed Ben Brahim, Wassim Drira, and Fethi Filali. Roadside units placement within city-scaled area in vehicular ad-hoc networks. In Connected Vehicles and Expo (ICCVE), 2014 International Conference on, pages 1010–1016. IEEE, 2014.

[17] Salim Bitam and Abdelhamid Mellouk. Its-cloud: Cloud computing for intelligent transportation system. In Global Communications Conference (GLOBECOM), 2012 IEEE, pages 2054–2059. IEEE, 2012.

[18] Transportation Research Board. HCM 2010 - Highway capacity manual. National Research Council, 2010.

[19] Flavio Bonomi. Connected vehicles, the internet of things, and fog computing. In The Eighth ACM International Workshop on Vehicular Inter-Networking (VANET), Las Vegas, USA, pages 13–15, 2011.

[20] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing, pages 13–16. ACM, 2012.

[21] Celso ARL Brennand, Azzedine Boukerche, Rodolfo Meneguette, and Leandro A Villas. A novel urban traffic management mechanism based on fog. In Computers and Communications (ISCC), 2017 IEEE Symposium on, pages 377–382. IEEE, 2017.

[22] Celso ARL Brennand, Azzedine Boukerche, Rodolfo Meneguette, and Leandro A Villas. A novel urban traffic management mechanism based on fog. In Computers and Communications (ISCC), 2017 IEEE Symposium on, pages 377–382. IEEE, 2017.

[23] Celso ARL Brennand, Felipe Cunha, Guilherme Maia, Eduardo Cerqueira, Antonio AF Loureiro, and Leandro A Villas. Fox: A traffic management system of computer-based vehicles fog. In 2016 IEEE Symposium on Computers and Communication (ISCC), pages 1–6. IEEE, 2016.

[24] Celso ARL Brennand, Felipe Cunha, Guilherme Maia, Eduardo Cerqueira, Antonio AF Loureiro, and Leandro A Villas. Fox: A traffic management system of computer-based vehicles fog. In 2016 IEEE Symposium on Computers and Communication (ISCC), pages 1–6. IEEE, 2016.

[25] Celso ARL Brennand, Allan M de Souza, Guilherme Maia, Azzedine Boukerche, Heitor Ramos, Antonio AF Loureiro, and Leandro A Villas. An intelligent transportation system for detection and control of congested roads in urban centers. In Computers and Communication (ISCC), 20th IEEE Symposium on, pages 476–481. IEEE, 2015.

[26] Celso ARL Brennand, P Rocha Geraldo Filho, Guilherme Maia, Felipe Cunha, Daniel L Guidoni, and Leandro A Villas. Towards a fog-enabled intelligent transportation system to reduce traffic jam. Sensors, 19(18):3916, 2019.

[27] Jordi Calabuig, Jose F Monserrat, David Gozalvez, and Oliver Klemp. Safety on the roads: Lte alternatives for sending its messages. IEEE Vehicular Technology Magazine, 9(4):61–70, 2014.

[28] D. B. Carr, R. J. Littlefield, W. L. Nicholson, and J. S. Littlefield. Scatterplot matrix techniques for large n. Journal of the American Statistical Association, 82(398):424–436, 1987.

[29] NING Chen and XING Xu. Information-fusion method for urban traffic flow based on evidence theory combining with fuzzy rough set. Journal of Theoretical & Applied Information Technology, 49(2):560–566, 2013.

[30] Jeonghee Chi, Yeongwon Jo, Hyunsun Park, Taehyeon Hwang, and Soyoung Park. An effective rsu allocation strategy for maximizing vehicular network connectivity. Int. J, 6(4):259–270, 2013.

[31] Jeonghee Chi, Yeongwon Jo, Hyunsun Park, and Soyoung Park. Intersection-priority based optimal rsu allocation for vanet. In Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on, pages 350–355. IEEE, 2013.

[32] RACC Automobile Club. Cvis: Utility, usability and user acceptance. DEL DEPN 4.1a User Survey V1.0, 2008. Date: 15/01/2008.

[33] CAMP Vehicle Safety Communications Consortium et al. Vehicle safety communications project: task 3 final report: identify intelligent vehicle safety applications enabled by dsrc. National Highway Traffic Safety Administration, US Department of Transportation, Washington DC, 2005.

