UNIVERSITÀ
DEGLI STUDI
FIRENZE

Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione
Curriculum: Telecomunicazioni e Sistemi Telematici

Mobile Computing and Networking Architectures for the Internet of Vehicles

Candidate
Alessio Bonadio

Supervisors
Prof. Romano Fantacci

Prof. Francesco Chiti

PhD Coordinator
Prof. Fabio Schoen

ciclo xxxiii, 2017-2020

Università degli Studi di Firenze, Dipartimento di Ingegneria
dell'Informazione (DINFO).

"Did it for the memes"
— Elon Musk

Acknowledgments

I would like to express my sincere gratitude to my supervisors, Prof. Romano Fantacci and Prof. Francesco Chiti. Their patience, motivation, and continuous support guided me during the three years of my Ph.D. study.

I thank all my colleagues of the Data Communication and Network Systems (DaCoNetS) Lab with whom I shared this period of time. In particular my thanks go to Benedetta and Giulio for the stimulating discussions and for all the fun we have had in the last three years.

I would like to thank also Prof. Carlo Fischione for his kind support and hospitality during my stay at KTH Royal Institute of Technology, where I carried out part of my research.

Last but not the least, I would like to thank my family and my friends for supporting me throughout these years.

Abstract

In recent years, the transport industry has seen rapid technological changes in the application of intelligent driver assistance systems. This trend has as its objective the realization of cars that achieve a high level of automation on the road. At the same time, interest in intelligent transportation systems has spread, intended as advanced platforms able to provide innovative and traffic management services. Cars have gone from simple vehicles to actual smart devices equipped with hundreds of sensors and actuators. Consequently, the amount of data acquired by each car must necessarily be shared and processed. The perspective is the emergence of an innovative range of services. From a technological point of view, collaborative interactions between vehicles will lead to a scenario where it is necessary to integrate vehicles within a specific telecommunications network, or even to give rise to a specific ad hoc one, capable of both transferring and computing a large amount of data, many of which with strict delay constraints. The Internet of Mobile Things is poised to meet these demands. However, it poses two main challenges, the first with respect to the amount of data exchanged, the second with respect to the need to perform heavy computations on such data. This thesis starts by considering the problem of data dissemination, investigating several network schemes. Besides, to ensure the consistency of the data collected, a distributed consensus sensing application is designed. Then, a mobile Edge computing system is modeled. This paradigm provides computational capabilities at the edge of the network and is able to fulfill the requirements of the Internet of Mobile Things. The model is used to derive the minimum number of processors to be allocated to obtain a given requests dropping probability. Finally, mobile Edge computing and Cloud computing systems are compared. Two analytical models are developed and validated, considering the total service time as a key metric. The comparison gives some insight on how these systems should be designed to handle a given load.

The main contributions of this dissertation can be summarized as follows:

- An overview of the state-of-the-art in vehicular networking, also regarding the enabling network technologies.
- Design of an integrated vehicular network architecture. In particular, a specific co-designed approach involving Application and Networks Layers is proposed. We resort to blockchain principle to develop a distributed consensus sensing application. Performance analysis has been conducted over realistic scenarios.
- Performance evaluation of a mobile Edge computing system providing computational capabilities to users within a limited area. In particular, a Markov multi-server queuing system model with requests reneging is proposed and validated by comparing the obtained analytical predictions with numerical results derived by simulations carried out under realistic operating conditions.
- Comparison of mobile Edge computing and Cloud computing systems, considering the total service time, including also the delays due to communications. Specifically, we design two analytical models and propose a statistical approach to understand how the system load and the amount of resources allocated influence offloading policies.

Contents

List of Acronyms

AR	augmented reality
BC	blockchain
BF	blind flooding
BS	base station
CC	Cloud computing
CDF	cumulative distribution function
CS	consensus sensing
DTN	delay-tolerant networking
FC	Fog controller
FIFO	first in, first out
ITS	intelligent transportation system
IoMT	Internet of Mobile Things
IoT	Internet of Things
IoV	Internet of Vechicles
MANET	mobile ad hoc network
MEC	mobile Edge computing
MED	mobile Edge device
NC	network coding
P2P	peer-to-peer
PDF	probability density function
PF	probability-based flooding
PoET	proof of elapsed time

PoW	proof of work
QoS	quality of service
SaaS	software as a service
TF	TTL-based flooding
URLLC	Ultra-Reliable Low-Latency Communications
V2I	vehicle-to-infrastructure
V2V	vehicle-to-vehicle
V2X	vehicle-to-everything
VANET	vehicular ad hoc network
VFC	vehicular Fog controller
VFD	vehicular Fog domain
VFN	vehicular Fog node
VM	virtual machine
VR	virtual reality
VSN	vehicular social networks

Chapter 1

Introduction

In recent years, the transport industry has seen rapid technological changes in the application of intelligent driver assistance systems. This trend has as its objective the realization of cars that achieve a high level of automation on the road. In this sense, vehicles have been equipped with systems that allow individual acquisition of context information, however, to achieve a true intelligent system, it is necessary to share and process this information collectively. The automotive industry will therefore need to complement the telecommunications industry in order to achieve a truly autonomous vehicle system.

At the same time, interest in intelligent transportation systems (ITSs) has spread, intended as advanced platforms able to provide innovative traffic management services, allowing users to be better informed and make safer, coordinated and "smart" use of transport networks [20]. In fact, within a decade, cars have gone from simple vehicles to actual smart devices equipped with hundreds of sensors and actuators, connected to the local interface and the vehicle control unit. Thanks to recent developments in the field of artificial intelligence, autonomous vehicles are able to understand the external environment, with specific reference to the other cars involved, in real time through the combined use of different techniques, such as radar, lidar and GPS [24], and to make decisions based on this knowledge. However, to date, each vehicle remains essentially an intelligent but isolated system.

Consequently, in order to increase the potential of ITS systems, the amount of data acquired by each car must necessarily be shared and processed. The interfaces that are required between vehicles and/or with the

infrastructure converge in the new concept of vehicle-to-everything (V2X) communications. Vehicles interconnected with road infrastructure and each other must collect information about the environment and exchange this information with neighboring entities in real time [16], all this for the benefit of driving safety, quality and efficiency of transport systems.

The perspective is the emergence of an innovative range of services based on the Mobile Crowdsensing vehicular paradigm [84]. From a technological point of view, collaborative interactions between vehicles will lead to the birth of the so-called vehicular social networks (VSN). A VSN is, in fact, a network formed by vehicles that share interests, preferences or needs in a given time context or for a given proximity on the road [80]. It is clear that information about the local context is an essential element for the existence of vehicle networks. In addition, the enormous amount of data involved requires that the network itself also assists the vehicles to perform the most demanding computations. In this scenario currently being defined, it is necessary to integrate vehicles within a specific telecommunications network capable of both transferring and computing a large amount of data, many of which with strict delay constraints.

A first reference model could be identified in the so called vehicular ad hoc networks (VANETs) paradigm, that is a special kind of mobile ad hoc networks (MANETs) composed by vehicles as well as access points located at the edges of the roads [60,76,82]. VANETs respond to the need to move data quickly between vehicles. There is a second very important aspect needed to achieve an intelligent system: being able to process the large amount of data acquired. However, the computation to be performed might exceed the capabilities of the hardware on board the vehicle. In this case it is necessary to loosen the requirement of a strictly ad-hoc network, and to consider some external devices, which could be part of the network infrastructure.

Advances in the Internet of Things (IoT) have created a promising prospect to meet the demands of vehicular networking [36]. However, considering the high mobility of the vehicular context, it is more appropriate to refer to the Internet of Mobile Things (IoMT). The differences between IoT and IoMT span several areas. From this work perspective, the most important is definitely the context, *e.g.*, where the mobile device is located, which is not sufficiently taken into account by the simple IoT paradigm. Hence, when considering IoMT, mobility becomes a first class concern and one has to look at the IoMT separately from IoT [57]. IoMT poses two main chal-

lenges, the first with respect to the amount of data exchanged, the second with respect to the need to perform heavy computations on such data.

In most IoMT applications, devices are subject to very high interaction, demanding a rich scenario of communication among themselves such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) and V2X, which unifies them and makes them indistinguishable from the networking point of view. IoMT could be particularly beneficial in the fields of traffic safety and management, since vehicles could transmit traffic-related information (such as vehicle position, speed, hazards) to maintain traffic safety. As a consequence devices are involved in data gathering and dissemination, fulfilling the requirements to enhance the reactiveness to sudden context variations of real-time data [67, 69, 79]. It is clear how the IoMT paradigm is much more than the simple ad-hoc networking provided by the VANET model.

The transition to IoMT is pushed forward by the fifth generation (5G) technology, which is designed to integrate several different interfaces and make them appear as seamless. By using 5G technologies the transition to the V2X model is even more natural, and communication can actually happen with everything. 5G, in fact, is not to be thought of only as a smartphone oriented technology, the aim is to unify different contexts and create a common layer for different application scenarios, enabling "post-smartphone" services. The IoMT paradigm poses some challenges to the networking layer, particularly in terms of the constant need for connection and the large amount of data to be exchanged, combined with the extreme mobility of nodes. It is therefore necessary a joint design of the computing subsystem with the networking subsystem, so that application overlay and connectivity underlay are not mismatched, and that both application requirements and network characteristics are considered.

IoMT presents a scenario in which unbounded streams of data are arriving at the sensing devices (*i.e.*, sensors installed on a traveling vehicle or a personal device) as a high rate data stream. These data have to be processed “on the fly” to detect anomalies, operational exceptions, deliver real-time alerts, and trigger automated actions. Given the enormous amount of data involved, it might be a good idea to use the services offered by a Cloud computing (CC) system to perform this computation. However, a remote CC node can not always meet the latency constraints of the novel IoMT services. This is where the newly emerged mobile Edge computing (MEC) paradigm comes into play by providing CC capabilities at the edge of the network

in close proximity to the end users and enabling important additional capabilities, such as location awareness, in order to provide highly localized services [25].

The MEC-based computational offloading offers a large number of advantages, enabling new applications and services. However, these systems do not have an infinite amount of resources and, being co-located with network base stations (BSs), they cannot be too large or use too sophisticated cooling systems. It follows that the services offered by a MEC node are to be considered "valuable", and a technique must be provided to use them only in case of real need, *i.e.*, when a better result is guaranteed compared to resorting to classic CC. The problem is already known in the literature as dispatching [34] in a hybrid MEC-CC system, meaning that in the considered scenario both MEC and CC systems are present and available to receive tasks, but has not yet been adequately investigated. In particular, it is essential to understand what the trade-off between MEC and CC is, to ensure that the performance of the MEC system does not degrade too much. MEC system is necessary to ensure the low-latency requirements imposed by some services, depending on the system load, but is a trade-off with CC is also needed to understand when resorting to a classic CC [25].

Chapter 2

Literature review

This chapter gives a brief survey of related work. The first part of the chapter introduces the topics related to vehicular networking. The literature on MEC is presented below, together with some use cases enabled by this architecture. Finally, the last part of the chapter deals with the problem of task dispatching in mixed MEC-CC systems.

2.1 State-of-the-art of Vehicular Networking

A consolidated but still evolving field of research is represented by MANET, which differ from infrastructured ones in that they do not rely on a pre-existing infrastructure [18]. However, this differentiation is bound to diminish in favor of integration with 5G technology, which aims to merge the two types of networks.

In the large family of MANETs, much attention is paid to VANET, *i.e.*, those networks designed to support communication between vehicles (V2V) and communication between vehicles and roadside infrastructure (V2I) [35]. There are organizations, such as the Car 2 Car Communication Consortium (C2C-CC), made up of European car manufacturers, which develop standards to enable and facilitate communication between vehicles of different brands [1].

VANETs can be based on virtually any technology: WLAN, WPAN or cellular. While the actual reference standard for Internet of Vechicles (IoV) is represented by IEEE 1609/WAVE (Wireless Access in the Vehicular En-

vironment), with both V2V and V2I [78] interfaces, VANETs are expected to rely on V2X, an abstract and flexible communication mode [24]. This solution seems to be the most promising one not only to support the exponential growth of data generated by the unlimited number of future connected devices, but also to support different applications and meet new user expectations.

In recent years, applications for mobile agents are becoming increasingly complex, generating a huge amount of data and requiring additional computational power. In addition, given the diffusion of mobile IoT based architectures, a huge number of extremely constrained devices are becoming pervasive in our physical environments, bringing the vision of Intelligent Environments closer to reality [9]. Vehicles are expected to follow the IoT architecture, constituting an actual IoV, thus processing large amounts of data with stringent requirements in end-to-end latency, to achieve effective autonomous driving [6]. Computational offloading could provide several advantages to IoV, realizing a distributed processing environment [29].

Another IoT categorization is the IoMT, where the smart things can move independently and remain accessible within the network [8]. IoMT specifies the connection between moving sensors and devices instead of stationary things. Thus, IoMT encompasses the majority of IoT connected mobile things, including mobile robots and vehicles on highways. IoV is indeed a good example of IoMT. The authors in [57] identify four challenges to be addressed for the IoMT: data collection, data analytics, energy management, and security and privacy. Authors in [74] propose a middleware which allows the objects in the IoMT to move autonomously and remain remotely accessible over the Internet. The authors in [39] propose a platform as a service (PaaS) model for IoMT that is geospatially distributed, large-scale, and latency-sensitive.

The growing demand of novel pervasive and more powerful applications for mobile users, such as virtual reality (VR) and augmented reality (AR), ultra-high-definition (UHD) video streaming, image processing, face detection and recognition, and real-time interactive gaming, to name a few [53], clashes with some of the characteristics of mobile devices. Often these services also have strict end-to-end low-latency constraints [26, 81]. However, due to limited storage, computing capability, and battery lifetime, it is very challenging for a mobile device to support these computation-and-energy-consuming applications. Moreover, from 50 to 100 billion smart devices are

expected to connect to the Internet in the next future, which will stimulate ever more rapid growth of data traffic [11].

CC systems were initially proposed to support applications matching these requirements [5, 58]. However, they have some inherent drawbacks, such as the excessive latency introduced by communication with IoT devices, which makes it very difficult to meet the strict constraints of several IoT applications [65]. To face these issues, a single CC systems can be divided into smaller subsystems and moved closer to mobile devices, at the edges of the network, according to the MEC paradigm [12, 41, 56]. In addition, using the potential of 5G networks, the characteristics of MEC systems can be further improved [26]. In particular, 5G supports use cases such as Ultra-Reliable Low-Latency Communications (URLLC) and massive Machine Type Communications (mMTC) [2], so there are all the elements to support applications with stringent latency requirements at the network level and manage robust connection of lots of IoT devices preventing network overload.

The MEC paradigm, firstly proposed by the European Telecommunications Standard Institute (ETSI) in 2014 [3,41], relies on moving computation power and storage at the network edges to enable computation-intensive, real time IoT applications. In such a context, it is well known that MEC outperforms the CC alternative, mainly in terms of latency reduction and possibility to support real-time data computations [17, 21, 53], due to a closer access and lower computing facilities congestion than the Cloud, usually shared by a very high number of users. Additional advantages of MEC compared to CC, due to the proximity to end users, are a better exploitation of the context awareness, which is paramount in vehicular applications, and an enhanced privacy and security for the supported services [25, 59]. In recent years, several survey papers have been published to provide overviews of the MEC systems, such as [25, 53] or to deal with specific issues as that of identifying suitable policies to handle the tasks offloading from mobile Edge devices (MEDs) to the MEC facilities [15, 48, 70, 88]. Nowadays, MEC is commonly considered as a key emerging technology and a fundamental component of future generation networks, including the 5G case, enabling a huge number of novel opportunities to future data network operators, as well as to equipment vendors [68]. Finally, it is important to stress that as a consequence of the MEC capability of supporting applications in proximity of mobile users we have significant improvements also in terms of MEDs battery life and service quality as highlighted in [63].

Indeed, the main advantage of MEC systems is to run applications and related processing tasks in the proximity of mobile users. Therefore, network congestion is reduced, battery life is enhanced and service experience is improved. Several investigations recently overviewed MEC paradigm, pointing out that it could be represent a key technology and an important component of 5G networks [41].

2.2 Use Cases for Mobile Edge Computing

Many use cases have been defined for MEC systems, which introduced a whole series of new applications. MEC systems stands out as a promising solution for prolonging the battery of IoT devices: specifically, computation intensive tasks can be offloaded from IoT devices to MEC nodes to reduce their energy consumption [63].

MEC systems are also critical to enable powerful and computation consuming applications on devices with limited computational capabilities. In this case, it is also important to emphasize the time requirement, which, for some applications, may be very strict. Low-latency constraints are generally very difficult to meet for more limited devices. The classic solution is to offload the most intensive computations to remote CC systems. However, due to the high and variable latency to a distant CC server, it can not always be a suitable for each type of task, especially for real-time immersive applications [81].

AR mobile applications are gaining increasing momentum due to the their ability to combine computer-generated data with the physical reality. AR applications are computational-intensive and delay-sensitive. To address this problem, the most time and energy-consuming computations has to be offloaded to MEC servers [7]. An even more demanding application is VR, which is heavily limited by the transmission latency. The concept of MEC strikes a balance between communication latency and computing latency by providing high computational resources close to the users [26].

Another important use cases is IoV, which can require huge amount of data processing with low-latency requirements. Automotive systems have strict quality of service constraints in terms of ultralow-latency for V2X communications [6]. In ITS systems, the amount of data collected, as well as the computational capacity needed to process it, is skyrocketing and the connectivity available to transport information from devices to data centers

is not always adequate for the purpose. In particular, applications with stringent requirements in terms of delay or security are not best supported by the current CC paradigm. It is preferable in principle that data requiring lighter processing and more stringent delay requirements remain local, while others are transferred to the cloud. It is therefore natural to divide the data into two categories:

- data concerning small term decisions that have strict delay requirements and are processed closer to vehicles;
- data concerning medium-term decisions that are processed and stored in the cloud.

MEC, leveraging computing and data storage capabilities physically close to the devices and in highly distributed modes, is an efficient solution to meet these requirements, *e.g.*, to support the autonomous driving services [29,85]. It can indeed provide flexible vehicle coverage and is essential to enable a range of applications for VANET networks.