[34] G Cookson and B Pishue. Inrix global traffic scorecard. inrix research (2019).

[35] Graham Cookson and Bob Pishue. Inrix global traffic scorecard, 2017.

[36] Felipe D Cunha, Guilherme G Maia, Aline C Viana, Raquel A Mini, Leandro A Villas, and Antonio A Loureiro. Socially inspired data dissemination for vehicular ad hoc networks. In Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, pages 81–85. ACM, 2014.

[37] Joahannes BD da Costa, Allan M de Souza, Denis Rosário, Eduardo Cerqueira, and Leandro A Villas. Efficient data dissemination protocol based on complex networks' metrics for urban vehicular networks. Journal of Internet Services and Applications, 10(1):15, 2019.

[38] Felipe D. da Cunha, Leandro Villas, Azzedine Boukerche, Guilherme Maia, Aline Carneiro Viana, Raquel A. F. Mini, and Antonio A. F. Loureiro. Data communication in vanets: Protocols, applications and challenges. Ad Hoc Networks, 44:90–103, 2016.

[39] Erik Dahlman, Stefan Parkvall, Johan Skold, and Per Beming. 3G evolution: HSPA and LTE for mobile broadband. Academic press, 2010.

[40] Elizabeth M Daly, Freddy Lecue, and Veli Bicer. Westland row why so slow?: fusing social media and linked data sources for understanding real-time traffic conditions. In Proceedings of the 2013 international conference on Intelligent user interfaces, pages 203–212. ACM, 2013.

[41] Roniel S de Sousa, Azzedine Boukerche, and Antonio AF Loureiro. Distrac: A distributed and low-overhead protocol for traffic congestion control using vehicular networks. In 2019 IEEE Symposium on Computers and Communications (ISCC), pages 1–6. IEEE, 2019.

[42] A. M. de Souza, R. S. Yokoyama, L. C. Botega, R. I. Meneguette, and L. A. Villas. Scorpion: A solution using cooperative rerouting to prevent congestion and improve traffic condition. In Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on, pages 497–503, Oct 2015.

[43] A. M. de Souza, R. S. Yokoyama, G. Maia, A. Loureiro, and L. Villas. Real-time path planning to prevent traffic jam through an intelligent transportation system. In 2016 IEEE Symposium on Computers and Communication (ISCC), pages 726–731, June 2016.

[44] Allan M de Souza, Roberto S Yokoyama, Guilherme Maia, Antonio Loureiro, and Leandro Villas. Real-time path planning to prevent traffic jam through an intelligent transportation system. In Computers and Communication (ISCC), 2016 IEEE Symposium on, pages 726–731. IEEE, 2016.

[45] Soufiene Djahel, Ronan Doolan, Gabriel-Miro Muntean, and John Murphy. A communications-oriented perspective on traffic management systems for smart cities: challenges and innovative approaches. Communications Surveys & Tutorials, IEEE, 17(1):125–151, 2015.

[46] R. Doolan and G.-M. Muntean. Vanet-enabled eco-friendly road characteristics-aware routing for vehicular traffic. In Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th, pages 1–5, June 2013.

[47] Ronan Doolan and Gabriel-Miro Muntean. Ecotrec—a novel vanet-based approach to reducing vehicle emissions. IEEE Transactions on Intelligent Transportation Systems, 18(3):608–620, 2017.

[48] Prabhakar D Dorge, Sanjay S Dorle, Megha B Chakole, and Dhiraj K Thote. Improvement of qos in vanet with different mobility patterns. In Radar, Communication and Computing (ICRCC), 2012 International Conference on, pages 206–209. IEEE, 2012.

[49] Hesham El-Sayed and Moumena Chaqfeh. Exploiting mobile edge computing for enhancing vehicular applications in smart cities. Sensors, 19(5):1073, 2019.

[50] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. ACM computing surveys (CSUR), 35(2):114–131, 2003.

[51] Andreas Festag. Cooperative intelligent transport systems standards in europe. Communications Magazine, IEEE, 52(12):166–172, 2014.

[52] Ryan N Fries, Mostafa Reisi Gahrooei, Mashrur Chowdhury, and Alison J Conway. Meeting privacy challenges while advancing intelligent transportation systems. Transportation Research Part C: Emerging Technologies, 25:34–45, 2012.