2.3 Task Dispatching in Mixed Mobile Edge-Cloud Systems

Task dispatching refers to the process of deciding where to offload a task for computation. In our architecture the choice is between processing it on a MEC node, closer to the devices that generated the request, or using CC services, further away from the end user. Several articles are investigating when a task should be executed on the MEC or CC in a mixed MEC-CC system.

Authors in [34] formulate the problem and propose the first online task dispatching and scheduling algorithm in MEC-CC systems, with the goal to minimize the response time, which has been defined as the interval between the task's generation in a mobile device and the time when it is finished and the result is received by the device. Tong *et al.* [75] proposed a hierarchical architecture for MEC-CC systems, dividing them according to their distance to from MEDs, so that a peak load can be offloaded to a higher tier to minimize the mean response time. Authors in [86] study the joint optimization problem of service placement and load dispatching in mobile cloud systems, defining a problem which takes into account the cost of resource usage on MEC and CC. Then they design an online solution to that problem which

aims to optimize the trade-off between access latency and resource usage, from the standpoint of service provider.

Other articles complement the task dispatching problem with scheduling on the different nodes. Meng *et al.* [54] propose an online scheduling policy that considers the scheduling of both networking bandwidth and computing resources in MEC-CC systems. In [83] authors are proposing an heuristic which jointly addressed task dispatching and scheduling. The goal of their work is to balance the request processing between MEC and CC nodes to avoid overload or wasted resources, and eventually minimize the overall response time of requests. The response time is again defined considering the transmission times.

As can be noted, the main constraint of a mixed MEC-CC system very often is the service time requirement. This emphasizes the importance of considering the low-latency constraint when performing the analysis of MEC-CC systems. In the previous literature, the consideration of system parameters was not required. However, for low latency services these parameters are fundamental, and with our applications their characterization is of paramount importance. Compared to the state of the art, our work proposes a new investigation. We analyze from a statistical point of view the fundamental properties of MEC and CC before deciding on tasks offloading, to obtain useful information for the subsequent design of task dispatching algorithms and heuristics, while the state-of-the-art analyzes tasks offloading after the basic parameters have been set. Therefore, our work is orthogonal to the state-of-the-art approach and is original in the field of task dispatching.

Chapter 3

Data Dissemination in V2V Internet of Vehicles

In this chapter an integrated system architecture is presented. It is applied to achieve a full context awareness for vehicular networks and, consequently, to react on traffic anomalous conditions. In particular, we propose to adopt a specific co-designed approach involving Application and Networks Layers. For the latter one, as no infrastructure usually exists, effective routing protocols are needed to guarantee a certain level of reliability of the information collected from individual vehicles. As a consequence, we investigated classical Epidemic Flooding based, Network Coding inspired and Chord protocols. Besides, we resort to blockchain (BC) principle to develop a distributed consensus sensing application. Performance analysis has been conducted over realistic scenarios in terms of consensus making overhead, latency and scalability, pointing out the better trade-off allowing the overlay peer-to-peer (P2P) network formation and the complete context awareness achieved by the vehicles community.[1]

To ensure consistency, it is often essential that the vehicles participating in a VANET reach agreement on the data they acquire. In this perspective, we applied the BC principle, that was originally conceived as a digital

[1]This chapter has been published as part of "An Integrated Framework for blockchain inspired Fog Communications and Computing in Internet of Vehicles" in *Journal of Ambient Intelligence and Humanized Computing*.

payment framework. It enables participants to read from and update to a common shared ledger (or BC) whose state is collectively maintained by the network in a decentralized fashion [87]. BC is updated via the consensus protocol that ensures a unambiguous ordering of transactions to guarantee the integrity and consistency of the BC across geographically distributed nodes. In addition, BC presents some advantages over existing electronic frameworks for both the sender and the beneficiary, among that: transparency, verifiability, limited exchange cost, instantaneous transactions, and network security [62].

As a consequence, a self-adaptable and efficient networking paradigm is needed [22, 45] to gracefully adapt the IoV ecosystem with respect to the operative context. Since the previous applications are typically latency-and-space sensitive, we resort to a specific Fog oriented lightweight BC keeping a local log of past transactions related to a specific context (*i.e.*, an accident occurred) that can not be adversarially modified or repudiated by any vehicles involved in.

To face these issues, we proposed an integrated approach, managing the network topology set-up and maintaining, together with the information dissemination and validation in a decentralized and autonomous way. In particular, it involves the investigation of state-of-the-art routing approaches, from classical delay-tolerant networking (DTN) oriented schemes, up to massive network coding (NC) and Chord approaches, to achieve a cooperation gain which reduces the consensus making latency at the increasing of network size and vehicles (dis)connections rate due to mobility.

3.1 Integrated Framework

The reference vehicular Fog (VF) architecture considered for our proposal is represented in Figure 3.1, where two vehicular Fog domains (VFDs) comprised of vehicular Fog nodes (VFNs) and vehicular Fog controllers (VFCs), are depicted. In addition logical (*i.e.*, related to the overlaying application) and physical (*i.e.*, specific to the underlying network) communications interfaces are represented with dashed or solid lines, respectively. A Fog controller (FC) has been also introduced to allow the interoperability among VFDs.

The VFN functional model, adopted in designing our integrated approach, is shown in Figure 3.2, that is comprised of:

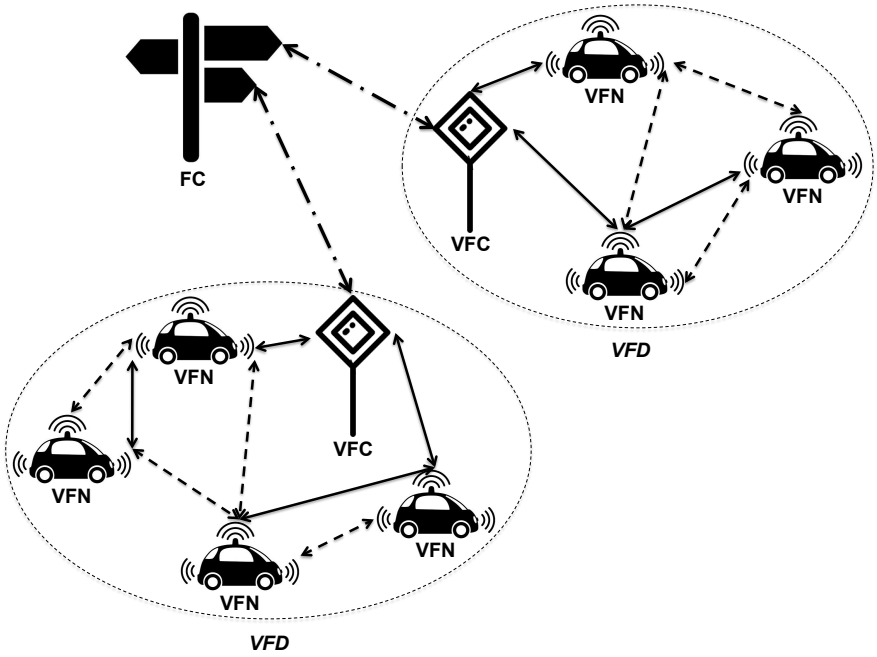

Figure 3.1: Reference Fog IoV functional architecture in terms of logical nodes and interfaces. In particular two domains are depicted comprised of Fog nodes and controllers, VFN and VFC, respectively, together with a high level controller (FC).

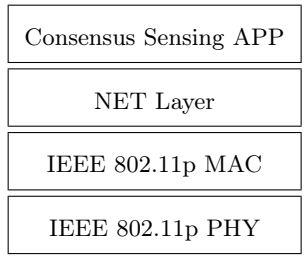

Figure 3.2: Proposed protocol architecture, involving standardized layer (IEEE 802.11p PHY and MAC) and ad hoc Networking-Application ones.

- Consensus sensing (CS) application designed according to BC technology;

- Network Layer functionalities, as defined below;
- Physical and Data Link Layers compliant with IEEE 802.11p specifications.

Transport Layer functionalities are not included in the proposed architecture as usual in VANETs; therefore, our application directly uses services provided by NET Layer.

As a reference scenario, we consider a typical traffic congestion, characterized by slowing speeds and vehicular queuing. This usually occurs in the presence of a red traffic light or even due to an accident. In this situation, it is particularly relevant that the involved vehicles quickly achieve context awareness by means of distributed data collection and fusion about the road/traffic conditions. Thus, vehicles are requested to (i) arrange the pool into a network and then (ii) start the CS application. Further, this enables a distributed decision process, relaying on the information generated by each vehicle.

3.1.1 Consensus Sensing Application

According to the reference scenario previously introduced, vehicles are expected to achieve a full context awareness by means of a distributed data gathering procedure. This requires also a local information reconciliation by applying a CS application, that has been designed according to the BC technology. As a matter of fact, this allows all the participants to read from a distributed ledger, that records all the observations from vehicles. The derived chain is updated using a protocol, which guarantees a common view of the overall information. Moreover, it assures the integrity and the consistency of the ledger and its non ambiguous ordering.

Specifically, a permissionless blockchain has been adopted, where all the the vehicles potentially involved in a correlated traffic episode take part in the consensus making process. In particular, we considered an alternative proof of work (PoW) similar to the recently proposed proof of elapsed time (PoET), which implements a lightweight (*i.e.*, not mining intensive) and time-based consensus mechanism, specifically designed to reduce complexity and to improve reactiveness [27]. According to PoET, each network participant (*i.e.*, miner) runs a trustworthy piece of code that idles for a randomly determined interval of time. The node that firstly becomes awake is the leader of the consensus round and receives a reward, which consists in a priority to be spent in successive rounds of distributed scheduling process. It

is worth noticing that this scheme is particularly suited for an opportunistic networking model (*i.e.*, IoV), where hosts are usually resource constrained and prone to fault [73].

According to our approach, PoW has been modelled with a uniform random distribution. We also set the block size B, representing the amount of information to be validated as a whole. In particular, we perform a validation via the joint consensus of at least half of the network nodes, so $B = N/2$, where N is the number of nodes.

Finally, we design two message types:

- `ObservationMessage` (OM), which carries the information collected by a vehicle;
- `ValidationMessage` (VM), which contains the validation of an information block.

Once the network is formed, all vehicles send the information they collected by the sensors with OMs. All nodes update the content of their block as information is received. Once a vehicle fills a block, it initiates the validation phase and, after a random delay, sends the validated block to other nodes using a VM. The procedure starts again with the next block.

3.1.2 Routing Protocols

Before starting CS application, vehicles are requested to form a network; this is dictated by the need of quickly achieving a distributed decision, which considers all the local information. To this purpose, several routing protocols to provide Network Layer functionalities have been investigated and optimized to our use case, as presented in the following.

DTN oriented approach. DTN paradigm enable transfer of data over links that may lack continuous connectivity. For this reason, the purpose of many existing DTN routing protocols is to increase the likelihood of finding a path between sender and receiver [10]. Data dissemination in VANETs is performed with DTN protocols usually divided into two main families [77]: (i) Epidemic protocols, representing the simplest approach and (ii) Geographic protocols, which are based on nodes location. We focus on Epidemic protocols, as their inherent anycast addressing scheme is better suited to the characteristics of CS applications. Specifically, three specific techniques are investigated:

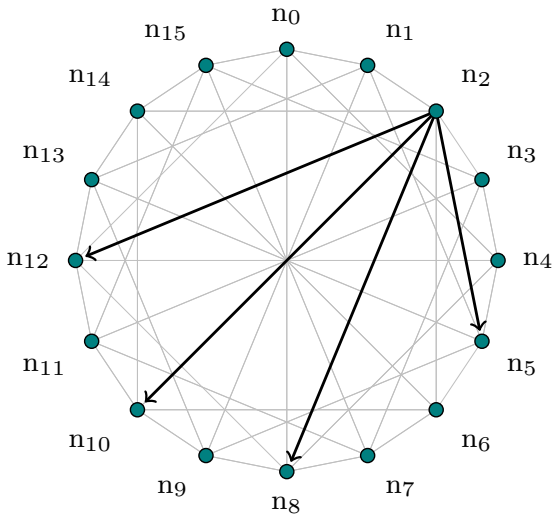

Figure 3.3: Chord achieved P2P overlay network topology.

- blind flooding (BF), *i.e.*, the simplest version of epidemic protocol, where each node forwards the received message to all its neighbors;
- TTL-based flooding (TF), that limits the retransmission of a message according to a time to live (TTL) counter;
- probability-based flooding (PF), each node retransmits the message to its neighbors with a probability P, with $0 < P < 1$.

It is worth noticing that the previous general routing approaches encompass the most relevant standardized routing schemes for the DTN domain, as the Epidemic that is well suited for individual and independent mobility models [43], as well as Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) protocol which relies on a delivery probability particularly suited in case of group mobility [4].

Network Coding. Another effective protocol investigated in this paper for information block dissemination to a community of vehicles relies on the generalized multiflow NC principle [44]. It enhance the basic DTN approach by allowing each device to store, carry and forward a random linear combination of the previously received packets. In particular, we adopted an epidemic DTN routing joined with a fountain code to limit the complexity

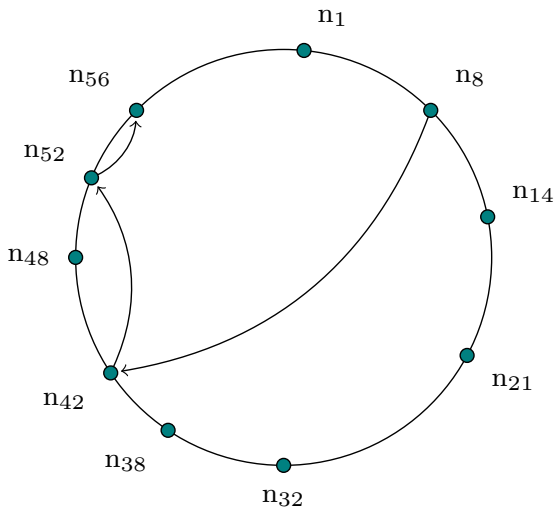

Figure 3.4: Typical Chord lookup resolution.

and overhead without requiring any topology information, and effectively exploiting the nodes mobility by iterate the basic scheme.

Chord. This approach is based on distributed hash tables (DHTs), used to realize a decentralized P2P overlay network [72], as shown in Figure 3.3. Chord provides the mapping of keys into nodes, used to pursue the resolution between L2 and L3 addresses.

Chord is extremely efficient in lookups resolving. Indeed, it needs only $O(\log N)$ messages [72] to reach any node in the network, as it depicted in Figure 3.4. Moreover, each node maintains information only about $O(\log N)$ other node, so that join and leave events only requires no more than $O(\log^2 N)$ messages [72]. When integrated in our framework, Chord provides a dynamic and distributed address resolution table, which adapts to join and leave events.

3.2 Performance Evaluation

The proposed architecture has been implemented and tested within the OMNeT++ environment, that is an object-oriented, time discrete message pass-

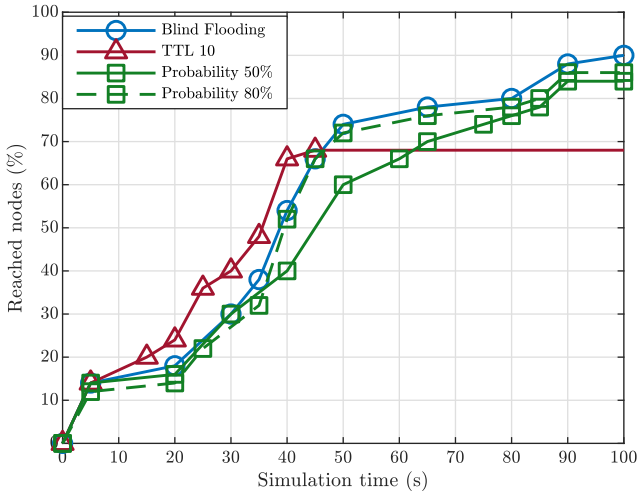

Figure 3.5: Percentage of reached nodes with DTN oriented approach.

ing driven network simulator, widely adopted for its modularity, high fidelity and flexibility. In addition to this it fully supports IEEE 802.11p standard through a comprehensive suite of models included in its Veins framework [23], which could be also extended towards 5G system via VeinsLTE. This allows our proposal to be aligned with the de facto standard for automotive industry, to accurately model the vehicular domain and to test its performance by means of network simulations as realistic as possible.

We first focus on the Network layer to eventually take into account the overall system figures.

3.2.1 DTN Oriented Approach

The scenario under investigation is a grid plan city, whose map has been imported from Open Street Map. When the simulation starts an accident occurs, and the damaged vehicle has to inform the surrounding ones (which we limited to 50 cars).

This simulation is realized using Veins' `Car` and `RSU` modules. The communication between nodes is provided by `Nic80211p`, which allow them to send and receive information via `WaveShortMessage`s. We also set the playground as a square of 10 km × 10 km, and the length of the simulation to

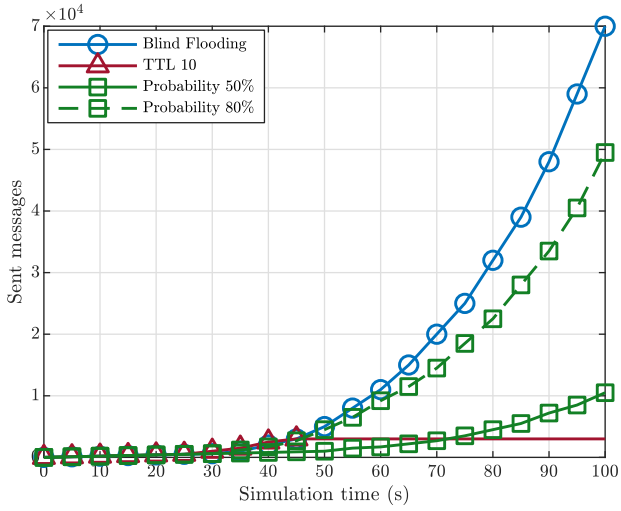

Figure 3.6: Protocol overhead in terms of sent messages by DTN oriented approach.

100 s.

We start with analyzing the percentage of reached cars in Figure 3.5. In particular TF based technique presents the worst performance, reaching about 70% of the nodes, and stopping after about 45 s. This is due to the fact that it often inhibits retransmissions, while BF and PF approaches improve the performance almost in the same way.

In addition to this, the protocol overhead, *i.e.*, the number of sent messages, is pointed out in Figure 3.6, where the differences among the various approaches are more remarkable. In particular, it is evident the advantage of PF methods, as they requires less overhead to achieve the same percentage of alerted vehicles. It is worth noting that adopting $P = 0.5$ the number of sent messages is about $1/7$ of the ones needed by BF.

3.2.2 Network Coding

For evaluating the performance of NC technique we focus on the typical diamond topology shown in Figure 3.7, with two Relay nodes between Sender (S) and Receiver (R). The Relay nodes can only perform store, combine and forward of each received packet.

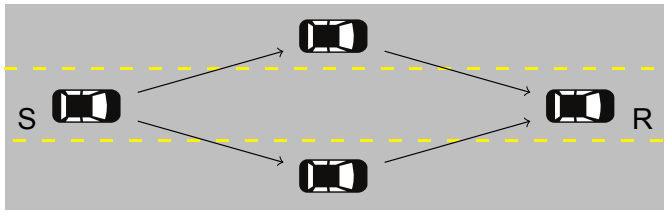

Figure 3.7: Reference scenario for testing the performance of the NC approach.

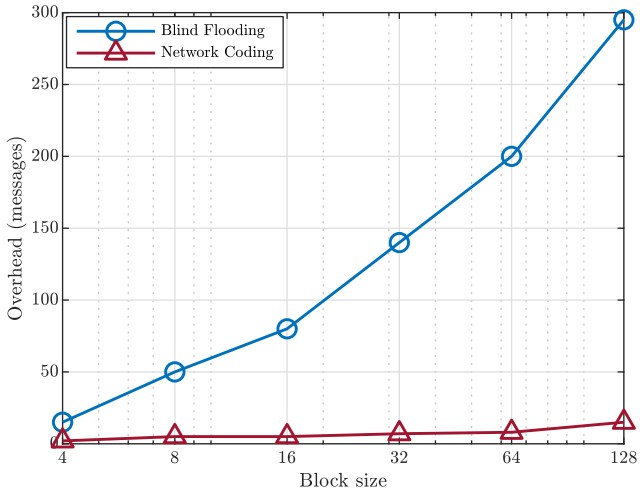

Figure 3.8: Protocol overhead in terms of sent messages by NC appraoch.

This simulation is realized relying on an external library, `Eigen`, which manages the coding and decoding of messages. Given the complexity of these phases, we necessarily had to limit the number of nodes. In this scenario we used a module defined from scratch.

In Figure 3.8 we can observe the overhead introduced by NC; in this particular scenario, NC is very efficient. The gap w.r.t. BF is remarkable and increases at the increasing of packet block size, thank to a kind of diversity gain provided by the two independent Relays.

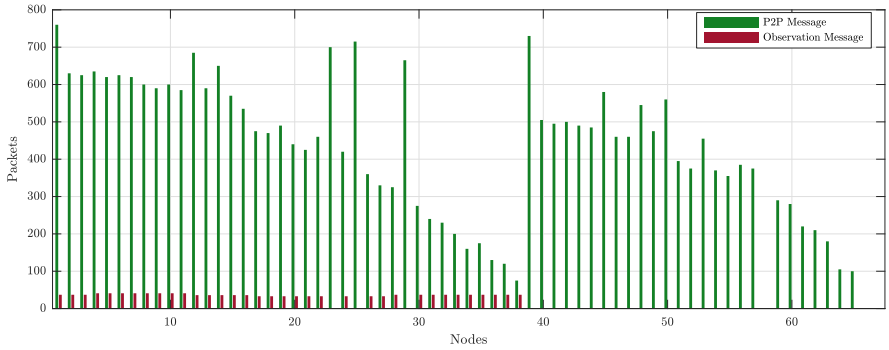

Figure 3.9: Protocol overhead in terms of sent messages by Chord approach.

3.2.3 Chord

In order to evaluate the integrated framework, we adopted a more realistic map (*i.e.*, the default Erlangen map provided by SUMO mobility simulator), with realistic traffic patterns. This scenario is realized using `Car` and `RSU` modules from Veins and the communication is provided by `Nic80211p`. To build and maintain the Chord network consistency we also implement a new message type, `P2PMessage` (P2PM). In our simulation the network is composed on the average by 35 cars and it stops when the validation of a specific block is reached. In fact, this condition allows to establish vehicles communities are established that can communicate with each other according to the Small-World Network paradigm.

As shown in Figure 3.9, we evaluate the overall number of packets sent by each node. Here we consider only messages P2PMs and OMs, which are the most relevant in our scenario. We can observe the presence of two different networks formed, wherein nodes present the same tendency. This is due to the signaling of Chord protocol: for each node joining the network, some routing information needs to be updated. In particular, we can observe that the first nodes to join the network are those that send more messages, whilst the overhead gradually decreases for the other ones. We can also point out that the first community completed the network forming process, and its nodes start to send OMs. The other group has not yet formed, as the simulation is programmed to stop when the first block is filled. Moreover, the number of P2PMs is much higher than OMs, this highlights that the formation of the Chord network is the more critical phase.

In conclusion, it could be pointed out that the number of messages per vehicles needed to disseminate a block of information is roughly equal to 2×10^3, 10^3, 10^2, 2×10^2, for BF, DTN, NC and Chord, respectively, but the latter always allows a reliably data distribution, thus representing the best candidate.

Although this is not the specific focus of our paper, however, privacy mechanism need to be addressed in a VANET to provide protection for the user data from profiling and tracking. Privacy violation can also severely affect the message dissemination via a specific routing scheme, as it happens in the presence of Sybil attack [89]. In addition, privacy represents an open issue for BC, where anonymous or non-anonymous schemes are still under discussion in order to provide user to publish transactions without tracking their network addresses, or to fetch details of a specific transaction without revealing which transactions they seek [37].

A couple of approaches, usually used for providing anonymous services, could be integrated in the proposed framework, which are Group Signature and Pseudonymous Authentication schemes, where, in the former one, a vehicle is issued a group private key with which it signs a message; while in the latter one, each vehicle stores a set of identities [51].

Chapter 4

Performance Analysis of an Edge Computing SaaS System

The main feature of MEC is to move computing and storage to the network edges (e.g., radio access point/base stations), enabling resource-limited mobile devices to support computation-intensive and latency-critical applications. This chapter deals with the performance evaluation of a MEC system providing computational capabilities to users within a limited area, according to the software as a service (SaaS) paradigm. In particular, a Markov multi-server queuing system model with requests reneging is proposed in order to accomplish the performance evaluation of the MEC system and derive the minimum number of processors to be allocated in order to fulfill specific service requirements in terms of resulting requests dropping probability. Finally, the pertinence of the proposed Markov queueing model is validated by comparing the obtained analytical predictions with numerical results derived by resorting to extensive computer simulations carried out under the assumption of realistic operating conditions.[1]

This approach is based on an application scenario where MEC facilities are provided within a limited geographical area according to the SaaS paradigm to MEDs [40]. This service model is suitable for specific real-time

[1]This chapter has been published as part of "Performance Analysis of an Edge Computing SaaS System for Mobile Users" in *IEEE Transactions on Vehicular Technology*.

applications such as cooperative gaming, AR, and smart tourism. The SaaS facility is assumed to be located at the network BS [11, 21]. Being the outcomes of the processed tasks of interest only within the SaaS service area, migration of uncompleted tasks to adjacent SaaS servers is not supported, *i.e.*, any uncompleted task within the sojourn time of the associated MED in the service area of the SaaS server is dropped. From the users point of view, a dropped uncompleted task is considered more annoying than a refused processing request. As a consequence, SaaS providers are interested in avoiding uncompleted tasks dropping as much as possible.

This chapter provides a suitable analytical model to properly design the computation capability of a MEC node offering its computational power for the users to offload their tasks. The goal of this investigation is to guarantee a task dropping probability lower than a given target value. The proposed analytical model has been necessarily based on some simplifying assumptions to make the problem affordable in a closed form. However, the accuracy of the proposed approach will be validated by providing comparisons of the obtained analytical predictions with simulation results, derived under the assumptions of actual operating conditions. Summarizing, the main contributions of this chapter are:

- the proposal of a Markov model with requests reneging to analyze the behavior of a MEC system performance in terms of task dropping probability and mean time spent by a computation request in the MEC system;
- validation of the proposed analytical model by providing comparisons between the obtained analytical predictions and simulation results;
- a design procedure based on the proposed analysis devoted to identify the minimum number, *i.e.*, optimal, of computation resources to be allocated at the MEC system in order to fulfill specific quality of service (QoS) requirements in relation to different working conditions;
- validation of the proposed design procedure by comparing the obtained analytical predictions with simulation results derived under the assumption of real world working conditions.

4.1 System Model

4.1.1 Reference Scenario

We consider here a MEC system to support SaaS applications for MEDs within a limited service area. A 5G cellular wireless system is supposed to enable, according to a URLLC usage scenario, MEC applications with strict latency and reliability requirements [64]. According to this, we assume here a loss-free, reliable, low latency access to the MEC facilities by each MED within the service area. Network function virtualization (NFV) is also supposed to be used, in order to realize a truly virtualized platform [52,53,71] able to perform a dynamic allocation of computational resources to different requests. Moreover, we assuming that the required computing resources of all the MEDs' requests are the same for all the supported applications and configured as individual virtual machines (VMs), from here on referred to as servers. Moreover, the MEC platform is assumed to be deployed by the service providers in the neighborhood of the network access point (*i.e.*, the BSs of the 5G network) [53]. As a consequence, the service area of a MEC facility and the BS coverage area can be considered equivalent and are used interchangeably.

This work deals with a worst case scenario, since MEC services are provided only within the coverage area of the associated BS. As a consequence, migration of uncompleted service requests to adjacent SaaS facility is not supported, *i.e.*, any uncompleted service request within the sojourn time of the associated MED in the area of the BS is dropped.

4.1.2 Analytical Model

The proposed analytical approach is based on a generalization of a pure birth and death process [13] where the state variable is the number of service requests in the MEC facility, both in service or awaiting computation in a first in, first out (FIFO) queue. As a consequence, a Poisson process with rate λ is assumed for the requests arrivals, with each request requiring an exponentially distributed amount of computation time (*i.e.*, service) with mean value $\overline{T}_S = 1/\mu_S$. For the case under consideration, the departures from the MEC facility are given by completion of the requests service or by the expiration of the sojourn time of the MEDs with service requests in the MEC server. Hence, under the assumption of C servers (*i.e.*, VMs) available

	0.50 km	0.75 km	1.00 km	1.50 km	2.00 km	5.00 km	10.0 km	15.0 km	20.0 km
15 m/s	0.0215	0.0144	0.0108	0.0072	0.0054	0.0022	0.0011	0.0007	0.0005
20 m/s	0.0287	0.0192	0.0144	0.0096	0.0072	0.0029	0.0014	0.0010	0.0007
25 m/s	0.0359	0.0239	0.0180	0.0120	0.0090	0.0036	0.0018	0.0012	0.0009
30 m/s	0.0431	0.0287	0.0215	0.0144	0.0108	0.0043	0.0022	0.0014	0.0011
35 m/s	0.0503	0.0335	0.0251	0.0168	0.0126	0.0050	0.0025	0.0017	0.0013

Table 4.1: Values of Sojourn Time Expiration Rate μ_H (s^{-1}) for Different Values of MEC Area Radius R and Average MED Speed V

at the MEC site to MEDs service requests, the resulting queuing system can be considered as a multi-server Markov queuing system with customers reneging [33], either from queue or service.

Being T_S exponentially distributed with mean value $1/\mu_S$, we have that the departure rate due to a request service completion results to be μ_S. Likewise, an additional requests departure rate, μ_H, has to be taken into account due to the expiration of the sojourn time of MEDs with service requests in the MEC system in the associated MEC service area that, according to [30], has been assumed here as exponentially distributed[2] with mean value $1/\mu_H$, under the following assumptions:

- MEDs moving at constant speed on one of four orthogonal direction;
- MEC service area assumed to be hexagonal[3];
- area boundary crossings independent from a request service duration.

Therefore, μ_H results to be [30]:

$$\mu_H = \frac{3+2\sqrt{3}}{9}\frac{V}{R} \approx 0.7182\frac{V}{R}, \tag{4.1}$$

where V represents the average MED speed and R is the MEC service area

[2]The exponential distribution assumption for the MED sojourn time in a MEC service area is usually made in the literature mainly for reasons of mathematical tractability of the resulting performance evaluation problem [19,28,42,49,50]. In our case, we have validated the correctness of this assumption by comparing the obtained analytical predictions with the case of a hyper Erlang-j,k sojourn time distribution as suggested in [61] by resorting to computer simulations due to the complexity to solve the associated performance evaluation problem in a closed form. In particular, the exponential sojourn time assumption results in a slight upper bound with no a significant impact on the MEC parameters selection in relation to given service requirements.

[3]This may not be a thoroughly realistic assumption but it is classical and widely accepted in the literature [38, 55, 66, 90], mainly to address the performance evaluation problem on an analytical basis.

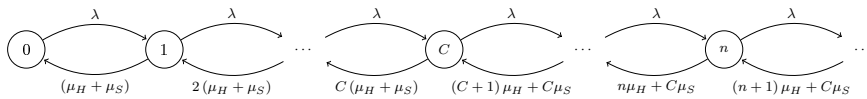

Figure 4.1: System's state transition diagram.

radius. Table 4.1 provides μ_H values for different scenarios (MED speed from urban roads to highways and MEC service area extension from femtocells, picocells up to macrocells).

Under the hypothesis of a steady-state condition, the state transition flow diagram of this birth-death process results as depicted in Figure 4.1, where n, *i.e.*, the number of requests in the MEC facility, is the system state and C is the maximum number of available VMs (*i.e.*, servers).

The parameters of the birth-death process under consideration are:

$$\lambda_n = \lambda \quad \forall n, \tag{4.2}$$

and

$$\mu_n = \begin{cases} n\left(\mu_H + \mu_S\right) & 1 \leq n < C \\ n\mu_H + C\mu_S & n \geq C \end{cases}. \tag{4.3}$$

In (4.3), we have that the upper row is referred to the case of a number of service requests in the MEC system less than the number of available servers, i.e., $1 \leq n < C$, so that any new arrived request receives service immediately. Conversely, with reference to lower row of (4.3), *i.e.*, $n \geq C$, we have that a queue is formed. Moreover, from standard queueing theory results [13], the overall mean request departure rate results to be $n\left(\mu_H + \mu_S\right)$ for the first case and $n\mu_H + C\mu_S$ if $n \geq C$.

Finally, we can note that the steady-state condition is reached if

$$\lambda < C\mu_S. \tag{4.4}$$

Hence, being (4.4) verified, P_n, the probability of having n requests in the MEC facility, both queued or in service, can be derived for $0 \leq n < C$ as:

$$P_n = P_0 \frac{\lambda^n}{n!\left(\mu_H + \mu_S\right)^n}, \tag{4.5}$$

and for $n \geq C$ as:

$$P_n = P_0 \frac{\lambda^n}{(C-1)!\,(\mu_H+\mu_S)^{C-1} \prod_{i=C}^{n} (i\mu_H + C\mu_S)}. \tag{4.6}$$

where P_0 is the probability of having an empty system, derived from the normalized condition, as in (4.7).

$$P_0(C) = \frac{1}{\sum_{n=0}^{C-1} \frac{\lambda^n}{n!\,(\mu_H+\mu_S)^n} + \frac{1}{(C-1)!\,(\mu_H+\mu_S)^{C-1}} \sum_{n=C}^{\infty} \frac{\lambda^n}{\prod_{i=C}^{n} (i\mu_H + C\mu_S)}} \tag{4.7}$$

Recalling that the statistics here considered refer to requests both in service or waiting in queue, it is straightforward to note that when C servers are available, while n is less than C a new arrived request receives service immediately. Conversely, if n is equal or greater than C, the incoming request has to wait and enters the MEC queue. Therefore, $P_Q(C)$, *i.e.*, the probability that when a request arrives when all the C servers are busy, is a parameter of interest for this analysis. In particular, through the application of standard queueing theory results based on (4.2), (4.3) and the steady-state transition probability diagram given in Figure 4.1, $P_Q(C)$ can be derived as in (4.8).

$$P_Q(C) = \frac{\frac{1}{(C-1)!\,(\mu_H+\mu_S)^{C-1}} \sum_{n=C}^{\infty} \frac{\lambda^n}{\prod_{i=C}^{n} (i\mu_H + C\mu_S)}}{\sum_{n=0}^{C-1} \frac{\lambda^n}{n!\,(\mu_H+\mu_S)^n} + \frac{1}{(C-1)!\,(\mu_H+\mu_S)^{C-1}} \sum_{n=C}^{\infty} \frac{\lambda^n}{\prod_{i=C}^{n} (i\mu_H + C\mu_S)}} \tag{4.8}$$

Then, the mean value, $\overline{T}_t(C)$ of the time that a request spends in the MEC facility with C servers, defined as the mean value of time elapsed from its arrival at the MEC facility to the instant of its departure, can be determined through the application of Little's Formula [13]. The final result is given in (4.9).

$$\overline{T}_t(C) = \frac{\sum_{n=1}^{C-1} \frac{n\lambda^n}{n!\,(\mu_H+\mu_S)^n} + \frac{1}{(C-1)!\,(\mu_H+\mu_S)^{C-1}} \sum_{n=C}^{\infty} \frac{n\lambda^n}{\prod_{i=C}^{n}(i\mu_H + C\mu_S)}}{\lambda\left[\sum_{n=0}^{C-1} \frac{\lambda^n}{n!\,(\mu_H+\mu_S)^n} + \frac{1}{(C-1)!\,(\mu_H+\mu_S)^{C-1}} \sum_{n=C}^{\infty} \frac{\lambda^n}{\prod_{i=C}^{n}(i\mu_H + C\mu_S)}\right]} \tag{4.9}$$

In a MEC system, the QoS increasing refers to the capability to lower the requests dropping probability, $P_D(C)$, defined as the probability that a request entering a MEC facility with C servers leaves it before service completion due to the expiration of the sojourn time of the related MED in the MEC service area. In particular, $P_D(C)$, can be obtained from P_n [31], as follows:

$$P_D(C) = \sum_{n=0}^{\infty} \frac{n\mu_H P_n}{\lambda}. \tag{4.10}$$

Hence, by taking into account the formal definition of the terms P_n given in (4.5) and (4.6), it is easily to verify that:

$$P_D(C) = \mu_H \overline{T}_t(C). \tag{4.11}$$

The goodness of this model will be verified later in Section 4.2 by comparing the obtained analytical predictions with the simulation results derived by considering real world cases.