[53] Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. IEEE Communications Magazine, 49(11):32–39, 2011.

[54] Mevlut Turker Garip, Mehmet Emre Gursoy, Peter Reiher, and Mario Gerla. Scalable reactive vehicle-to-vehicle congestion avoidance mechanism. In 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), pages 943–948. IEEE, 2015.

[55] Mario Gerla. Vehicular cloud computing. In Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean, pages 152–155. IEEE, 2012.

[56] Mario Gerla, Eun-Kyu Lee, Giovanni Pau, and Uichin Lee. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In Internet of Things (WF-IoT), 2014 IEEE World Forum on, pages 241–246. IEEE, 2014.

[57] Thiago S Gomides, Massilon Lourenço, Pedro Henrique de Souza, and Daniel Ludovico Guidoni. Sgtd: Sistema de gerenciamento de tráfego distribuído para redes veiculares. In Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, pages 1042–1055. SBC, 2019.

[58] Sebastian Gräfling, Petri Mähönen, and Janne Riihijärvi. Performance evaluation of ieee 1609 wave and ieee 802.11 p for vehicular communications. In 2010 Second International Conference on Ubiquitous and Future Networks (ICUFN), pages 344–348. IEEE, 2010.

[59] Michael Gross. A planet with two billion cars. Current Biology, 26(8):R307 – R310, 2016.

[60] Lin Gu, Deze Zeng, and Song Guo. Vehicular cloud computing: A survey. In 2013 IEEE Globecom Workshops (GC Wkshps), pages 403–407. IEEE, 2013.

[61] Daniel L Guidoni, Guilherme Maia, Fernanda SH Souza, Leandro A Villas, and Antonio AF Loureiro. Vehicular traffic management based on traffic engineering for vehicular ad hoc networks. IEEE Access, 2020.

[62] Fredrik Gunnarsson, Martin N Johansson, Anders Furuskar, Magnus Lundevall, Arne Simonsson, Claes Tidestav, and Mats Blomgren. Downtilted base station antennas-a simulation model proposal and impact on hspa and lte performance. In 2008 IEEE 68th Vehicular Technology Conference, pages 1–5. IEEE, 2008.

[63] Harald Haas. High-speed wireless networking using visible light. SPIE Newsroom, 2013.

[64] H. Hartenstein and K. P. Laberteaux. A tutorial survey on vehicular ad hoc networks. IEEE Communications Magazine, 46(6):164–171, 2008.

[65] Xueshi Hou, Yong Li, Min Chen, Di Wu, Depeng Jin, and Sheng Chen. Vehicular fog computing: A viewpoint of vehicles as the infrastructures. IEEE Transactions on Vehicular Technology, 65(6):3860–3873, 2016.

[66] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. Mqtt-s—a publish/subscribe protocol for wireless sensor networks. In 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08), pages 791–798. IEEE, 2008.

[67] Rasheed Hussain, Junggab Son, Hasoo Eun, Sangjin Kim, and Heekuck Oh. Rethinking vehicular communications: Merging vanet with cloud computing. In Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, pages 606–609. IEEE, 2012.

[68] Jakub Jakubiak and Yevgeni Koucheryavy. State of the art and research challenges for vanets. In 2008 5th IEEE Consumer Communications and Networking Conference, pages 912–916. IEEE, 2008.

[69] Jaehoon Jeong, Hohyeon Jeong, Eunseok Lee, Tae Oh, and David HC Du. Saint: Self-adaptive interactive navigation tool for cloud-based vehicular traffic optimization. IEEE Transactions on Vehicular Technology, 65(6):4053–4067, 2016.

[70] Kang Kai, Wang Cong, and Luo Tao. Fog computing for vehicular ad-hoc networks: paradigms, scenarios, and issues. the journal of China Universities of Posts and Telecommunications, 23(2):56–96, 2016.

[71] Georgios Karagiannis, Onur Altintas, Eylem Ekici, Geert Heijenk, Boangoat Jarupan, Kenneth Lin, and Timothy Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. Communications Surveys & Tutorials, IEEE, 13(4):584–616, 2011.

[72] John B Kenney. Dedicated short-range communications (dsrc) standards in the united states. Proceedings of the IEEE, 99(7):1162–1182, 2011.