Moreover, we can easily verified that $P_D(C)$ reaches a minimum value, $P_{D_{min}}$, for given values of λ, μ_S, μ_H, when $C \to \infty$. Recalling the definition of $\overline{T}_t(C)$ in (4.9), through some algebraic manipulations (see Section A.1) it is straightforward to verify:

$$\lim_{C\to\infty} \overline{T}_t(C) = \frac{1}{\mu_H + \mu_S}, \tag{4.12}$$

consequently, $P_{D_{\min}}$ results as follows:

$$P_{D_{\min}} = \lim_{C\to\infty} P_D(C) = \frac{\mu_H}{\mu_H + \mu_S}. \tag{4.13}$$

Finally, it is important to highlight that the results obtained on the basis of the proposed analytical model, as shown in more detail later, hint at

an optimal definition of parameter C, *i.e.*, C_{opt}, in order to match specific constraints occurring in MEC applications in terms of target values, $P_{D_T}(C)$, of $P_D(C)$.

4.2 Numerical Results

This Section provides some numerical results to validate the proposed Markov analytical model and the derivation of C_{opt} under different traffic load and MEDs mobility conditions. In particular, the simulation results provided here have been derived by resorting to the use of the OMNeT++ simulator and averaging the obtained numerical results over 10^2 runs assuming 10^7 requests for each considered case. Two different requests service time distributions have been considered, namely, a geometric distribution, to better represent the discrete nature of the requests service time, and a Pareto distribution, well known as a good model for real data traffic [46]. In both cases, the same mean value as the exponential distribution assumed in defining the proposed analytical model has been considered. In particular, this mean value, $\overline{T}_S = 1/\mu_S$, has been set equal to 4 s in order to be representative of the duration of a request service[4]. In addition to this, the value of the MEC load factor ρ, defined as λ/μ_S, ranges from 0.15 to 15.0, with λ set accordingly, in order to take into account MEC usage conditions from low to high, respectively. Likewise, three different values of μ_H are used to model the expiration of the sojourn time of a MED in the MEC service area, $\{0.0015, 0.0100, 0.0250\}\,\mathrm{s}^{-1}$, representing low, medium, and high MEDs mobility profile, respectively, as shown in Table 4.1. Finally, P_{D_T} is defined using Δ as a bias factor with respect to $P_{D_{\min}}$ as follows:

$$P_{D_T} = \big(1 + \Delta\big) P_{D_{\min}}, \tag{4.14}$$

with $P_{D_{\min}}$, given as in (4.13), dependent on the selected working conditions.

Our analysis starts by focusing on the case of a geometric distribution for the requests service time in order to take into account the discrete nature of the requested service, as it can be decomposed into a finite series of service times, *i.e.*, the atomic operations to be executed. Hence, the distribution for

[4]It is important to note that, due to the assumption of a 5G network supporting URLLC services, we can assume as negligible the network delay with respect to the assumed mean service time value.

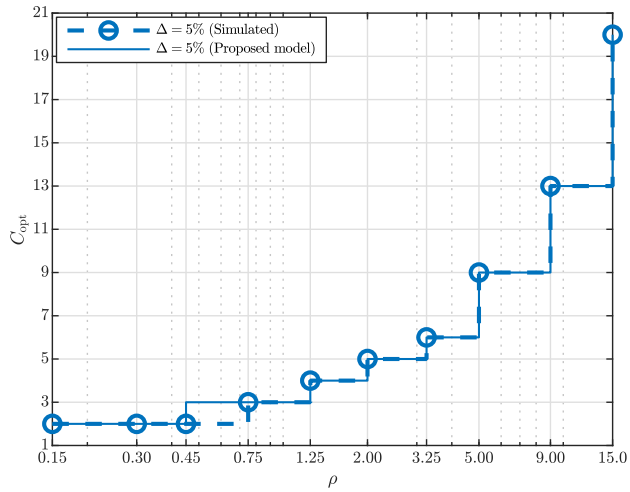

Figure 4.2: C_{opt} when ρ varies, with $\mu_H = 0.0015s^{-1}$, $\Delta = 5\%$, and geometric distribution of T_S.

T_S with mean $1/\mu_S$ results to be:

$$\mathbb{P}\{T_S = \alpha x\} = \alpha \left(1 - \frac{\alpha\mu_S}{\alpha\mu_S + 1}\right)^x \frac{\alpha\mu_S}{\alpha\mu_S + 1}\,, \tag{4.15}$$

where α is the service time of the single instruction, x is the number of instructions to be executed, and αx is the actual service time.

In Figures 4.2–4.4 the C_{opt} values derived according to the proposed analytical approach are compared with those derived by simulations for the case of a geometric distribution for the requests service time. A good agreement between analytical predictions and simulation results is evident in all the Figures. Moreover, we can note from these Figures that the MEDs mobility slightly influences the resulting C_{opt} values as a consequence of (4.13), resulting somewhat appreciable only under high traffic load conditions.

As an alternative to model the requests service time, we have considered the Pareto distribution defined as:

$$f_S(x) = \frac{1}{k\sigma}\left(1 + \frac{x}{\sigma}\right)^{-\left(\frac{1+k}{k}\right)} \qquad x \geq 0,\ 0 < k < 1, \tag{4.16}$$

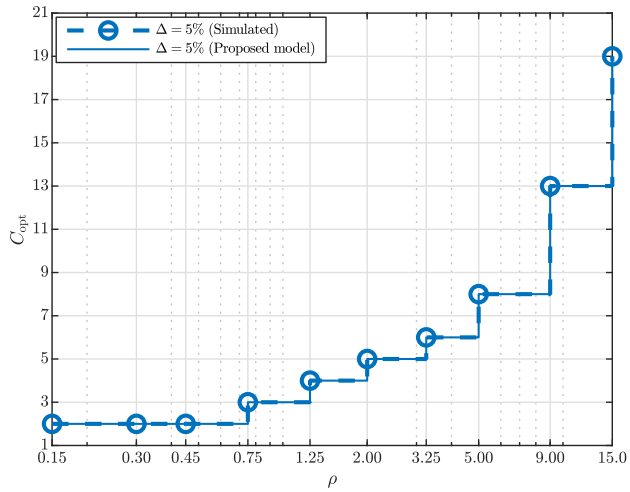

Figure 4.3: C_{opt} when ρ varies, with $\mu_H = 0.0100s^{-1}$, $\Delta = 5\%$, and geometric distribution of T_S.

with σ given by

$$\sigma = \frac{1}{\mu_S}\left(\frac{1-k}{k}\right), \tag{4.17}$$

in order to have a mean value equal to $1/\mu_S$.

As before, in Figures 4.5–4.7 the analytical predictions for the C_{opt} values are compared with those derived by resorting to Monte Carlo method. Here again we can note a good agreement between analytical predictions and simulation results in addition to a slight influence of the considered MEDs mobility profiles on the obtained results.

In Table 4.2 some additional C_{opt} values are shown. An high MEDs mobility profile is assumed, *i.e.*, $\mu_H = 0.0250s^{-1}$. Furthermore, three different values of Δ are considered to evaluate P_{D_T}, *i.e.*, 1%, 3%, and 5%, respectively. This Table confirms the effectiveness of the proposed analytical approach in deriving the number of servers needed to have a $P_D(C)$ value below a given P_{D_T} in all the considered cases. A slight overestimation is only evident under low value of Δ and high traffic load conditions in the case of the Pareto distribution.

To complete our analysis, Figures 4.8–4.11 show the request dropping

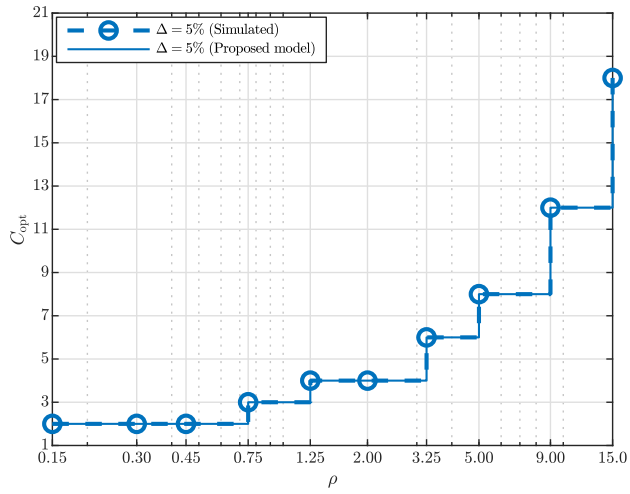

Figure 4.4: C_{opt} when ρ varies, with $\mu_H = 0.0250s^{-1}$, $\Delta = 5\%$, and geometric distribution of T_S.

probability, $P_D(C)$, as a function of C (ρ) for different values of ρ (C). Due to space limitation only the case of a MEDs medium mobility profile is considered. Here, a good agreement for the case of a geometric distribution of T_S can be noted, while a slight overestimation (as expected) arises in the case of the considered Pareto distribution.

	ρ	0.15	0.30	0.45	0.75	1.25	2.00	3.25	5.00	9.00	15.0
Geometric	$\Delta = 1\%$	2	3	3	4	4	6	7	9	14	20
	$\Delta = 3\%$	2	2	3	3	4	5	6	8	13	19
	$\Delta = 5\%$	2	2	2	3	4	4	6	8	12	18
Pareto	$\Delta = 1\%$	1	2	2	2	3	3	4	6	10	16
	$\Delta = 3\%$	1	2	2	2	3	3	4	6	10	16
	$\Delta = 5\%$	1	2	2	2	3	3	4	6	10	16
Proposed model	$\Delta = 1\%$	2	3	3	4	4	6	7	9	14	20
	$\Delta = 3\%$	2	2	3	3	4	5	6	8	13	19
	$\Delta = 5\%$	2	2	2	3	4	4	6	8	12	18

Table 4.2: C_{opt} Values with $\mu_H = 0.0250s^{-1}$

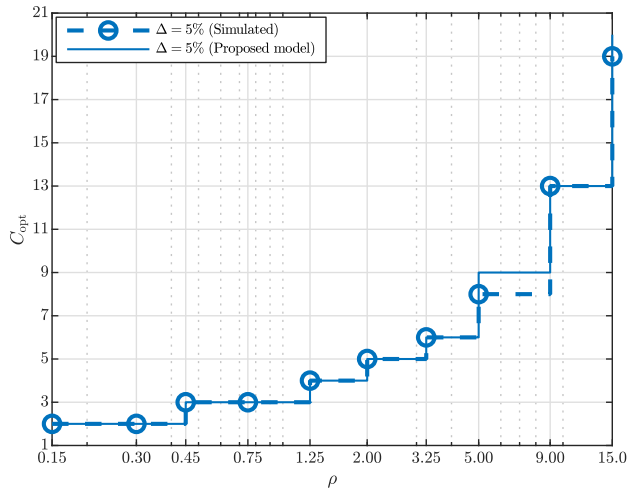

Figure 4.5: C_{opt} when ρ varies, with $\mu_H = 0.0015s^{-1}$, $\Delta = 5\%$, and Pareto distribution of T_S.

Finally, to complete the analysis, Figures 4.12–4.15 show the average time a service request spends in the MEC system, $\overline{T}_t(C)$, as a function of parameters C and ρ, respectively, in comparison with the obtained analytical predictions. The same considerations raised in discussing the results provided in Figures 4.8–4.11 can be applied also to Figures 4.12–4.15. Moreover, it is important to note in these Figures that the minimum value of $\overline{T}_t(C)$ is lower than $\overline{T}_S$. This result is due to the reneging process, related to the expiration of the MEDs' sojourn time in the MEC service area. In particular, we can note that the impact of the reneging process is more evident in the case of the Pareto distribution as shown in Figures 4.13 and 4.15.

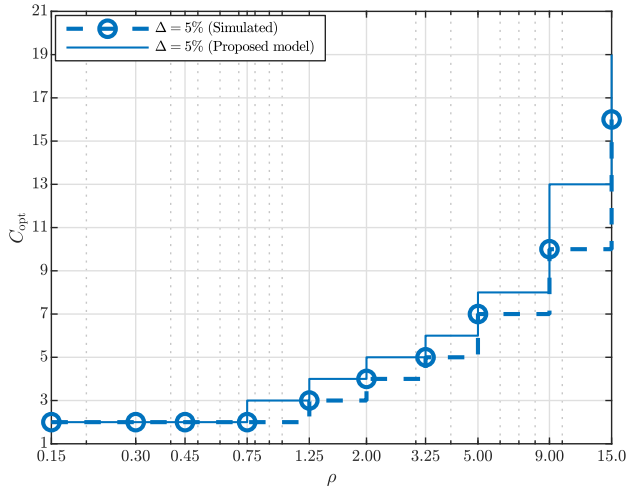

Figure 4.6: C_{opt} when ρ varies, with $\mu_H = 0.0100s^{-1}$, $\Delta = 5\%$, and Pareto distribution of T_S.

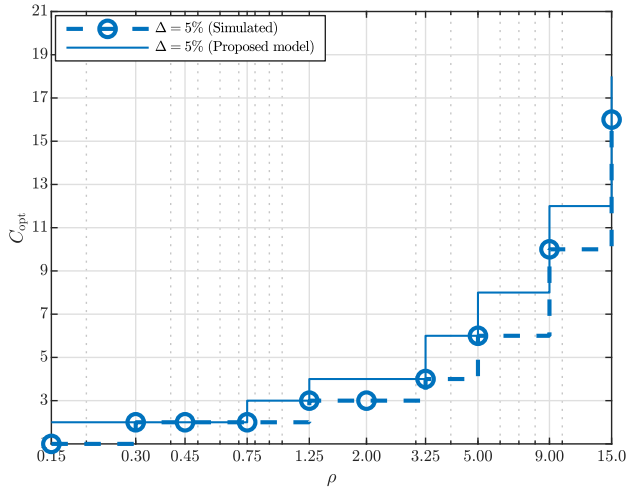

Figure 4.7: C_{opt} when ρ varies, with $\mu_H = 0.0250s^{-1}$, $\Delta = 5\%$, and Pareto distribution of T_S.

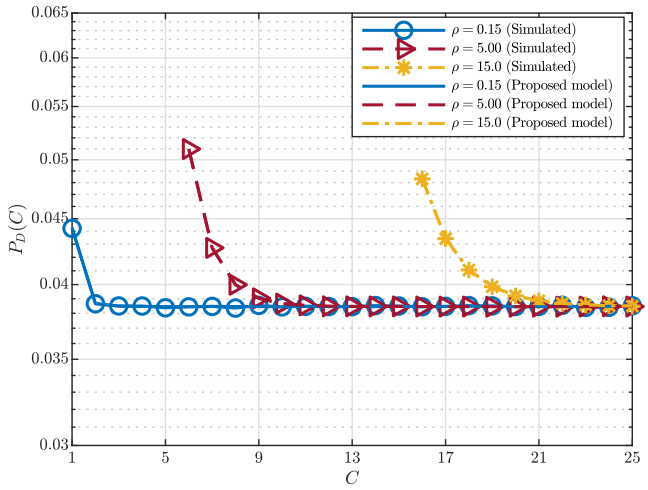

Figure 4.8: $P_D(C)$ when C varies, for different values of ρ, with $\mu_H = 0.0100s^{-1}$ and geometric distribution of T_S.

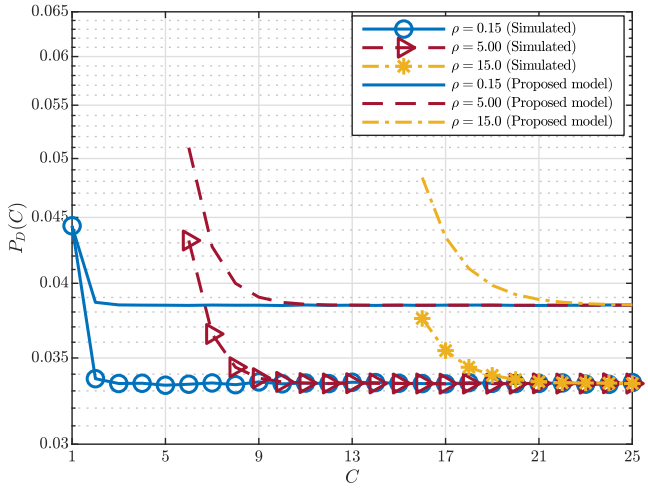

Figure 4.9: $P_D(C)$ when C varies, for different values of ρ, with $\mu_H = 0.0100s^{-1}$ and Pareto distribution of T_S.

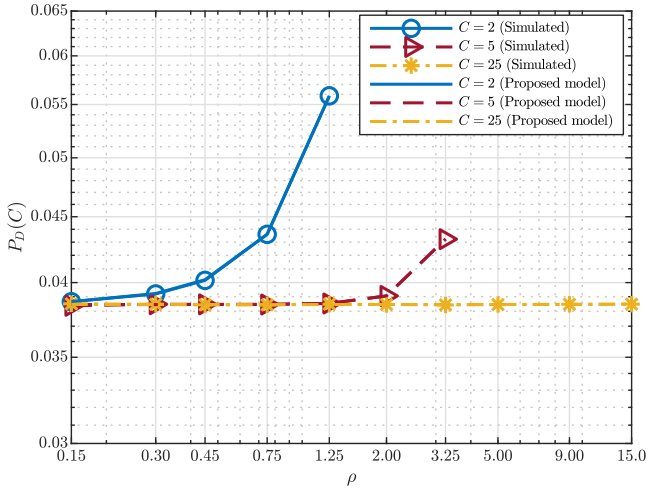

Figure 4.10: $P_D(C)$ when ρ varies, for different values of C, with $\mu_H = 0.0100s^{-1}$ and geometric distribution of T_S.