[73] CHIN Kian Keong and Grace ONG. Smart Mobility 2030 – ITS Strategic Plan for Singapore. Land Transport Authority of Singapore, 2015.

[74] Diyar Khairi and Amine Berqia. Li-fi the future of vehicular ad hoc networks. Transactions on Networks and Communications, 3(3):31, 2015.

[75] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, et al. Optimization by simmulated annealing. science, 220(4598):671–680, 1983.

[76] Zhufang Kuang, Linfeng Li, Jie Gao, Lian Zhao, and Anfeng Liu. Partial offloading scheduling and power allocation for mobile edge computing systems. IEEE Internet of Things Journal, 2019.

[77] Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. A survey of mobile phone sensing. Communications Magazine, IEEE, 48(9):140–150, 2010.

[78] Freddy Lécué, Anika Schumann, and Marco Luca Sbodio. Applying semantic web technologies for diagnosing road traffic congestions. In The Semantic Web–ISWC 2012, pages 114–130. Springer, 2012.

[79] Uichin Lee and Mario Gerla. A survey of urban vehicular sensing platforms. Computer Networks, 54(4):527–544, 2010.

[80] Ilias Leontiadis, Paolo Costa, and Cecilia Mascolo. A hybrid approach for content-based publish/subscribe in vehicular networks. Pervasive and Mobile Computing, 5(6):697–713, 2009.

[81] Fan Li and Yu Wang. Routing in vehicular ad hoc networks: A survey. Vehicular Technology Magazine, IEEE, 2(2):12–22, June 2007.

[82] Teng Li, Min Xia, Jiahong Chen, Shujun Gao, and Clarence De Silva. A hexagonal grid-based sampling planner for aquatic environmental monitoring using unmanned surface vehicles. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pages 3683–3688. IEEE, 2017.

[83] Xin Li, Nikolaos Bamiedakis, Jinlong Wei, Jonathan JD McKendry, Enyuan Xie, Ricardo Ferreira, Erdan Gu, Martin D Dawson, Richard V Penty, and Ian H White. μled-based single-wavelength bi-directional pof link with 10 gb/s aggregate data rate. Journal of Lightwave Technology, 33(17):3571–3576, 2015.

[84] Zilu Liang and Yasushi Wakahara. Real-time urban traffic amount prediction models for dynamic route guidance systems. EURASIP Journal on Wireless Communications and Networking, 2014(1):1–13, 2014.

[85] Thomas Liebig, Nico Piatkowski, Christian Bockermann, and Katharina Morik. Dynamic route planning with real-time traffic predictions. Information Systems, 64:258–265, 2017.

[86] Jie Lin, Wei Yu, Xinyu Yang, Qingyu Yang, Xinwen Fu, and Wei Zhao. A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems. IEEE Transactions on Vehicular Technology, 66(3):2551–2566, 2017.

[87] Yen-Wen Lin, Jie-Min Shen, and Hao-Jun Weng. Cloud-assisted gateway discovery for vehicular ad hoc networks. In Information Science and Service Science (NISS), 2011 5th International Conference on New Trends in, volume 2, pages 237–240. IEEE, 2011.

[88] Chih Wei Ling, Anwitaman Datta, and Jun Xu. A case for distributed multilevel storage infrastructure for visual surveillance in intelligent transportation networks. IEEE Internet Computing, 22(1):42–51, 2018.

[89] Christian Lottermann, Mladen Botsov, Peter Fertl, Robert Müllner, Giuseppe Araniti, Claudia Campolo, Massimo Condoluci, Antonio Iera, and Antonella Molinaro. Lte for vehicular communications. In Vehicular ad hoc Networks, pages 457–501. Springer, 2015.

[90] Massilon Lourenço, Thiago S Gomides, Pedro Henrique de Souza, Cristiano Maciel da Silva, and Daniel Ludovico Guidoni. Uma solução baseada em infraestrutura auxiliar para o problema de gerenciamento de tráfego em vanets. In Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, pages 293–306. SBC, 2019.

[91] Massilon Lourenço, Fernanda SH Souza, Cristiano M Silva, Rodolfo I Meneguette, and Daniel L Guidoni. A hybrid v2i and v2v approach for urban traffic management in vehicular networks. In 2019 IEEE Latin-American Conference on Communications (LATINCOM), pages 1–6. IEEE, 2019.

[92] Tom H Luan, Longxiang Gao, Zhi Li, Yang Xiang, and Limin Sun. Fog computing: Focusing on mobile users at the edge. arXiv preprint arXiv:1502.01815, 2015.

[93] Rodolfo I. Meneguette, Geraldo P. R. Filho, Daniel L. Guidoni, Gustavo Pessin, Leandro A. Villas, and Jó Ueyama. Increasing intelligence in inter-vehicle communications to reduce traffic congestions: Experiments in urban and highway environments. PLOS ONE, 11(8):1–25, 08 2016.

[94] Rodolfo I Meneguette, Guilherme Maia, Edmundo RM Madeira, Antonio AF Loureiro, and Leandro A Villas. Autonomic data dissemination in highway vehicular ad hoc networks with diverse traffic conditions. In 2014 IEEE Symposium on Computers and Communications (ISCC), pages 1–6. IEEE, 2014.

[95] Farouk Mezghani, Riadh Dhaou, Michele Nogueira, and André-Luc Beylot. Content dissemination in vehicular social networks: taxonomy and user satisfaction. Communications Magazine, IEEE, 52(12):34–40, 2014.

[96] Farouk Mezghani, Riadh Dhaou, Michele Nogueira, and André-Luc Beylot. Offloading cellular networks through v2v communications—how to select the seed-vehicles? In 2016 IEEE International Conference on Communications (ICC), pages 1–6. IEEE, 2016.

[97] Zeeshan Hameed Mir and Fethi Filali. Lte and ieee 802.11 p for vehicular networking: a performance evaluation. EURASIP Journal on Wireless Communications and Networking, 2014(1):1, 2014.

[98] MiXiM. Mixim. last visited in September, 2013, `http://mixim.sourceforge.net/`.

[99] Ikbal Chammakhi Msadaa, Pasquale Cataldi, and Fethi Filali. A comparative study between 802.11 p and mobile wimax-based v2i communication networks. In 2010 Fourth International Conference on Next Generation Mobile Applications, Services and Technologies, pages 186–191. IEEE, 2010.

[100] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris. Smarter cities and their innovation challenges. Computer, 44(6):32–39, June 2011.

[101] Office of the Assistant Secretary for Research and U.S. Department of Transportation (US DOT) Technology (OST-R). Intelligent transport system, joint program office. last visited in May, 2019, `http://www.its.dot.gov/`.

[102] Stephan Olariu and Michele C Weigle. Vehicular networks: from theory to practice. Chapman and Hall/CRC, 2009.

[103] OMNET++. Omnet++ network simulation framework. last visited in November, 2015, `http://omnetpp.org/`.

[104] J. Pan, I. Sandu Popa, and C. Borcea. Divert: A distributed vehicular traffic rerouting system for congestion avoidance. IEEE Transactions on Mobile Computing, PP(99):1–1, 2016.

[105] Juan Pan, Iulian Sandu Popa, Karine Zeitouni, and Cristian Borcea. Proactive vehicular traffic rerouting for lower travel time. IEEE Transactions on vehicular technology, 62(8):3551–3568, 2013.

[106] S. Panichpapiboon and W. Pattara-Atikom. A review of information dissemination protocols for vehicular ad hoc networks. Communications Surveys Tutorials, IEEE, 14(3):784–798, Third 2012.

[107] Panos Papadimitratos, Arnaud De La Fortelle, Knut Evenssen, Roberto Brignolo, and Stefano Cosenza. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. IEEE Communications Magazine, 47(11):84–95, 2009.

[108] Bryan Parno and Adrian Perrig. Challenges in securing vehicular networks. In Workshop on hot topics in networks (HotNets-IV), pages 1–6, 2005.

[109] Yi Qian and Nader Moayeri. Design of secure and application-oriented vanets. In Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE, pages 2794–2799. IEEE, 2008.

[110] Kashif Naseer Qureshi and Abdul Hanan Abdullah. A survey on intelligent transportation systems. Middle-East Journal of Scientific Research, 15(5):629–642, 2013.

[111] Cisco Technology Radar. Tracking technology trends that will change the future of the industry. fostering innovation. last visited in August, 2016, `https://techradar.cisco.com/technology/fog-computing/`.