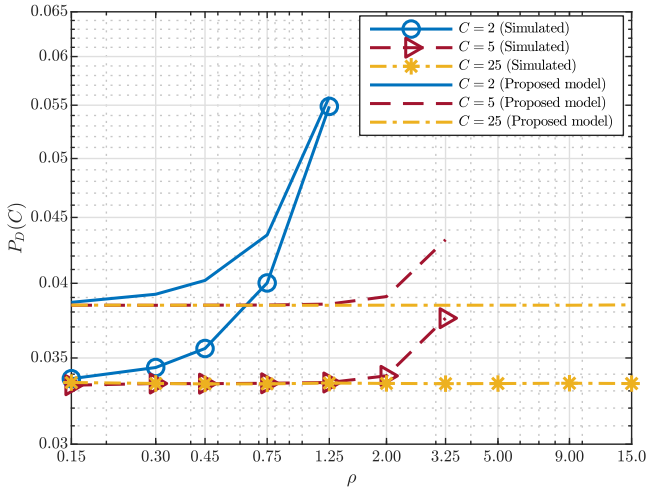

Figure 4.11: $P_D(C)$ when ρ varies, for different values of C, with $\mu_H = 0.0100s^{-1}$ and Pareto distribution of T_S.

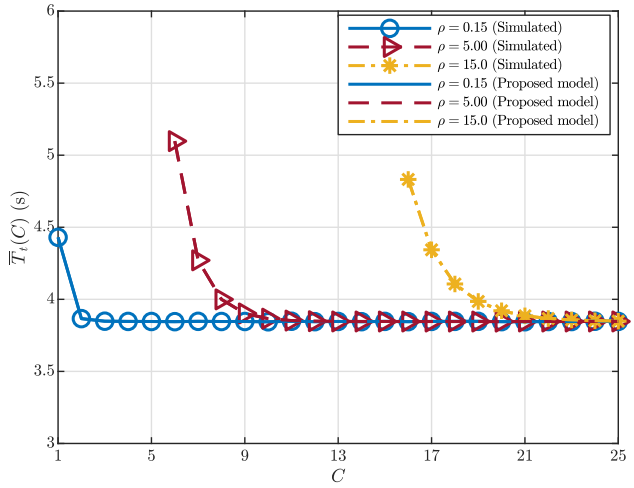

Figure 4.12: $\overline{T}_t(C)$ when C varies, for different values of ρ, with $\mu_H = 0.0100s^{-1}$ and geometric distribution of T_S.

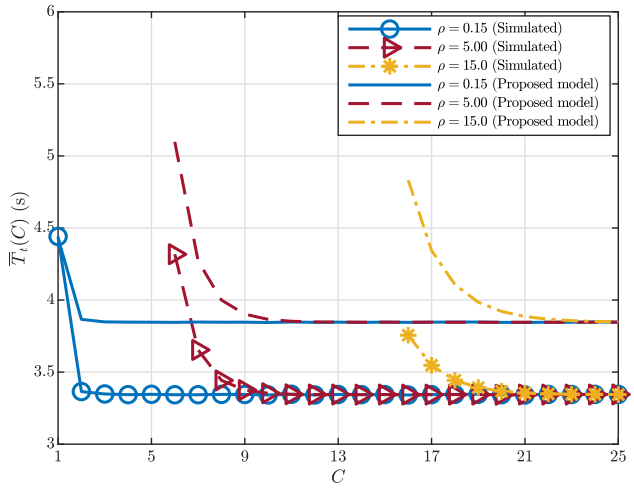

Figure 4.13: $\overline{T}_t(C)$ when C varies, for different values of ρ, with $\mu_H = 0.0100s^{-1}$ and Pareto distribution of T_S.

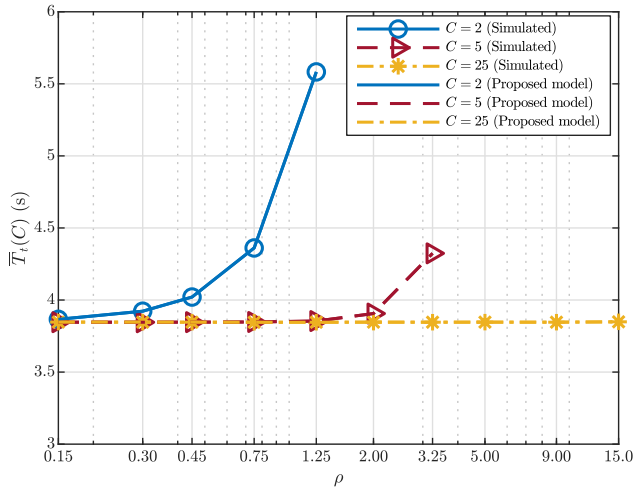

Figure 4.14: $\overline{T}_t(C)$ when ρ varies, for different values of C, with $\mu_H = 0.0100s^{-1}$ and geometric distribution of T_S.

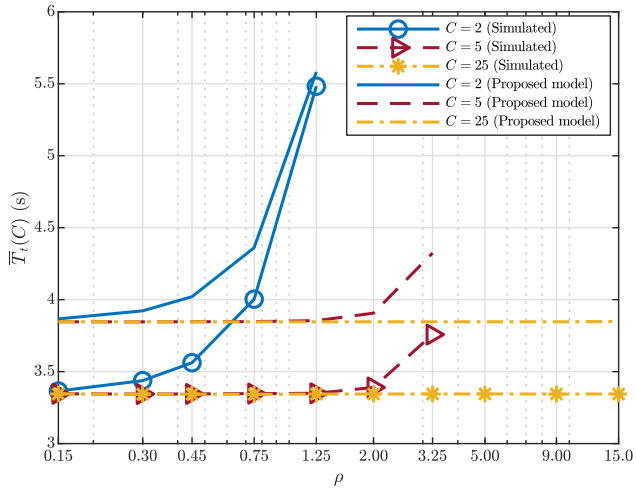

Figure 4.15: $\overline{T}_t(C)$ when ρ varies, for different values of C, with $\mu_H = 0.0100s^{-1}$ and Pareto distribution of T_S.

Chapter 5

Cloud vs. Edge Computing

This chapter deals with the comparison of MEC and CC systems, without considering service migration and assuming to provide services with low-latency requirements to IoT devices. Two analytical models are developed and validated to carry out the comparison, considering the total service time as perceived by users, i.e., including also the delays due to communications, as a key metric. The comparison shows some information on how a MEC system should be designed to handle a given load, and when it is convenient to use it. This is the first work that makes a similar comparison considering the choice of systems' fundamental parameters, thanks to the analytical models. In conclusion, depending on the system load and the resources provided to the MEC system, it is actually more convenient to use it, while for increasing load it is better to use the services offered by a CC system.[1]

Although there are several works on task dispatching in mixed MEC-CC systems [34, 54, 75, 83, 86], these investigate real-time procedures to decide whether an offloaded task should be served on the MEC node or the CC node, when these nodes have given and pre-determined fundamental parameters. The novelty of this work relies on a different approach: thanks to the analytical models we develop for MEC and CC, every fundamental parameter of the systems can be accurately predicted and, in principle, optimized

[1]Part of this work was conducted while the author was a visiting Ph.D. student at KTH Royal Institute of Technology, Stockholm (Sweden), from October 2019 to March 2020 (working with Prof. Carlo Fischione).

to minimize the resource consumption regardless when the real-time offload task dispatching should be taken. Thus we can consider these parameters in our study, and understand how they influence the choice of the system to offload requests on. This study, which to the best of our knowledge has not been already performed in the literature, is essential to minimize the resources used at MEC and CC sites, as they eventually turn in hardware costs to be sustained by internet service providers (ISPs) or application service providers (ASPs).

Specifically, we propose a statistical approach to understand how the system load and the amount of resources allocated influence offloading policies to MEC or CC systems. By defining total service time and probability of service completion as functions of the fundamental system parameters, we provide insights on how to design a MEC node to meet given requirements. On the other hand, this work can also assist in understanding when it is better to rely on a classic CC system, which has plenty of additional available resources, or even to blend them. Finally, considering that MEC systems are usually based on virtualized platforms [3], it is possible to use the results of this study to realize a dynamic management of resources for such systems, depending on the load of incoming requests.

In this scenario each MEC node is deployed near to the network access point [85], that is the BS of the 5G network, to better benefit from the advantages offered by the network, both in terms of latency and bandwidth. Thus the characteristics of the link connecting MEDs to the MEC node are more predictable and controllable. Under this natural assumption, each MEC node shares the coverage area with its host BS. Even though the network provides communication handover support, in this work the service handover is not taken into account, and MEC systems are assumed to work independently from each other [85]. Coherently, we address a worst case condition, since the migration of tasks to adjacent MEC servers is not considered. Moreover, we do not rely on a particular placement of MEC nodes. As a consequence, if a task is not completed before the user leaves the area managed by a MEC server, it is discarded.

CC nodes, on the other hand, are deployed within a much wider geographical area and are positioned much farther away from MED. In this case, the benefits of being on the 5G access network are lost, as data have to cross many different networks with unpredictable characteristics, and eventually experience much higher latency. To achieve comparable performance

with the MEC case, CC systems are also supposed to be affected by task dropping due to mobility. Actually, in this work we consider that the results of a computation are relevant only in a given geographical area, the service area, which is supposed to match the coverage area of a MEC node. If a MED leaves the service area before its task is completed, the computation is discarded.

Starting from this hybrid MEC-CC scenario, this chapter presents results about the total service time of offloaded requests both for MEC and CC nodes. A comparison between the performance of these two systems is shown, also considering applications with different latency requirements. Besides, it is pointed out how the system load and the amount of resources available to the systems affects the service time as perceived by MEDs. The performance analysis is then verified with data obtained from a simulator under realistic conditions. Summarizing, the main contributions of this chapter are:

- the proposal of two analytical models for MEC and CC systems to evaluate the total time an offloaded task spend in the systems;
- the validation of proposed analytical models by providing comparisons between the obtained predictions and simulation results;
- a comparison of the performance offered by MEC and CC systems in the case of applications with different latency requirements, and with different mobility profiles of MEDs;
- an analysis of the results obtained in order to identify a threshold useful to understand when to switch from MEC to CC and vice versa.

5.1 System Model

The architecture we propose is shown in Figure 5.1. A system to support computational offloading for low-latency constrained applications is considered, where each request can either be served by MEC or CC system in a mutually exclusive way. The MEC platform is assumed to be deployed by ISPs or ASPs in the neighborhood of the network access point (*i.e.*, the BS of the 5G network) [53], so that the time required to a MED to access the MEC facility is comparable to that typically required to reach a BS. As a consequence, the service area of a MEC system and the related BS coverage area can be considered equivalent. On the other hand, the CC system is deployed farther from MEDs and its communication latency is bigger and harder to model.

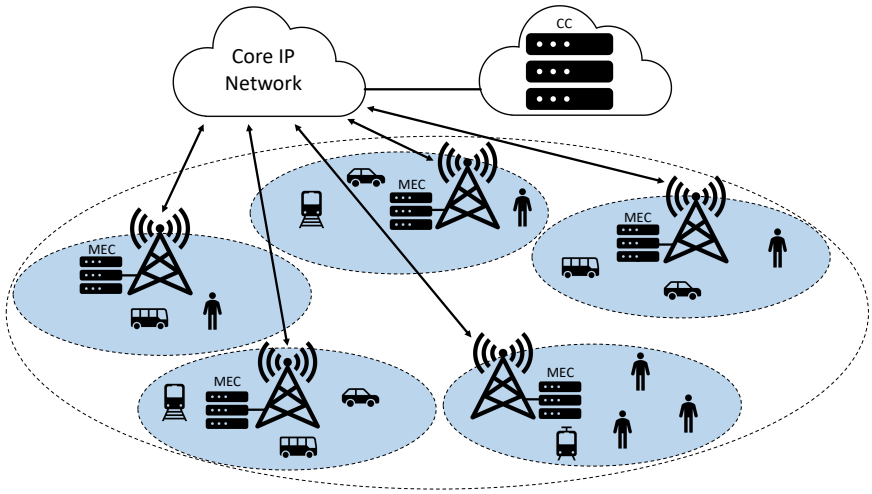

Figure 5.1: Hybrid MEC-CC reference scenario. Each BS has its own MEC node, their coverage areas are depicted in blue. BSs access to core IP network via wired connections. The CC server is located several hops away from MEDs.

Since each MEC systems are supposed to work independently from each others [85], the offloading service is provided only within the coverage area of the associated BS. Thus, the migration of uncompleted service requests to adjacent MEC is not supported *i.e.*, any uncompleted service request within the sojourn time of the associated MED in the MEC service area has to be interrupted and is considered as a dropping event. Moreover, without considering the migration of services, the presented results do not depend on a particular MEC nodes placement.

When the computational offloading request is sent to the CC system, the service is not performed on a specific network access point, but in a remote node. To represent a type of service tightly related to the geographical position, the CC service area is defined as equivalent to that of MEC node. As a consequence, even the CC system is characterized by a task dropping rate caused by MEDs mobility. Under this assumption the comparison between the two solutions is more fair.

In the system like the one depicted in Figure 5.1, when a MED has some data to be computed, there are two possible alternatives.

1. Data can be offloaded to the nearest MEC server. Here, the latency experienced by MEDs is low, but each service has to be completed before the MED leaves the MEC service area.
2. Alternatively, data can be offloaded to the CC node. In this case, the delays due to communications are higher, but the CC system has many more resources available.

The amount of computing resources provided to MEC and CC systems is a fundamental parameters of this work. The central issue that we want to investigate in this chapter is about the minimum amount of needed resources and when it is more convenient to perform the computation at the MEC or the CC node, for low latency services.

5.1.1 Mobile Edge Computing Model

The analytical MEC model is based on a generalization of a multi-server M/M/C queueing system, where the state variable stands for the number of offloading requests in the system, both in service or awaiting computation in a FIFO queue [14]. A reneging process, either from queue and service, is used to model the expiration of the MEDs' sojourn time in the MEC service area. Offloading requests are supposed to arrive according to a Poisson process with rate λ_e. Each task requires an exponentially distributed amount of computation time (*i.e.*, service) with mean value $\overline{T}_S = 1/\mu_S$, which is provided by C_e servers. The service time depends on the specific application under consideration, assuming that the C_e servers work at the maximum rate enabled by state of the art technologies. An additional requests departure rate, μ_H, has to be taken into account due to the reneging process that, according to [30], has been assumed here as exponentially distributed with mean value $1/\mu_H$.

$$P_{Q_e}(C_e) = \frac{P_{0_e}(C_e)}{(C_e-1)!\,(\mu_H+\mu_S)^{C_e-1}} \sum_{n=C_e}^{\infty} \frac{\lambda_e^n}{\prod_{i=C_e}^{n}(i\mu_H + C_e\mu_S)} \tag{5.1}$$

Under the steady-state condition, $\lambda_e < C_e\mu_S$, the probability that a request arrives at the MEC facility when all the C_e servers are busy, $P_{Q_e}(C_e)$, can be derived through the application of standard queueing theory results based on the steady-state probabilities as in (5.1), where $P_{0_e}(C_e)$ is as defined in (5.2).

$$P_{0_e}(C_e) = \left[\sum_{n=0}^{C_e-1} \frac{\lambda_e^n}{n!\,(\mu_H+\mu_S)^n} + \right.$$
$$\left. + \frac{1}{(C_e-1)!\,(\mu_H+\mu_S)^{C_e-1}} \sum_{n=C_e}^{\infty} \frac{\lambda_e^n}{\prod_{i=C_e}^{n}(i\mu_H + C_e\mu_S)} \right]^{-1} \quad (5.2)$$

An important feature of the system under consideration is the service completion probability, $P_{S_e}(C_e)$, defined as the probability that a request entering a MEC facility with C_e servers does not leave it, due to mobility, before service completion. In such a scenario, indeed, a departure due to mobility event is defined as the expiration of the sojourn time of the related MED in the service area. In [14] authors show how to obtain its complementary, $P_D(C_e)$, the probability that a request entering a MEC facility with C_e servers leaves it before service completion, which can be obtained from $\overline{T}_{t_e}(C_e)$, the mean of the total time a request spends in the MEC facility with C_e servers. As a result, $P_{S_e}(C_e)$ can be determined as follows:

$$P_{S_e}(C_e) = 1 - P_{D_e}(C_e) = 1 - \mu_H \overline{T}_{t_e}(C_e). \quad (5.3)$$

In general, in a M/M system with FIFO queueing policy the time needed for a service completion, T_C, is given by the sum of the time spent both in server and queue. Here T_{C_e} is defined as perceived by MEDs, *i.e.*, including also delays due to transmissions to and from the computation node. Moreover, the requests service completion probability has to be considered. A model of T_{C_e}, which can be used to obtain an analytical formulation, can be defined as follows:

$$T_{C_e} = P_{S_e}(C_e)\Big[T_S + P_{Q_e}(C_e)T_{Q_e}\Big] + \Delta_e, \quad (5.4)$$

where T_S is exponentially distributed with mean value $\overline{T}_S = 1/\mu_S$, T_{Q_e}, the time spent in the queue of the MEC node, can be approximated by an exponential distribution with mean value $\overline{T}_{Q_e}$, and Δ_e is used to represent transmission delays. As a simplifying hypothesis, we suppose that the sum of delays of task upload and results download follows a continuous uniform distribution on the interval $[0, 2\alpha_e]$, where α_e is the maximum latency we expect on each single link to and from the MEC server. The validity of this assumption will be verified later in Section 5.2 by using the simulation results derived under more realistic conditions. $P_{Q_e}(C_e)$ is used to consider

only the fraction of requests that actually enter the queue, while $P_S(C_e)$ is used to account only for requests that reach service completion, since uncompleted tasks can not be considered in T_{C_e}. From standard queueing theory results [13], the mean value of the time spent in queue, $\overline{T}_{Q_e}$, can be defined as:

$$\overline{T}_{Q_e} = \frac{Q_e}{\lambda_e} = \frac{\sum_{n=0}^{\infty} nP_{n+C_e}}{\lambda_e}, \tag{5.5}$$

where Q_e is the average number of offloading requests in the queue. Through some algebraic manipulations it is straightforward to verify (5.6).