[112] Ram Ramanathan and Jason Redi. A brief overview of ad hoc networks: challenges and directions. IEEE communications Magazine, 40(5):20–22, 2002.

[113] Maxim Raya and Jean-Pierre Hubaux. Securing vehicular ad hoc networks. Journal of Computer Security, 15(1):39–68, 2007.

[114] H Rehborn, M Koller, and BS Kerner. Traffic data fusion of vehicle data to detect spatiotemporal congested patterns. In 19th ITS World Congress, 2012.

[115] Gianluca Rizzo, Maria Rita Palattella, Torsten Braun, and Thomas Engel. Content and context aware strategies for qos support in vanets. In 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), pages 717–723. IEEE, 2016.

[116] F.J. Ros, P.M. Ruiz, and I. Stojmenovic. Acknowledgment-based broadcast protocol for reliable and efficient data dissemination in vehicular ad hoc networks. Mobile Computing, IEEE Transactions on, 11(1):33–46, 2012.

[117] Mohammad A Salahuddin, Ala Al-Fuqaha, Mohsen Guizani, and Soumaya Cherkaoui. Rsu cloud and its resource management in support of enhanced vehicular applications. In 2014 IEEE Globecom Workshops (GC Wkshps), pages 127–132. IEEE, 2014.

[118] Pierpaolo Salvo, Francesca Cuomo, Andrea Baiocchi, and Andrea Bragagnini. Road side unit coverage extension for data dissemination in vanets. In Wireless On-demand Network Systems and Services (WONS), 2012 9th Annual Conference on, pages 47–50. IEEE, 2012.

[119] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. IEEE pervasive Computing, 8(4), 2009.

[120] Basic Transmission Scheme. Lte: the evolution of mobile broadband. IEEE Communications magazine, 45, 2009.

[121] Ramon S Schwartz, Rafael RR Barbosa, Nirvana Meratnia, Geert Heijenk, and Hans Scholten. A directional data dissemination protocol for vehicular environments. Computer Communications, 34(17):2057–2071, 2011.

[122] Riccardo Scopigno, Alessia Autolitano, Tankut Acarman, Çağdaş Yaman, and Suat Topsu. The potential benefits of on-board li-fi for the cooperation among vehicles. In 2015 17th International Conference on Transparent Optical Networks (ICTON), pages 1–6. IEEE, 2015.

[123] Baraa T Sharef, Raed A Alsaqour, and Mahamod Ismail. Vehicular communication ad hoc routing protocols: A survey. Journal of Network and Computer Applications, 40:363–396, 2014.

[124] T.H. Silva, P.O.S. Vaz de Melo, J.M. Almeida, and A.A.F. Loureiro. Large-scale study of city dynamics and urban social behavior using participatory sensing. Wireless Communications, IEEE, 21(1):42–51, February 2014.

[125] C. Sommer, R. German, and F. Dressler. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. Mobile Computing, IEEE Transactions on, 10(1):3–15, Jan 2011.

[126] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. IEEE Transactions on Mobile Computing, 10(1):3–15, 2011.

[127] Ivan Stojmenovic and Sheng Wen. The fog computing paradigm: Scenarios and security issues. In Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on, pages 1–8. IEEE, 2014.

[128] O.K. Tonguz, N. Wisitpongphan, and Fan Bai. Dv-cast: A distributed vehicular broadcast protocol for vehicular ad hoc networks. Wireless Communications, IEEE, 17(2):47–57, 2010.

[129] IBM Smart Traffic. Epa - united state environmental protection agency. `http://www.ibm.com/smarterplanet/us/en/traffic_congestion/ideas`, 2015.

[130] Martin Treiber, Arne Kesting, and R Eddie Wilson. Reconstructing the traffic state by fusion of heterogeneous data. Computer-Aided Civil and Infrastructure Engineering, 26(6):408–419, 2011.

[131] Sandesh Uppoor and Marco Fiore. Large-scale urban vehicular mobility for networking research. In Vehicular Networking Conference (VNC), 2011 IEEE, pages 62–69. IEEE, 2011.

[132] Sandesh Uppoor, Oscar Trullols-Cruces, Marco Fiore, and Jose M Barcelo-Ordinas. Generation and analysis of a large-scale urban vehicular mobility dataset. IEEE Transactions on Mobile Computing, 13(5):1061–1075, 2014.

[133] L. A. Villas, A. Boukerche, R. B. Araujo, A. A. F. Loureiro, and J. Ueyama. Network partition-aware geographical data dissemination. In 2013 IEEE International Conference on Communications (ICC), pages 1439–1443, June 2013.

[134] Leandro A. Villas, Heitor S. Ramos, Azzedine Boukerche, Daniel L. Guidoni, Regina B. Araujo, and Antonio A.F. Loureiro. An efficient and robust data dissemination protocol for vehicular ad hoc networks. In Proceedings of the 9th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN '12, pages 39–46, New York, NY, USA, 2012. ACM.

[135] Leandro Aparecido Villas, Azzedine Boukerche, Guilherme Maia, Richard Werner Pazzi, and Antonio A.F. Loureiro. Drive: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks. Computer Networks, 75, Part A(0):381 – 394, 2014.

[136] Alexey Vinel. 3gpp lte versus ieee 802.11 p/wave: which technology is able to support cooperative vehicular safety applications? IEEE Wireless Communications Letters, 1(2):125–128, 2012.

[137] W. Viriyasitavat, O.K. Tonguz, and Fan Bai. UV-CAST: an urban vehicular broadcast protocol. IEEE Communications Magazine, 49(11):116–124, 2011.

[138] Shen Wang, Soufiene Djahel, Zonghua Zhang, and Jennifer McManis. Next road rerouting: A multiagent system for mitigating unexpected urban traffic congestion. IEEE Transactions on Intelligent Transportation Systems, 17(10):2888–2899, 2016.

[139] Tian Wang, Jiyuan Zhou, Anfeng Liu, Md Zakirul Alam Bhuiyan, Guojun Wang, and Weijia Jia. Fog-based computing and storage offloading for data synchronization in iot. IEEE Internet of Things Journal, 2018.

[140] Xiaojie Wang, Zhaolong Ning, Xiping Hu, Lei Wang, Bin Hu, Jun Cheng, and Victor CM Leung. Optimizing content dissemination for real-time traffic management in large-scale internet of vehicle systems. IEEE Transactions on Vehicular Technology, 68(2):1093–1105, 2018.

[141] Axel Wegener, Michał Piórkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean-Pierre Hubaux. Traci: an interface for coupling road traffic and network simulators. In Proceedings of the 11th communications and networking simulation symposium, pages 155–163. ACM, 2008.

[142] Md Whaiduzzaman, Mehdi Sookhak, Abdullah Gani, and Rajkumar Buyya. A survey on vehicular cloud computing. Journal of Network and Computer Applications, 40:325–344, 2014.

[143] Elias Yaacoub, Fethi Filali, and Adnan Abu-Dayya. Svc video streaming over cooperative lte/802.11 p vehicle-to-infrastructure communications. In Computer and Information Technology (WCCIT), 2013 World Congress on, pages 1–5. IEEE, 2013.

[144] Gongjun Yan, Ding Wen, Stephan Olariu, and Michele C Weigle. Security challenges in vehicular cloud computing. Intelligent Transportation Systems, IEEE Transactions on, 14(1):284–294, 2013.

[145] Shanhe Yi, Zijiang Hao, Zhengrui Qin, and Qun Li. Fog computing: Platform and applications. In 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), pages 73–78. IEEE, 2015.

[146] Maram Bani Younes and Azzedine Boukerche. A performance evaluation of an efficient traffic congestion detection protocol (ecode) for intelligent transportation systems. Ad Hoc Networks, 24:317–336, 2015.

[147] Maram Bani Younes, Azzedine Boukerche, and Graciela Rom'an-Alonso. An intelligent path recommendation protocol (icod) for vanets. Computer Networks, 64:225–242, 2014.

[148] M.B. Younes and A Boukerche. Efficient traffic congestion detection protocol for next generation vanets. In Communications (ICC), 2013 IEEE International Conference on, pages 3764–3768, June 2013.

[149] Rong Yu, Yan Zhang, Stein Gjessing, Wenlong Xia, and Kun Yang. Toward cloud-based vehicular networks with efficient resource management. Network, IEEE, 27(5):48–55, 2013.