$$\overline{T}_{Q_e}(C_e) = \frac{P_{0_e}(C_e)}{(C_e-1)!\,(\mu_H+\mu_S)^{C_e-1}} \sum_{n=C_e+1}^{\infty} \frac{(n-C_e)\,\lambda_e^{n-1}}{\prod_{i=C_e}^{n}(i\mu_H + C_e\mu_S)} \tag{5.6}$$

Recalling the definition of T_{C_e} in (5.4), its distribution can be defined as sum of two independent exponential distributions, composed with a uniform distribution. As a result, the probability density function (PDF) is as follows,

$$f_{T_{C_e}}(t) = \begin{cases} 0 & t \le 0 \\ \dfrac{1-\mathrm{e}^{-\mu_S t/P_{S_e}} - \mu_S P_{Q_e}\overline{T}_{Q_e}\left(1-\mathrm{e}^{-t/\left(P_{S_e}P_{Q_e}\overline{T}_{Q_e}\right)}\right)}{2\alpha_e\left(1-\mu_S P_{Q_e}\overline{T}_{Q_e}\right)} & 0 < t \le 2\alpha_e \\ \dfrac{\left(\mathrm{e}^{-2\alpha_e\mu_S/P_{S_e}}-1\right)\mathrm{e}^{-\mu_S t/P_{S_e}} - \mu_S P_{Q_e}\overline{T}_{Q_e}\mathrm{e}^{-t/\left(P_{S_e}P_{Q_e}\overline{T}_{Q_e}\right)}\left(\mathrm{e}^{2\alpha_e/\left(P_{S_e}P_{Q_e}\overline{T}_{Q_e}\right)}-1\right)}{2\alpha_e\left(1-\mu_S P_{Q_e}\overline{T}_{Q_e}\right)} & t > 2\alpha_e \end{cases} \tag{5.7}$$

where the dependence on C_e is omitted for the sake of compactness. As previously defined, the considered distributions are $T_S \sim \exp(\mu_S)$, $T_{Q_e} \sim \exp(1/\overline{T}_{Q_e})$, and $\Delta_e \sim \mathrm{unif}(0, 2\alpha_e)$. The definition of the cumulative distribution function (CDF) can be expressed integrating $f_{T_{C_e}}(t)$ as follows:

$$F_{T_{C_e}}(t) = \int_{-\infty}^{t} f_{T_{C_e}}(\tau)\,\mathrm{d}\tau, \tag{5.8}$$

which is equivalent to the probability that $T_{C_e} \le t$. The expanded version of (5.8) is in Section A.2. As can be noted, $F_{T_{C_e}}(t)$ is of paramount importance to evaluate the performance of the MEC system. It enables the comparison of the service completion time with a given target time, which can either represent a deadline for the considered application, or the performance offered by a competing system, *e.g.*, a classic CC one.

5.1.2 Cloud Computing Model

The CC model is based on a multi-server M/M/C system with the reneging process, either from queue and service, to model the exit of MEDs from the service area. A Poisson process with rate λ_c is assumed for the requests arrivals, with each request requiring an exponentially distributed amount of computation time (*i.e.*, service) with mean value $\overline{T}_S = 1/\mu_S$, provided by C_c servers working at the maximum rate. The task reneging rate, μ_H, is supposed to be the same as in the MEC case, since the service areas are supposed to be equivalent, and the same holds for the speeds of the MEDs. As a consequence, the reneging process is assumed as exponentially distributed with mean value $1/\mu_H$. The probability that a request arrives at the CC facility when all the C_c servers are busy, $P_{Q_c}(C_c)$, can be derived similarly to (5.1), and the probability of having an empty system, $P_{0_c}(C_c)$, can be determined similarly to (5.2).

As in the MEC case, the time needed for a service completion in the CC system, T_{C_c}, is defined as perceived by MEDs, including transmission delays. As a result, to obtain an analytical formulation, T_{C_c} can be modeled as follows:

$$T_{C_c} = P_{S_c}(C_c)\Big[T_S + P_{Q_c}(C_c)T_{Q_c}\Big] + \Delta_c, \tag{5.9}$$

where $P_{S_c}(C_c)$ is the requests service completion probability for a CC facility with C_c servers, defined similarly to (5.3), T_S is the exponentially distributed service time, T_{Q_c} can again be approximated by an exponential distribution with mean value $\overline{T}_{Q_c}$, defined as:

$$\overline{T}_{Q_c} = \frac{\sum_{n=0}^{\infty} nP_{n+C_c}}{\lambda_c}, \tag{5.10}$$

and Δ_c is used to represent communications delays, defined as a continuous uniform on the interval $[0, 2\alpha_c]$, where α_c is the maximum delay we suppose for uplink and downlink.

	MEC	CC
$\overline{T}_S$	4.0 s	4.0 s
ρ	0.15–15	1–300
C	1–25	50–500
μ_H	$0.025\,\mathrm{s}^{-1}$	$0.025\,\mathrm{s}^{-1}$

Table 5.1: MEC and CC Simulation Parameters

Then, the PDF of T_{C_c} results as follows,

$$f_{T_{C_c}}(t) = \begin{cases} 0 & t \leq 0 \\ \frac{1-\mathrm{e}^{-\mu_S t/P_{S_c}} - \mu_S P_{Q_c} \overline{T}_{Q_c} \left(1-\mathrm{e}^{-t/\left(P_{S_c} P_{Q_c} \overline{T}_{Q_c}\right)}\right)}{2\alpha_c \left(1-\mu_S P_{Q_c} \overline{T}_{Q_c}\right)} & 0 < t \leq 2\alpha_c \\ \frac{\left(\mathrm{e}^{-2\alpha_c \mu_S / P_{S_c}} - 1\right) \mathrm{e}^{-\mu_S t/P_{S_c}} - \mu_S P_{Q_c} \overline{T}_{Q_c} \mathrm{e}^{-t/\left(P_{S_c} P_{Q_c} \overline{T}_{Q_c}\right)} \left(\mathrm{e}^{2\alpha_c/\left(P_{S_c} P_{Q_c} \overline{T}_{Q_c}\right)} - 1\right)}{2\alpha_c \left(1-\mu_S P_{Q_c} \overline{T}_{Q_c}\right)} & t > 2\alpha_c \end{cases} \tag{5.11}$$

where the considered distributions are $T_S \sim \exp(\mu_S)$, $T_{Q_c} \sim \exp(1/\overline{T}_{Q_c})$, and $\Delta_c \sim \mathrm{unif}(0, 2\alpha_c)$. The CDF for the CC case be expressed integrating $f_{T_{C_c}}(t)$, as follows:

$$F_{T_{C_c}}(t) = \int_{-\infty}^{t} f_{T_{C_c}}(\tau)\,\mathrm{d}\tau, \tag{5.12}$$

while the expanded version is in Section A.3. As before, $F_{T_{C_c}}(t)$ is equivalent to the probability that $T_{C_c} \leq t$, and is used to evaluate the performance of the CC system. This model, together with the one defined in (5.8), is validated in the next Section.

5.2 Numerical Results

This Section provides some numerical results to validate the previous hypotheses, and to evaluate how the given $F_{T_C}(t)$ models behave w.r.t. variations in the number of servers. The simulation results have been derived using a simulated scenario built resorting to the OMNeT++ framework and averaging the obtained numerical results over 10^2 runs, assuming 10^7 requests for each considered case.

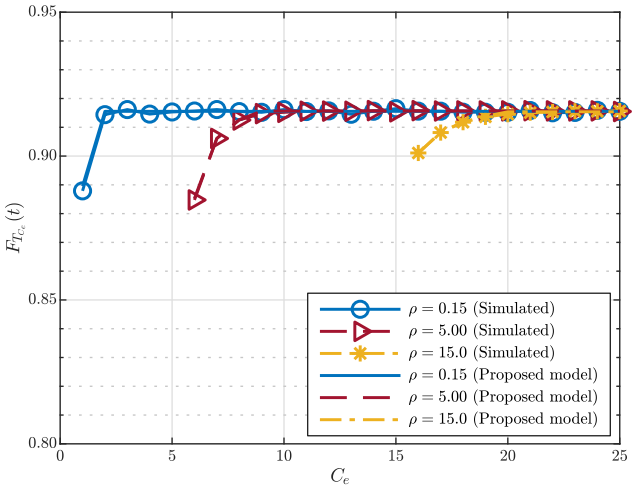

Figure 5.2: MEC simulated vs. proposed model comparison of $F_{T_{C_e}}(t)$ when C_e varies, with $t = 9.0\,\mathrm{s}$.

5.2.1 Mobile Edge Computing Model Validation

The following parameters are applied to both the simulated scenario and the proposed model to be validated. The mean service time, $\overline{T}_S = 1/\mu_S$, equal to 4.0 s, is chosen to be representative of the duration of a request service with a very loose time constraint, easily achievable also by a CC system [32]. The value of load factor ρ_e, defined as λ_e/μ_S, ranges from 0.15 to 15, with λ_e set accordingly, to consider several MEC usage conditions, from low to high request arrival rates, respectively. The number of server C_e ranges from 1 to 25, in order to represent both MEC nodes with fewer and more computational resources available. To model the expiration of the sojourn time of a MED in the MEC service area, μ_H is set to $0.025\,\mathrm{s}^{-1}$, which according to [14] represents high MED mobility. The chosen parameters are also shown in Table 5.1.

A realistic propagation delay is added to the simulated system, to represent the time needed to transmit the task from MED to MEC. The same time is used when the computation results are sent back to MEDs. It is modeled with a truncated normal distribution, to take into account the delay of a real MEC system. The lower tail of the distribution is truncated to

ensure that the obtained times are greater than zero. The resulting PDF is as follows:

$$\phi\left(t;\mu,\sigma\right)=\frac{\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\left(t-\mu\right)^{2}}{2\sigma^{2}}\right)}{\frac{\sigma}{2}\left(1+\operatorname{erf}\left(\frac{\mu}{\sigma\sqrt{2}}\right)\right)}. \tag{5.13}$$

The selected parameters are in accordance with the requirements of 5G systems for and URLLC [2]. The considered delay is quite small as the MEC node is supposed to be very close to the MED, ideally on the access network, and reachable with one or very few hops. On the other hand, in the proposed model α_e is chosen to be equal to the mean value of the delay described by (5.13), so that the two distributions have the same mean value.

The model provided in (5.8) is compared to simulated data w.r.t variations in the value of C_e. In this test t value is chosen as 225% of the mean service time, that is 9.0 s, and the obtained results are shown in Figure 5.2. As can be seen, the model provides $F_{T_{C_e}}(t)$ values which are almost indistinguishable with those obtainable from the simulation. Using the simplifying hypothesis of uniform delay distribution in the analytical model does not influence its effectiveness in predicting the performance of a more realistic system.

5.2.2 Cloud Computing Model Validation

The same kind of validation performed for MEC system is repeated here for the CC case. To assess the validation, the parameters showed in Table 5.1 are applied to both the simulated scenario and the proposed model. The average duration of a service is maintained the same as in the previous case *i.e.*, 4.0 s, and the times are still distributed exponentially. A larger values of C_c is chosen, to represent the higher resource availability of a typical CC system. In particular, this value ranges between 50 and 500. The additional departure rate due to reneging, μ_H is set to $0.025\,\mathrm{s}^{-1}$. Since service area is considered equivalent to the MEC case, using the same value of μ_H results in the same mobility profile. Finally, the value of load factor ρ_c, ranges from 1 to 300, with λ_c values set accordingly. A ρ_c closer to 1 represent the behavior of a Cloud system with a load comparable to a single MEC node, while values closer to 300 take into account the fact that a Cloud system is serving more MEC areas simultaneously.

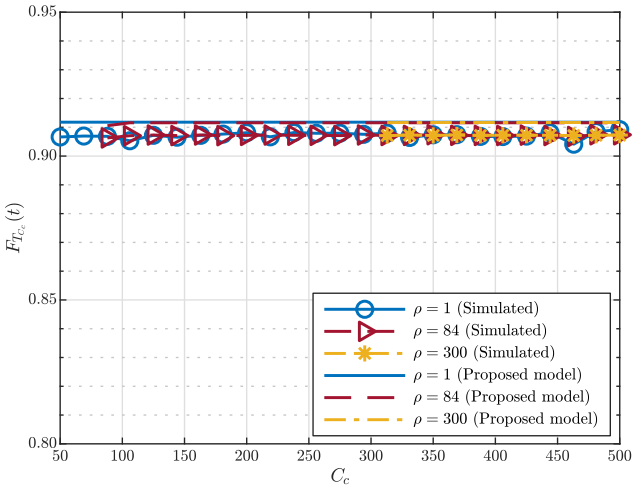

Figure 5.3: CC simulated vs. proposed model comparison of $F_{T_{C_c}}(t)$ when C_c varies, with $t = 9.0\,\text{s}$.

A realistic propagation delay is added to the simulation even for the Cloud system. Based on the results from [47], a bimodal distribution is introduced to account for both closer and farther CC facilities. The resulting distribution is a sum of two normal distributions, where the lower tails are truncated in zero to ensure that the obtained times are positive. The resulting PDF is:

$$\varphi\left(t; p, \mu_1, \mu_2, \sigma_1, \sigma_2\right) = p\,\phi\left(t; \mu_1, \sigma_1\right) + (1-p)\,\phi\left(t; \mu_2, \sigma_2\right), \tag{5.14}$$

where $\phi\left(t; \mu, \sigma\right)$ is the same as in (5.13). In the proposed model, however, α_c is chosen to be equal to the mean value of (5.14), so that the two distributions have the same mean value.

The model for $F_{T_{C_c}}(t)$ is compared to simulated data w.r.t variations in the value of C_c. To perform this validation, the same t value as in the MEC case is used, and the results obtained are shown in Figure 5.3. As can be noted, even in CC scenario, where delay is higher and has a more complex distribution, the analytical model is able to give a good prediction of system performance, even if with a higher deviation than in the MEC case.

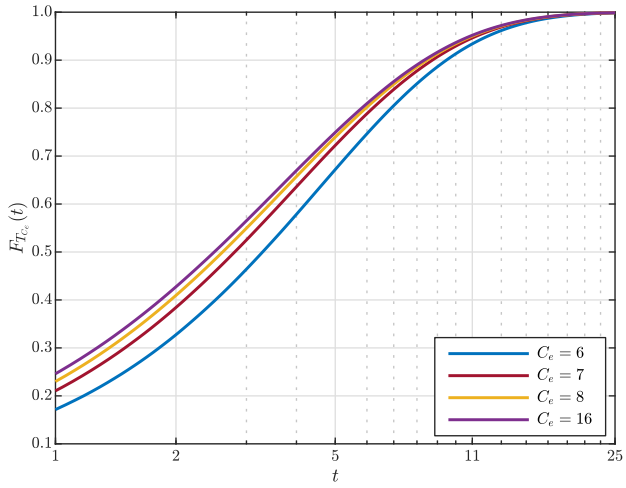

Figure 5.4: $F_{T_{Ce}}(t)$ when t varies, with $\rho_e = 5$.

5.2.3 Effects of Servers Number in Mobile Edge Computing Model

It is interesting to analyze how variations in the number of servers C_e affect the performance of the MEC model. As reported in Section 5.1, we suppose that the rate of service depends on the considered application, and servers work at the maximum rate enabled by state of the art technologies. As a consequence, the parameter that can be chosen to adapt the system performance to the load is C_e. In Figure 5.4 the effect of C_e variation on $F_{T_{Ce}}(t)$ model is evaluated. As shown, the amount of servers affects the performance of $F_{T_{Ce}}(t)$, although the provided gain flattens at the increase of them, as it can be seen in the difference between the case with 8 and 16 servers. Having doubled the computing resources available at the MEC server does not result in a remarkable increase in performance. As a reference, it should be noted that the value of C_{opt}, as defined in [14], for $\rho_e = 5$ is about 8. Hence, C_e has to be correctly chosen depending on the expected system load, as too high values do not necessarily reflect on system performance. Moreover, for probabilities of $T_{C_e} < t$ above 0.9, the effect of increasing C_e value is not very significant.

5.2.4 Effects of Mean Service Time and Reneging Departure Rate

So far, no consideration is given to how the mean service time affects the performance of these two systems. For this reason, an additional configuration is added to those presented above. To represent an application with stricter latency requirements, a scenario with mean service time equal to 0.4 s is added for both MEC and CC systems. This value is more challenging for the average CC system [32].

In the system planning phase, it is important to quantify the service dropping rate. For this reason a new performance metric is defined, η, which not only considers the service completion time, but also the probability of service completion. The definition is as follows:

$$\eta(\rho, C, t) = P_S(C) F_{T_C}(t). \tag{5.15}$$

This new metric represents the probability that a request is completed before a given time, its value ranges between 0 and 1, where 1 indicates that all requests are completed before the given time, and 0 indicates the contrary.

Figure 5.5 and 5.6 show the trend of this metric for both MEC and CC systems, with the two mean service time values and $\mu_H = 0.025\,\mathrm{s}^{-1}$. The application deadline is set at 225% of the mean service time. The system load is represented by ρ/C, normalizing the load factor to the number of servers available to MEC or CC systems.

To get an insight as much complete as possible, analytical results are complemented with numerical ones from a simulator, obtained in more realistic traffic conditions. For this purpose, the computation time required by each offloaded task is represented by a proper model for real network traffic, the Pareto distribution [46], and the propagation delays are represented by the previously defined distributions.

As can be noted in Figure 5.5, with mean service time equal to 4.0 s the performance offered by the MEC system is never as good as that of CC. In terms of the probability of requests completed before the deadline, the MEC based solution shows lower values, regardless of the system load. Moreover, changing the number of servers does not improve performance. As can be seen, even considering the highest C_e, there is an upper bound to a value which is about the one achievable by the CC system. Besides, the delay constraint is so low that it can easily include CC latencies. In this particular test, the data provided by the model are in good agreement

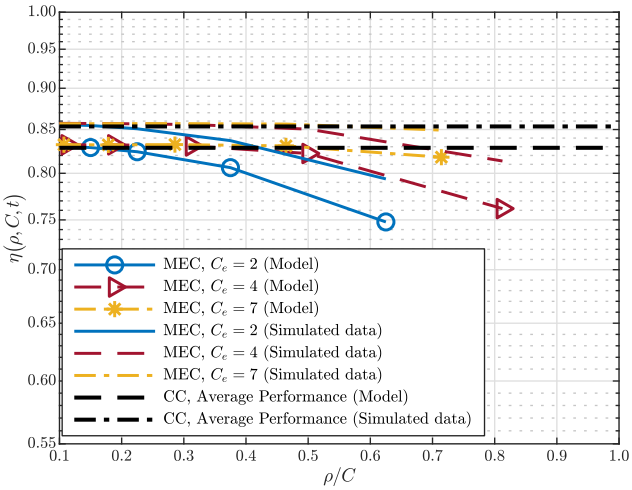

Figure 5.5: $\eta(\rho, C, t)$ with $t = 9.0\,\mathrm{s}$, $\mu_S = 4.0\,\mathrm{s}$, and $\mu_H = 0.025\,\mathrm{s}^{-1}$. Given the discrete nature of C, the entire range of ρ/C, can not always be fully covered.

with the simulator, except for a constant deviation, due to some necessary approximations.

In Figure 5.6 the performance of the scenario with mean service time equal to 0.4 s is presented, where the MEC system can really compete with CC. Here the deadline is close enough that the latency introduced by the communications with the CC system has a clear impact on performance. For low system load values, MEC system offers a greater performance than in the CC case. Then, as the load increases, $\eta(\rho, C_e, t)$ drops under the average CC performance. This is where it becomes more convenient to start offloading to CC. The data provided by the model give a clear picture of whether and when it is appropriate to use the MEC node. Besides, it can be observed that increasing or decreasing the number of servers available to the system has a very noticeable effect, as it shifts the crossing point. Regarding the latter Figure, it is important to underline the good adherence of the model to the simulated data, which makes it a very powerful tool to predict the behavior of similar systems.

To further complete the analysis, μ_H is also varied for both MEC and CC

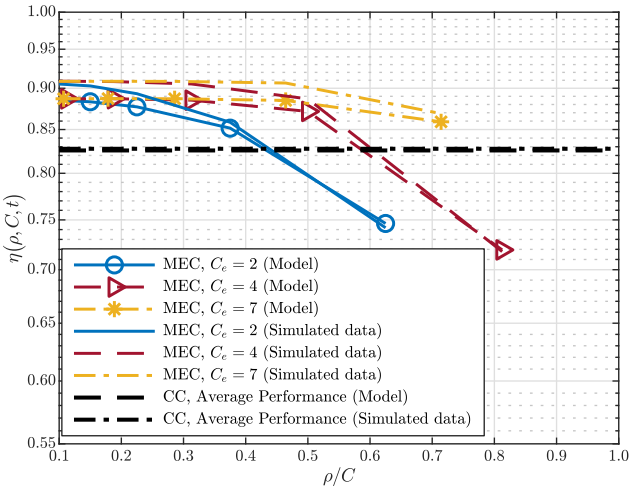

Figure 5.6: $\eta(\rho, C, t)$ with $t = 0.9\,\mathrm{s}$, $\mu_S = 0.4\,\mathrm{s}$, and $\mu_H = 0.025\,\mathrm{s}^{-1}$. Given the discrete nature of C, the entire range of ρ/C, can not always be fully covered.

cases. In Figures 5.7 and 5.8 the value of η is shown for both service time values and $\mu_H = 0.010\,\mathrm{s}^{-1}$, which represents a lower mobility scenario [14]. As can be noted in Figures 5.7 and 5.8, the effect of mobility on η is much smaller for $T_S = 0.4\,\mathrm{s}$ than for $T_S = 4\,\mathrm{s}$. This is due to how $P_S(C)$ behaves w.r.t. T_S. An example of this dependency can be observed by considering the minimum obtainable value for $C \to \infty$, as shown in Figure 5.9.

For this reason the plots in Figures 5.6 and 5.8 are very similar, while in Figures 5.5 and 5.7 the difference is instead more evident. Mobility does not affect the performance of systems in a significant way when service times are very short, and the difference in performance between MEC and CC systems is still mainly due to the delay introduced by the connections.

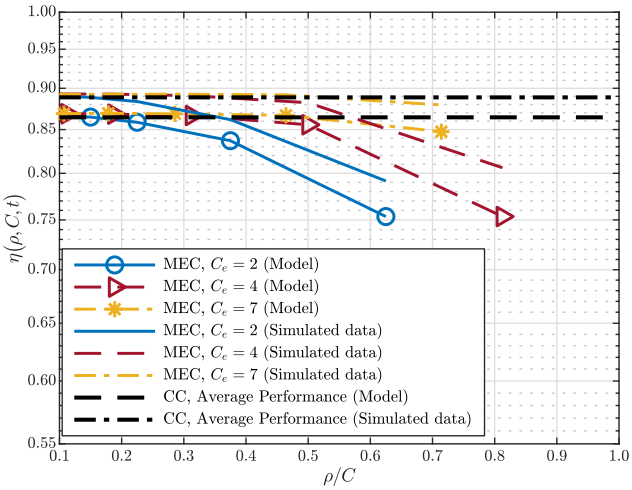

Figure 5.7: $\eta(\rho, C, t)$ with $t = 9.0\,\mathrm{s}$, $\mu_S = 4.0\,\mathrm{s}$, and $\mu_H = 0.010\,\mathrm{s}^{-1}$. Given the discrete nature of C, the entire range of ρ/C, can not always be fully covered.

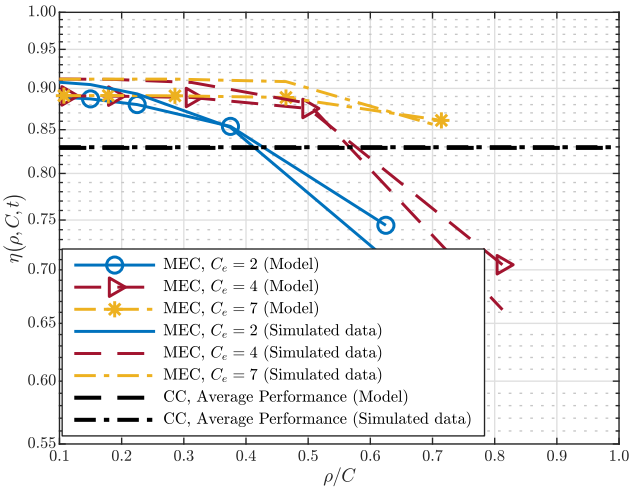

Figure 5.8: $\eta(\rho, C, t)$ with $t = 0.9\,\mathrm{s}$, $\mu_S = 0.4\,\mathrm{s}$, and $\mu_H = 0.010\,\mathrm{s}^{-1}$. Given the discrete nature of C, the entire range of ρ/C, can not always be fully covered.

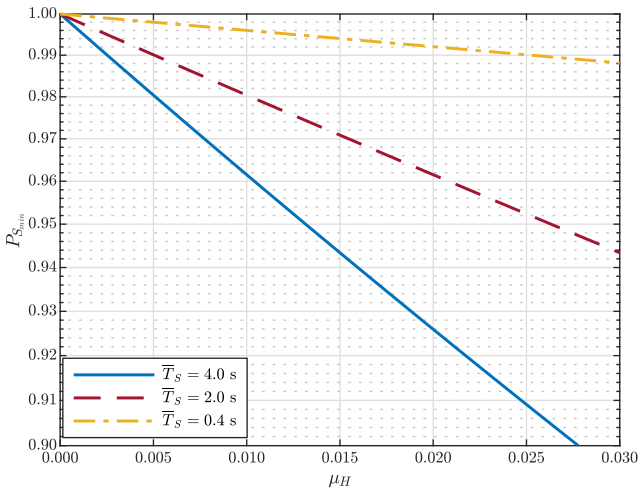

Figure 5.9: $P_{S_{\min}}$ when μ_H varies for different values of $\overline{T}_S$.

Chapter 6

Conclusion

This chapter summarizes the contribution of the thesis and discusses avenues for future research.

6.1 Summary of Contribution

This thesis presents two different approaches to vehicular networking. The first part is an integrated system architecture inspired to the Fog paradigm, applied to achieve a full context awareness for VANET and, consequently, to react on traffic anomalous conditions. Then the MEC paradigm is introduced, innovative services, previously impossible, such as computational offloading, could be offered by nodes outside the VANET, while maintaining a delay comparable to that of on-site computation.

In particular, regarding the Fog paradigm we propose to adopt a specific co-designed approach involving Application and Networks Layers. For the latter one, we investigated classical DTN Flooding based, NC multiflows and Chord protocols, while we resort to BC technology to achieve a distributed consensus. The system has been tested by resorting to OMNeT++ and Veins frameworks for their modularity, high fidelity and flexibility. Although the three methods are used in slightly different scenarios, interesting data can still be extracted. Performance analysis pointed out that DTN probability-based methods is in charge of reaching a significant percentage of cars, without requiring excessive overhead. On the other hand, NC protocols are an interesting alternative to the previous ones, with considerably better performance, even in a scenario where only 4 cars are involved, even

in the presence of packet loss. Finally, where Chord scheme is adopted good performance is pointed out in terms of consensus making overhead, while the higher cost is due to P2P network formation via Chord protocol which in turns allows a highly efficient and resilient topology control.

The MEC paradigm is considered as a novel approach to face the ever increasing demand of mobile computing applications, providing computing and storage capabilities at the network edges, in close proximity to end users. The possibility of direct wireless connections between MEDs and the MEC facility empowers the support of ultra-low latency applications and lowers MEDs' power consumption. In particular, the work focused on a MEC system devoted to provide computational capabilities within a limited area, according to the SaaS paradigm. A multi-server queuing system model with requests reneging has been proposed to derive the performance of the considered MEC system and to identify the optimal (*i.e.*, minimum) number of servers (*i.e.*, VMs) to be allocated at the MEC facility in order to have the probability of dropping an on-going service request lower than a target value. The proposed queueing system model has been finally validated by comparing the obtained analytical predictions with numerical results, derived by resorting to extensive computer simulations, under the assumption of actual operating conditions.

Finally, although MEC architecture is commonly considered as a solution to the growing need for distributed IoT, this work has shown that, depending on the load of the system, it may be preferable to find a balance between MEC and CC. In particular, we eventually focused on a comparison between MEC and CC systems devoted to providing computational capabilities with low-latency constraints. Two respective analytical models were investigated, and the resulting data were compared with more realistic data taken from a network simulator. We shed some light on what aspects should be considered when designing a MEC system, especially considering the amount of resources available to it, depending on the system load, which is a novelty compared to the past literature. It is clear from the presented results that there is no optimal solution for every condition. MEC architecture may not always be the answer for every type of service request. Our study shown that MEC systems clearly pointed out better performance than CC systems for applications with low-latency requirements. However, the system load clearly affects the performance of the MEC system, and therefore the choice of the best system to offload. Besides, the amount of resource provided to

the MEC node has a clear impact on the aforementioned threshold up to a given point, which is discussed in this work. This result shows the need for a dynamic system that is able to route offloading requests according to the state of the system, to maintain the overall performance at an optimal level.

6.2 Directions for Future Work

Vehicular networking is a very difficult field to deal with. The industry is still very young and it is not yet clear what services will be required by the cars of the future. The applications are constantly evolving, very often with requirements that pose challenges to the network, which typically evolves with different logic. Considering all the work done, it is clear that 5G technology is fundamental for the support of these new services. However, it is important that studies like this are continuously performed in order to align the capabilities of telecommunications networks with the demands of this fast growing market.

Data dissemination is perhaps the most challenging area considered. It is very difficult to obtain actual P2P communications, and the resulting network is very fragile, due to the unpredictable mobility of users. As shown many resources must be spent in the creation of an overlay that enables the exchange of messages. Research in this area cannot stop, and researchers must work closely with manufacturers and standardization bodies.

It is very important to continue to investigate which computing and networking schemes dynamically respond better to the specific application scenarios that the vehicular context poses, or that may emerge in the future. The comparison between MEC and CC schemes is a first step in this sense, revealing that there is not a very good choice, and it is always necessary to adapt both to the system load, and especially to the constraints of the particular service considered. The challenges are more at the management and policy levels.

The models presented in this work are very versatile and in the future, could be extended to consider several adjacent MEC systems. Since the 5G access network already provides the architecture needed to the handover process, it should also be taken into account, and not be considered as a loss event. Interactions among different MEC nodes should be studied, and, in particular, how delays associated with handover procedure can affect the service completion time. This will enable predictions of system behavior to

be made even in scenarios characterized by an even higher mobility, which is particularly useful for future vehicular use cases. Furthermore, the study of computational offloading with handover may be complemented by an analysis of task scheduling procedures. This level of detail could help to further improve the performance of the system, which would result in better accuracy and less waste of available resources.

For the future, sixth generation (or 6G) technologies have the potential to further increase the capacities observed in 5G networks. In particular, the ultramassive multiple-input and multiple-output (MIMO) scheme, specifically designed for contexts where a huge number of devices insist on the same geographical area, enables the large number of simultaneous transmissions required by this increasingly data-hungry sector.

Appendix A

Appendix

A.1 Proof of (4.12) in Section 4.1

To derive the limit of $\overline{T}_t(C)$ as in (4.12) some intermediate steps are needed. From (4.9) some initial considerations can be made. The right hand side term of both numerator and denominator can be omitted, as their values tend to zero. Hence, it results as follows:

$$\lim_{C\to\infty} \overline{T}_t(C) = \lim_{C\to\infty} \frac{\displaystyle\sum_{n=1}^{C-1} \frac{n\lambda^n}{n!\,(\mu_H+\mu_S)^n}}{\lambda\left[\displaystyle\sum_{n=0}^{C-1} \frac{\lambda^n}{n!\,(\mu_H+\mu_S)^n}\right]}. \tag{A.1}$$

The two summations can be resolved by referring to following power series:

$$\sum_{n=1}^{\infty} \frac{nx^n}{n!} = xe^x, \qquad \sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x,$$

As a consequence, the limit of $\overline{T}_t(C)$ results as follows:

$$\lim_{C\to\infty} \overline{T}_t(C) = \frac{\dfrac{\lambda}{\mu_H+\mu_S}\exp\Big(\frac{\lambda}{\mu_H+\mu_S}\Big)}{\lambda\exp\Big(\frac{\lambda}{\mu_H+\mu_S}\Big)}, \tag{A.2}$$

which gives (4.12).

A.2 Expansion of (5.8) in Section 5.1

The expanded version of (5.8), not included in the text for the sake of clarity, is shown below.

$$F_{T_{C_e}}(t) =$$

$$\begin{cases} 0 & t < 0 \\ \frac{\mu_S^2 P_{Q_e}\overline{T}_{Q_e}\left\{P_{S_e}P_{Q_e}\overline{T}_{Q_e}\left[1-\exp\left(\frac{-t}{P_{S_e}P_{Q_e}\overline{T}_{Q_e}}\right)\right]-t\right\}-P_{S_e}\left[1-\exp\left(\frac{-\mu_S t}{P_{S_e}}\right)\right]+\mu_S t}{2\alpha_e\mu_S\left(1-\mu_S P_{Q_e}\overline{T}_{Q_e}\right)} & 0 \le t < 2\alpha_e \\ \frac{\mu_S^2 P_{S_e}P_{Q_e}^2\overline{T}_{Q_e}^2\left\{1-\exp\left(\frac{-2\alpha_e}{P_{S_e}P_{Q_e}\overline{T}_{Q_e}}\right)-\left[\exp\left(\frac{-2\alpha_e-t}{P_{S_e}P_{Q_e}\overline{T}_{Q_e}}\right)-\exp\left(\frac{-t}{P_{S_e}P_{Q_e}\overline{T}_{Q_e}}\right)\right]\left[\exp\left(\frac{2\alpha_e}{P_{S_e}P_{Q_e}\overline{T}_{Q_e}}\right)-\exp\left(\frac{t}{P_{S_e}P_{Q_e}\overline{T}_{Q_e}}\right)\right]\right\}}{2\alpha_e\mu_S\left(1-\mu_S P_{Q_e}\overline{T}_{Q_e}\right)} & \\ +\frac{-2\alpha_e\mu_S^2 P_{Q_e}\overline{T}_{Q_e}+P_{S_e}\left[\exp\left(\frac{-\mu_S t}{P_{S_e}}\right)-\exp\left(\frac{\mu_S(2\alpha_e-t)}{P_{S_e}}\right)\right]+2\alpha_e\mu_S}{2\alpha_e\mu_S\left(1-\mu_S P_{Q_e}\overline{T}_{Q_e}\right)} & t \ge 2\alpha_e \end{cases} \tag{A.3}$$

A.3 Expansion of (5.12) in Section 5.1

The expanded version of (5.12), not included in the text for the sake of clarity, is shown below.

$$F_{T_{C_c}}(t) =$$

$$\begin{cases} 0 & t < 0 \\ \frac{\mu_S^2 P_{Q_c}\overline{T}_{Q_c}\left\{P_{S_c}P_{Q_c}\overline{T}_{Q_c}\left[1-\exp\left(\frac{-t}{P_{S_c}P_{Q_c}\overline{T}_{Q_c}}\right)\right]-t\right\}-P_{S_c}\left[1-\exp\left(\frac{-\mu_S t}{P_{S_c}}\right)\right]+\mu_S t}{2\alpha_c\mu_S\left(1-\mu_S P_{Q_c}\overline{T}_{Q_c}\right)} & 0 \le t < 2\alpha_c \\ \frac{\mu_S^2 P_{S_c}P_{Q_c}^2\overline{T}_{Q_c}^2\left\{1-\exp\left(\frac{-2\alpha_c}{P_{S_c}P_{Q_c}\overline{T}_{Q_c}}\right)-\left[\exp\left(\frac{-2\alpha_c-t}{P_{S_c}P_{Q_c}\overline{T}_{Q_c}}\right)-\exp\left(\frac{-t}{P_{S_c}P_{Q_c}\overline{T}_{Q_c}}\right)\right]\left[\exp\left(\frac{2\alpha_c}{P_{S_c}P_{Q_c}\overline{T}_{Q_c}}\right)-\exp\left(\frac{t}{P_{S_c}P_{Q_c}\overline{T}_{Q_c}}\right)\right]\right\}}{2\alpha_c\mu_S\left(1-\mu_S P_{Q_c}\overline{T}_{Q_c}\right)} & \\ +\frac{-2\alpha_c\mu_S^2 P_{Q_c}\overline{T}_{Q_c}+P_{S_c}\left[\exp\left(\frac{-\mu_S t}{P_{S_c}}\right)-\exp\left(\frac{\mu_S(2\alpha_c-t)}{P_{S_c}}\right)\right]+2\alpha_c\mu_S}{2\alpha_c\mu_S\left(1-\mu_S P_{Q_c}\overline{T}_{Q_c}\right)} & t \ge 2\alpha_c \end{cases} \tag{A.4}$$

Appendix B

Publications

This research activity has led to several publications in international journals and conferences. These are summarized below.[1]

International Journals

1. **A. Bonadio**, F. Chiti, R. Fantacci, and V. Vespri, "An Integrated Framework for Blockchain inspired Fog Communications and Computing in Internet of Vehicles", *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 2, pp. 755–762, Feb. 2020. [DOI:10.1007/s12652-019-01476-y] *10 citations*
2. **A. Bonadio**, F. Chiti, and R. Fantacci, "Performance Analysis of an Edge Computing SaaS System for Mobile Users", *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2049–2057, Feb. 2020. [DOI:10.1109/TVT.2019.2957938] *3 citations*

National Conferences

1. **A. Bonadio**, F. Chiti, and R. Fantacci, "An Integrated Framework for Fog Communications and Computing in Internet of Vehicles", in *Proceedings of the 5th International OMNeT++ Community Summit*, Pisa, Italy, Sep. 2018, pp. 84–92.
2. F. Nizzi, T. Pecorella, **A. Bonadio**, F. Chiti, R. Fantacci, D. Tarchi, and W. Cerroni, "FOG-oriented Joint Computing and Networking: the GAU-

[1] The author's bibliometric indices are the following: H-index = 2, total number of citations = 13 (source: Google Scholar on February 18, 2021).

ChO Project Vision", in *2018 AEIT International Annual Conference*, Bari, Italy, Oct. 2018, pp. 1–5.

Bibliography

[1] "Car 2 Car - Communication Constortium: Organisation," Accessed on Jul. 30, 2017. [Online]. Available: https://www.car-2-car.org/index.php

[2] *Minimum requirements related to technical performance for IMT-2020 radio interface(s)*, Report ITU-R M.2410-0, Nov. 2017.

[3] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[4] T. Abdelkader, K. Naik, A. Nayak, N. Goel, and V. Srivastava, "A performance comparison of delay-tolerant network routing protocols," *IEEE Network*, vol. 30, no. 2, pp. 46–53, Mar. 2016.

[5] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 337–368, Firstquarter 2014.

[6] A. Aissioui, A. Ksentini, A. M. Gueroui, and T. Taleb, "On Enabling 5G Automotive Systems Using Follow Me Edge-Cloud Concept," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5302–5316, Jun. 2018.

[7] A. Al-Shuwaili and O. Simeone, "Energy-Efficient Resource Allocation for Mobile Edge Computing-Based Augmented Reality Applications," *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 398–401, Jun. 2017.

[8] K. L. Ang and J. K. P. Seng, "Application Specific Internet of Things (ASIoTs): Taxonomy, Applications, Use Case and Future Directions," *IEEE Access*, vol. 7, pp. 56 577–56 590, 2019.

[9] J. C. Augusto, V. Callaghan, D. Cook, A. Kameas, and I. Satoh, "Intelligent Environments: a manifesto," *Human-centric Computing and Information Sciences*, vol. 3, no. 12, pp. 1–18, Jun. 2013.

[10] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN Routing As a Resource Allocation Problem," *SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 373–384, Aug. 2007.

[11] Y. Bi, G. Han, C. Lin, Q. Deng, L. Guo, and F. Li, "Mobility Support for Fog Computing: An SDN Approach," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 53–59, May 2018.

[12] K. Bilal, O. Khalid, A. Erbad, and S. U.Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," *Computer Networks*, vol. 130, pp. 94–120, Jan. 2018.

[13] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. New York, NY, USA: Wiley-Interscience, 1998.

[14] A. Bonadio, F. Chiti, and R. Fantacci, "Performance Analysis of an Edge Computing SaaS System for Mobile Users," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2049–2057, Feb. 2020.

[15] Z. Chang, Z. Zhou, T. Ristaniemi, and Z. Niu, "Energy Efficient Optimization for Computation Offloading in Fog Computing System," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, Singapore, Dec. 2017, pp. 1–6.

[16] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, "Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017.

[17] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[18] M. Conti and S. Giordano, "Multihop Ad Hoc Networking: The Theory," *IEEE Communications Magazine*, vol. 45, no. 4, pp. 78–86, 2007.

[19] L. Decreusefond and P. Moyal, *Stochastic Modeling and Analysis of Telecom Networks*. Wiley, 2013.

[20] Directive 2010/40/EU of the European Parliament and of the Council, "On the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport," *Official Journal of the European Union*, Jul. 2010.

[21] K. Dolui and S. K. Datta, "Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile Edge Computing," in *2017 Global Internet of Things Summit (GIoTS)*, Geneva, Switzerland, Jun. 2017, pp. 1–6.

[22] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "BlockChain: A Distributed Solution to Automotive Security and Privacy," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, Dec. 2017.

[23] F. Dressler, C. Sommer, D. Eckhoff, and O. K. Tonguz, "Toward Realistic Simulation of Intervehicle Communication," *IEEE Vehicular Technology Magazine*, vol. 6, no. 3, pp. 43–51, Sep. 2011.

[24] X. Duan, Y. Liu, and X. Wang, "SDN Enabled 5G-VANET: Adaptive Vehicle Clustering and Beamformed Transmission for Aggregated Traffic," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 120–127, 2017.

[25] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C. Lin, "Edge of Things: The Big Picture on the Integration of Edge, IoT and the Cloud in a Distributed Computing Environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2018.

[26] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward Low-Latency and Ultra-Reliable Virtual Reality," *IEEE Network*, vol. 32, no. 2, pp. 78–84, Mar. 2018.

[27] I. Eyal, "Blockchain Technology: Transforming Libertarian Cryptocurrency Dreams to Finance and Banking Realities," *Computer*, vol. 50, no. 9, pp. 38–49, 2017.

[28] R. Farha and A. Leon-Garcia, "Autonomic Resource Management for Multimedia Services Using Inventory Control," in *Real-Time Mobile Multimedia Services*, D. Krishnaswamy, T. Pfeifer, and D. Raz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 161–172.

[29] M. Gerla, E. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, Mar. 2014, pp. 241–246.

[30] R. A. Guerin, "Channel occupancy time distribution in a cellular radio system," *IEEE Transactions on Vehicular Technology*, vol. 36, no. 3, pp. 89–99, Aug. 1987.

[31] N. Gupta, "Performance analysis using some Queueing models," Ph.D. dissertation, Jaypee Institute of Information Technology, Noida, India, May 2010. [Online]. Available: http://hdl.handle.net/10603/3357

[32] K. Ha, P. Pillai, G. Lewis, S. Simanta, S. Clinch, N. Davies, and M. Satyanarayanan, "The Impact of Mobile Multimedia Applications on Data Center Consolidation," in *2013 IEEE International Conference on Cloud Engineering (IC2E)*, Redwood City, CA, USA, Mar. 2013, pp. 166–176.

[33] F. A. Haight, "Queueing With Reneging," *Metrika*, vol. 2, no. 1, pp. 186–197, Dec. 1959.

[34] Z. Han, H. Tan, X. Li, S. H. C. Jiang, Y. Li, and F. C. M. Lau, "OnDisc: Online Latency-Sensitive Job Dispatching and Scheduling in Heterogeneous Edge-Clouds," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2472–2485, Dec. 2019.

[35] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[36] W. He, G. Yan, and L. D. Xu, "Developing Vehicular Data Cloud Services in the IoT Environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587–1595, 2014.

[37] R. Henry, A. Herzberg, and A. Kate, "Blockchain Access Privacy: Challenges and Directions," *IEEE Security and Privacy*, vol. 16, no. 4, pp. 38–45, Jul. 2018.

[38] D. Hong and S. S. Rappaport, "Traffic Model and Performance Analysis for Cellular Mobile Radio Telephone Systems with Prioritized and Nonprioritized Handoff Procedures," *IEEE Transactions on Vehicular Technology*, vol. 35, no. 3, pp. 77–92, Aug. 1986.

[39] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile Fog: A Programming Model for Large-Scale Applications on the Internet of Things," in *Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing*, Hong Kong, China, Aug. 2013, p. 15–20.

[40] A. Hosseinian-Far, M. Ramachandran, and C. L. Slack, "Emerging Trends in Cloud Computing, Big Data, Fog Computing, IoT and Smart Living," in *Technology for Smart Futures*, M. Dastbaz, H. Arabnia, and B. Akhgar, Eds. Cham: Springer, 2018, ch. 2, pp. 29–40.

[41] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile Edge Computing A key technology towards 5G," White Paper No. 11, ETSI, Sophia Antipolis, France, 2015.

[42] C. Jarray and A. Giovanidis, "The Effects of Mobility on the Hit Performance of Cached D2D Networks," in *2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Tempe, AZ, USA, May 2016, pp. 1–8.

[43] E. P. C. Jones, L. Li, J. K. Schmidtke, and P. A. S. Ward, "Practical Routing in Delay-Tolerant Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 943–959, Aug. 2007.

[44] U. Lee, J.-S. Park, J. Yeh, G. Pau, and M. Gerla, "Code Torrent: Content Distribution Using Network Coding in VANET," in *Proceedings of MobiShare '06*. New York, NY, USA: ACM, 2006, pp. 1–5.

[45] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, Dec. 2017.

[46] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[47] M. Leonhard. CloudPing.info. Accessed on Jan. 13, 2020. [Online]. Available: https://www.cloudping.info/

[48] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective Optimization for Computation Offloading in Fog Computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, Feb. 2018.

[49] X. Liu, J. Zhang, X. Zhang, and W. Wang, "Mobility-Aware Coded Probabilistic Caching Scheme for MEC-Enabled Small Cell Networks," *IEEE Access*, vol. 5, pp. 17 824–17 833, 2017.

[50] K. Maheshwari and A. Kumar, "Performance Analysis of Microcellization for Supporting Two Mobility Classes in Cellular Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 321–333, Mar. 2000.

[51] S. S. Manvi and S. Tangade, "A survey on authentication schemes in VANETs for secured communication," *Vehicular Communications*, vol. 9, pp. 19–30, Jul. 2017.

[52] A. Manzalini and N. Crespi, "An Edge Operating System Enabling Anything-as-a-Service," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 62–67, Mar. 2016.

[53] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[54] J. Meng, H. Tan, X. Li, Z. Han, and B. Li, "Online Deadline-Aware Task Dispatching and Scheduling in Edge Computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 6, pp. 1270–1286, 2020.

[55] J. Mikulski, "DynaT*A*C cellular portable Radiotelephone System Experience in the U.S. and the UK," *IEEE Communications Magazine*, vol. 24, no. 2, pp. 40–46, Feb. 1986.

[56] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions," *IEEE Access*, vol. 6, pp. 47 980–48 009, 2018.

[57] K. Nahrstedt, H. Li, P. Nguyen, S. Chang, and L. Vu, "Internet of Mobile Things: Mobility-Driven Challenges, Designs and Implementations," in *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Berlin, Germany, Apr. 2016, pp. 25–36.

[58] F. Nakahara and D. Beder, "A context-aware and self-adaptive offloading decision support model for mobile cloud computing system," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 5, pp. 1561–1572, Oct. 2018.

[59] J. Pan and J. McElhannon, "Future Edge Cloud and Edge Computing for Internet of Things Applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, Feb. 2018.

[60] Y. Park, C. Sur, and K.-H. Rhee, "Pseudonymous authentication for secure V2I services in cloud-based vehicular networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 5, pp. 661–671, Oct. 2016.

[61] V. Pla and V. Casares-Giner, "Analytical-Numerical Study of the Handoff Area Sojourn Time," in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 1, Taipei, Taiwan, Nov. 2002, pp. 886–890.

[62] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, "The Blockchain as a Decentralized Security Framework [Future Directions]," *IEEE Consumer Electronics Magazine*, vol. 7, no. 2, pp. 18–21, Mar. 2018.

[63] A. Reiter, B. Prünster, and T. Zefferer, "Hybrid Mobile Edge Computing: Unleashing the Full Potential of Edge Computing in Mobile Device Use Cases," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, Madrid, Spain, May 2017, pp. 935–944.

[64] J. Rodriguez, *Fundamentals of 5G Mobile Networks*, 1st ed. Wiley Publishing, 2015.

[65] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[66] S. Shin, U. Lee, F. Dressler, and H. Yoon, "Analysis of Cell Sojourn Time in Heterogeneous Networks With Small Cells," *IEEE Communications Letters*, vol. 20, no. 4, pp. 788–791, Apr. 2016.

[67] F. Silva and C. Analide, "Ubiquitous driving and community knowledge," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 2, pp. 157–166, Apr. 2017.

[68] S. Singh, Y. Chiu, Y. Tsai, and J. Yang, "Mobile Edge Fog Computing in 5G Era: Architecture and Implementation," in *2016 International Computer Symposium (ICS)*, Chiayi, Taiwan, Dec. 2016, pp. 731–735.

[69] S. A. Soleymani, A. H. Abdullah, M. Zareei, M. H. Anisi, C. Vargas-Rosales, M. K. Khan, and S. Goudarzi, "A Secure Trust Model Based on Fuzzy Logic in Vehicular Ad Hoc Networks With Fog Computing," *IEEE Access*, vol. 5, pp. 15 619–15 629, 2017.

[70] E. S. Sopin, A. V. Daraseliya, and L. M. Correia, "Performance Analysis of the Offloading Scheme in a Fog Computing System," in *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Moscow, Russia, Nov. 2018, pp. 1–5.

[71] W. Stallings, *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*, 3rd ed. 800 East 96th Street, Indianapolis, Indiana 46240 USA: Pearson, 2019.

[72] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," *SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 149–160, Aug. 2001.

[73] D. Tacconi, I. Carreras, D. Miorandi, I. Chlamtac, F. Chiti, and R. Fantacci, "Supporting the Sink Mobility: a Case Study for Wireless Sensor Networks," in *2007 IEEE International Conference on Communications*, Glasgow, UK, Jun. 2007, pp. 3948–3953.

[74] L. E. Talavera, M. Endler, I. Vasconcelos, R. Vasconcelos, M. Cunha, and F. J. d. S. e. Silva, "The Mobile Hub concept: Enabling applications for the Internet of Mobile Things," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, St. Louis, MO, USA, Mar. 2015, pp. 123–128.

[75] L. Tong, Y. Li, and W. Gao, "A Hierarchical Edge Cloud Architecture for Mobile Computing," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[76] Y. Toor, P. Muhlethaler, A. Laouiti, and A. D. L. Fortelle, "Vehicle Ad Hoc networks: applications and related technical issues," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 3, pp. 74–88, Thirdquarter 2008.

[77] S. M. Tornell, C. T. Calafate, J. C. Cano, and P. Manzoni, "DTN Protocols for Vehicular Networks: An Application Oriented Overview," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 2, pp. 868–887, Secondquarter 2015.

[78] R. A. Uzcategui, A. J. D. Sucre, and G. Acosta-Marum, "Wave: A tutorial," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 126–133, May 2009.

[79] M. Veeresha and M. Sugumaran, "Optimal hybrid broadcast scheduling and adaptive cooperative caching for spatial queries in road networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 4, pp. 607–624, Aug. 2017.

[80] A. M. Vegni and V. Loscrí, "A Survey on Vehicular Social Networks," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2397–2419, Fourthquarter 2015.

[81] T. Verbelen, P. Simoens, F. D. Turck, and B. Dhoedt, "Leveraging Cloudlets for Immersive Collaborative Applications," *IEEE Pervasive Computing*, vol. 12, no. 4, pp. 30–38, Oct. 2013.

[82] C. Wan and J. Zhang, "Efficient identity-based data transmission for VANET," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1861–1871, Nov. 2018.

[83] H. Wu, S. Deng, W. Li, S. U. Khan, J. Yin, and A. Y. Zomaya, "Request Dispatching for Minimizing Service Response Time in Edge Cloud Systems," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, Hangzhou, China, Jul. 2018, pp. 1–9.

[84] L. Xiao, T. Chen, C. Xie, H. Dai, and V. Poor, "Mobile Crowdsensing Games in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1535–1545, 2018.

[85] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, "Efficient Mobility-Aware Task Offloading for Vehicular Edge Computing Networks," *IEEE Access*, vol. 7, pp. 26 652–26 664, 2019.

[86] L. Yang, J. Cao, G. Liang, and X. Han, "Cost Aware Service Placement and Load Dispatching in Mobile Cloud Systems," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1440–1452, May 2016.

[87] K. Yeow, A. Gani, R. W. Ahmad, J. J. P. C. Rodrigues, and K. Ko, "Decentralized Consensus for Edge-Centric Internet of Things: A Review, Taxonomy, and Research Issues," *IEEE Access*, vol. 6, pp. 1513–1524, 2018.

[88] Y. Zhang, D. Niyato, and P. Wang, "Offloading in Mobile Cloudlet Systems with Intermittent Connectivity," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, Dec. 2015.

[89] T. Zhou, R. R. Choudhury, P. Ning, and K. Chakrabarty, "P2DAP — Sybil Attacks Detection in Vehicular Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 582–594, Mar. 2011.

[90] M. M. Zonoozi and P. Dassanayake, "User Mobility Modeling and Characterization of Mobility Patterns," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1239–1252, Sep. 1997.